PIC32CM LE00/LS00/LS60

Ultra-Low Power, Secure and Enhanced Touch MCU

512-KB Flash, 64-KB SRAM with TrustZone, Crypto & Enhanced PTC

Operating Conditions

- 1.62V - 3.63V, -40ºC to +85ºC, DC to 48 MHz (PIC32CM LE00/LS00)
- 2.0V - 3.63V, -40ºC to +85ºC, DC to 48 MHz (PIC32CM LS60)

Core

- Arm® Cortex®-M23 CPU running at up to 48 MHz:
  - 2.64 CoreMark/MHz and 1.03 DMIPS/MHz
  - Single-cycle hardware multiplier
  - Hardware divider
  - Nested Vector Interrupt Controller (NVIC)
  - Memory Protection Unit (MPU)
  - Stack Limit Checking
  - TrustZone® for ARMv8-M (optional)

Memories

- 512/256 KB Flash
- 16/8 KB Data Flash Write-While-Read (WWR) section for non-volatile data storage
- 64/32 KB SRAM
- 512 bytes TrustRAM with physical protection features
- 32 KB Boot ROM

System

- Power-on Reset (POR) and programmable Brown-out Detection (BOD)
- 16-channel Direct Memory Access Controller (DMAC)
- 12-channel event system for Inter-peripheral Core-independent Operation
- CRC-32 generator

Low-Power and Power Management

- Active, Idle, Standby with partial or full SRAM retention and off sleep modes:
  - Active mode (< 40 µA/MHz for PL0, < 60 µA/MHz for PL2)
  - Idle mode (< 15 µA/MHz) with 1.5 µs wake-up time
  - Standby with Full SRAM retention (1.7 µA) with 2.7 µs wake-up time
  - Off mode (< 100 nA)
- Static and dynamic power gating architecture
- Sleepwalking peripherals
- Two performance levels
- Embedded Buck/LDO regulator with on-the-fly selection

Security and Safety

- Up to eight input pins and eight output pins for anti-tampering detections
- Data Flash
  - Optimized for secure storage
  - Address and Data Scrambling with user-defined key (optional)
  - Tamper erase of scrambling key and of one user-defined row
  - Silent access for data read noise reduction

Security and Safety (Continued)

- TrustRAM
  - Address and Data scrambling with user-defined key
  - Anti-tamper Active Shield on physical TrustRAM
  - Tamper Erase of scrambling key and TrustRAM data
  - Silent access for data read noise reduction
  - Data remanence prevention
- Peripherals
  - One True Random Number Generator (TRNG)
  - AES-256/192/128, SHA-256, and GCM cryptography accelerators (optional)
  - Secure pin multiplexing capability to isolate secure communication channels with external devices from the non-secure application (optional)
- TrustZone for flexible hardware isolation of memories and peripherals (optional)
  - Up to five regions for the Flash
  - Up to two regions for the Data Flash
  - Up to two regions for the SRAM
  - Individual security attribution for each peripheral, I/O, external interrupt line, and Event System Channel
- SHA-based or HMAC-based secure boot (optional)
- ATECC608B-TFLXTLS CryptoAuthentication™ device (optional)
  - One Permanent Primary P-256 Elliptic Curve Cryptography (ECC) Private Key
  - One Internal Sign Private Key for Key Attestation
  - Three Secondary P-256 ECC Private Keys
  - Signer Public Key from Signer Certificate
  - Public Key Validation Support
  - One Customizable Symmetric Secret Key Slot
  - I/O Protection Key Slot to Protect Communication
  - Secure Boot Enabled with Customizable Secure Boot Public Key
  - ECDH/KDF Key Slot Capable
  - X.509 Compressed Certificate Storage
  - Customizable Certificate Storage Slots
- Device Identity Composition Engine (DICE) security standard support with Unique Device Secret (UDS) (optional)
- Up to three debug access levels
- Up to three chip erase commands to erase part of or the entire embedded memories
- Unique 128-bit serial number

Timers/Output Compare/Input Capture

- Three 16-bit Timers/Counters (TC), each configurable as:
  - One 16-bit TC with two compare/capture channels
  - One 8-bit TC with two compare/capture channels
  - One 32-bit TC with two compare/capture channels, by using two TCs
- Three 24-bit Timers/Counters for Control (TCC), with configurable extensions:
  - Fault Detection
  - Dithering
  - Dead-time insertion
  - Pattern Generation
- One 16-bit Timers/Counter for Control (TCC), with fault detection support
- 32-bit Real-Time Counter (RTC) with clock/calendar functions
- PWM Outputs using TC and TCC peripherals:
  - Up to eight PWM outputs on each 24-bit TCC0 and TCC3
  - Up to four PWM outputs on 24-bit TCC1
  - Up to two PWM outputs on 16-bit TCC2
  - Up to two PWM outputs on each 16-bit TC
- Watchdog Timer (WDT) with Window mode

Advanced Analog and Touch

- One 12-bit 1 MSPS Analog-to-Digital Converter (ADC) with up to 24 channels
- Four Analog Comparators (AC) with window compare function
- Two 12-bit 1 MSPS Digital-to-Analog Converter (DAC) that can operate as:
  - Two independent DACs in Single-Ended mode
  - One single DAC in Differential mode
- Three Operational Amplifiers (OPAMP)
- One enhanced Peripheral Touch Controller (PTC):
  - Up to 32 self-capacitance channels
  - Up to 256 (16 x 16) mutual-capacitance channels
  - Low-power, high-sensitivity, environmentally robust capacitive touch buttons, sliders, wheels, and trackpads
  - Driven Shield Plus for better noise immunity and moisture tolerance
  - Parallel Acquisition (Boost Mode) through Polarity control
  - Hardware noise filtering and noise signal desynchronization for high conducted immunity
  - Supports wake-up on touch from Standby Sleep mode

Communication Interfaces

- USB Full Speed Device and Host
  - Crystal-less operation in Device mode using DFLL48M
  - Supports 8 IN endpoints and 8 OUT endpoints in Device mode, no endpoint size limitations
  - Supports 8 physical pipes in Host mode, no pipe size limitations
  - Integrated serial resistors
- Six Serial Communication Interfaces (SERCOM) that can operate as:
  - USART with full-duplex and single-wire half-duplex configuration
  - I^2C up to 3.4 Mbit/s (High-Speed mode)
  - Serial Peripheral Interface (SPI)
  - ISO7816
  - RS-485
  - LIN Host/Client
- One Inter-IC Sound Interface (I^2S) with up to 8-slot TDM and PDM microphone support

Clock Management

- Flexible clock distribution optimized for low power
- 32.768 kHz crystal oscillator (XOSC32K)
- 32.768 kHz ultra low-power internal RC oscillator (OSCULP32K)
- 0.4 to 32 MHz crystal oscillator (XOSC)
- 16/12/8/4 MHz low-power internal RC oscillator (OSC16M)
- 48 MHz digital Frequency-Locked Loop (DFLL48M)
- 32 MHz Ultra low-power digital Frequency-Locked Loop (DFLLULP)
- 32-96 MHz fractional digital Phase-Locked Loop (FDPLL96M)
- Clock Failure Detection on both crystal oscillators (CFD)
- One frequency meter (FREQM)

Input/Output (I/O)

- Up to 80 programmable I/O lines
- Sixteen external interrupts (EIC)
- One non-maskable interrupt (NMI)
- One Four-LUTs Configurable Custom Logic (CCL) that supports:
  - Combinatorial logic functions, such as AND, NAND, OR, and NOR
  - Sequential logic functions, such as Flip-Flop and Latches

Debugger Development Support

- Two-pin Serial Wire Debug (SWD) programming and debugging interface
- Four hardware breakpoints, two data watchpoints

Table 1. Packages

Type	VQFN(1)		TQFP(2)		
Pin Count	48	64	48	64	100
I/O Pins (up to)	34	48	34	48	80
Contact/Lead Pitch	0.5 mm	0.5 mm	0.5 mm	0.5 mm	0.5 mm
Dimensions Body	7x7x0.90 mm	9x9x1 mm	7x7x1 mm	10x10x1 mm	14x14x1 mm

Notes:

1. 48-pin and 64-pin VQFN packages are with wettable flanks.
2. 48-pin TQFP package is only available for PIC32CM LE00/LS00.

Table of Contents

1. Configuration Summary

Table 1-1. PIC32CM LE00/LS00/LS60 Family Features

Feature	PIC32CM LE00 Family	PIC32CM LS00 Family	PIC32CM LS60 Family
TrustZone for ARMv8-M	No	Yes	
MPU	1 with 12 regions	2 with 12 regions each (Secure/Non-Secure)	
SysTick Timers	1	2 (Secure/Non-Secure)	
ATECC608B CryptoAuthentication	No		Yes
ATECC608B-based secure boot	No		Yes
SHA- or HMAC-based secure boot	No	Yes	
DICE Security Standard Support	No	Yes	
Address and Data Scrambling	TrustRAM	TrustRAM, Data Flash	
Secure Pin Multiplexing (on SERCOM1)	No	Yes	No
Crypto Accelerators	No	Yes	
TRNG	Yes		
Debug Access Levels (DAL)	2	3	
DMA Channels	16		
Event System Channels	12		
CRC	Yes		
External Interrupt Lines/NMI	16/1		
Brown-out Detection	VDD, VDDCORE and VDDPLL		
USB FS/LS Host & Device	Yes		
SERCOM (Serial Communication)	6	6	6 (SERCOM1 reserved for ATECC608B)
I^2S	1		
TC (PWM Outputs)	3x TC (Up to 6 PWM Outputs)		
TCC (PWM Outputs)	3 x 24-bit TCC (Up to 20 PWM Outputs) / 1 x 16-bit TCC (Up to 2 PWM Outputs)		
RTC	1		
WDT (Watchdog)	1		
ADC	1		
DAC	2 independent DACs in Single-Ended mode / 1 single DAC in Differential mode		
AC (Analog Comparator)	4		
OPAMP	3		
PTC (Peripheral Touch Controller)	1		
CCL (Configurable Custom Logic)	1 (4 LUTs)		
FREQM (Frequency Meter)	1		

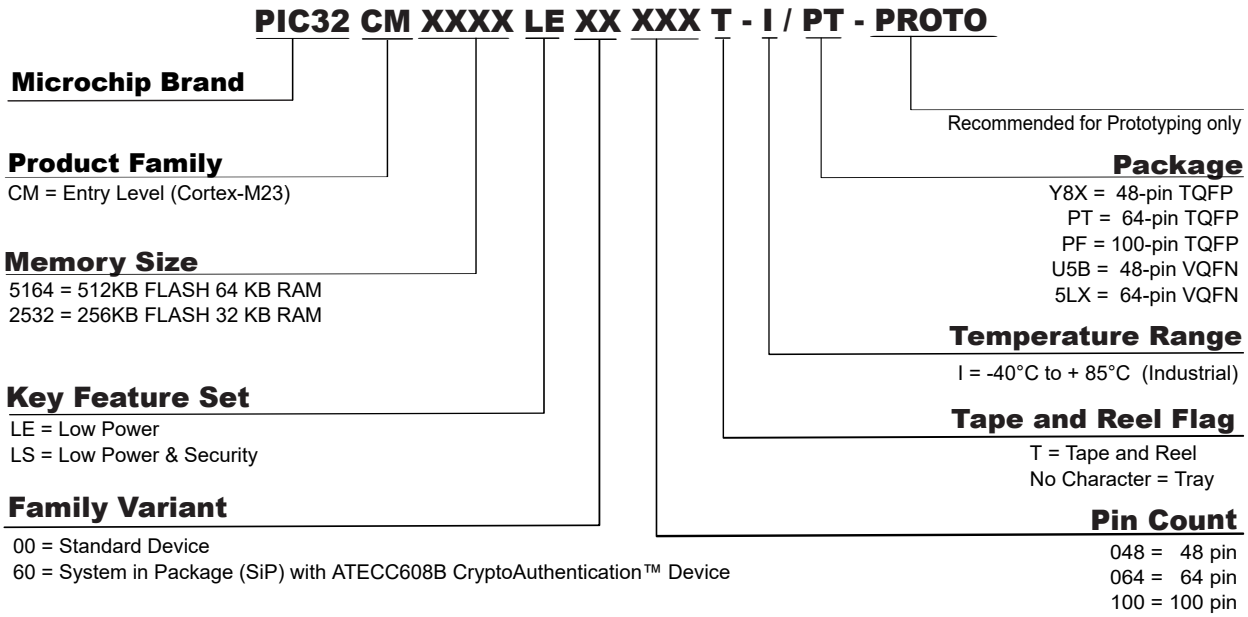
Table 1-2. PIC32CM LE00/LS00/LS60 Device-specific Features (1,2,3)

Device	Packages	Flash + Data Flash Memory (KB)	SRAM (KB)	TrustRAM (Bytes)	Pins	I/O Pins	SERCOM USART/ SPI /I²C	ADC Channels	Analog Comparators Inputs	OPAMP Outputs	EIC External Interrupts	CCL Inputs	GCLK I/Os	PTC Self-capacitance/ Mutual-capacitance Channels	Tamper Pins (Inputs/ Outputs)
PIC32CM2532LE00048	VQFN,TQFP	256+8	32	512	48	34	6/6/4	14	6	2	12	9	6	26/169	4/4
PIC32CM2532LS00048															
PIC32CM5164LE00048	VQFN,TQFP	512+16	64												
PIC32CM5164LS00048															
PIC32CM5164LS60048	VQFN														
PIC32CM2532LE00064	VQFN, TQFP	256+8	32	512	64	48	6/6/6	20	8	3	14	12	8	32/256	6/6
PIC32CM2532LS00064															
PIC32CM5164LE00064		512+16	64												
PIC32CM5164LS00064															
PIC32CM5164LS60064															
PIC32CM2532LE00100	TQFP	256+8	32	512	100	80	6/6/6	24	8	3	16	12	8	32/256	8/8
PIC32CM2532LS00100															
PIC32CM5164LE00100		512+16	64												
PIC32CM5164LS00100															
PIC32CM5164LS60100															

Notes:

1. I²C is not supported on all SERCOM pins. Refer to the “SERCOM I²C Peripheral” chapter for the list of supported features for each peripheral instance.
2. For the PIC32CM LS60 family, SERCOM1 is reserved for the ATECC608B.
3. This table only details the specifics of all variants.

2. Ordering Information



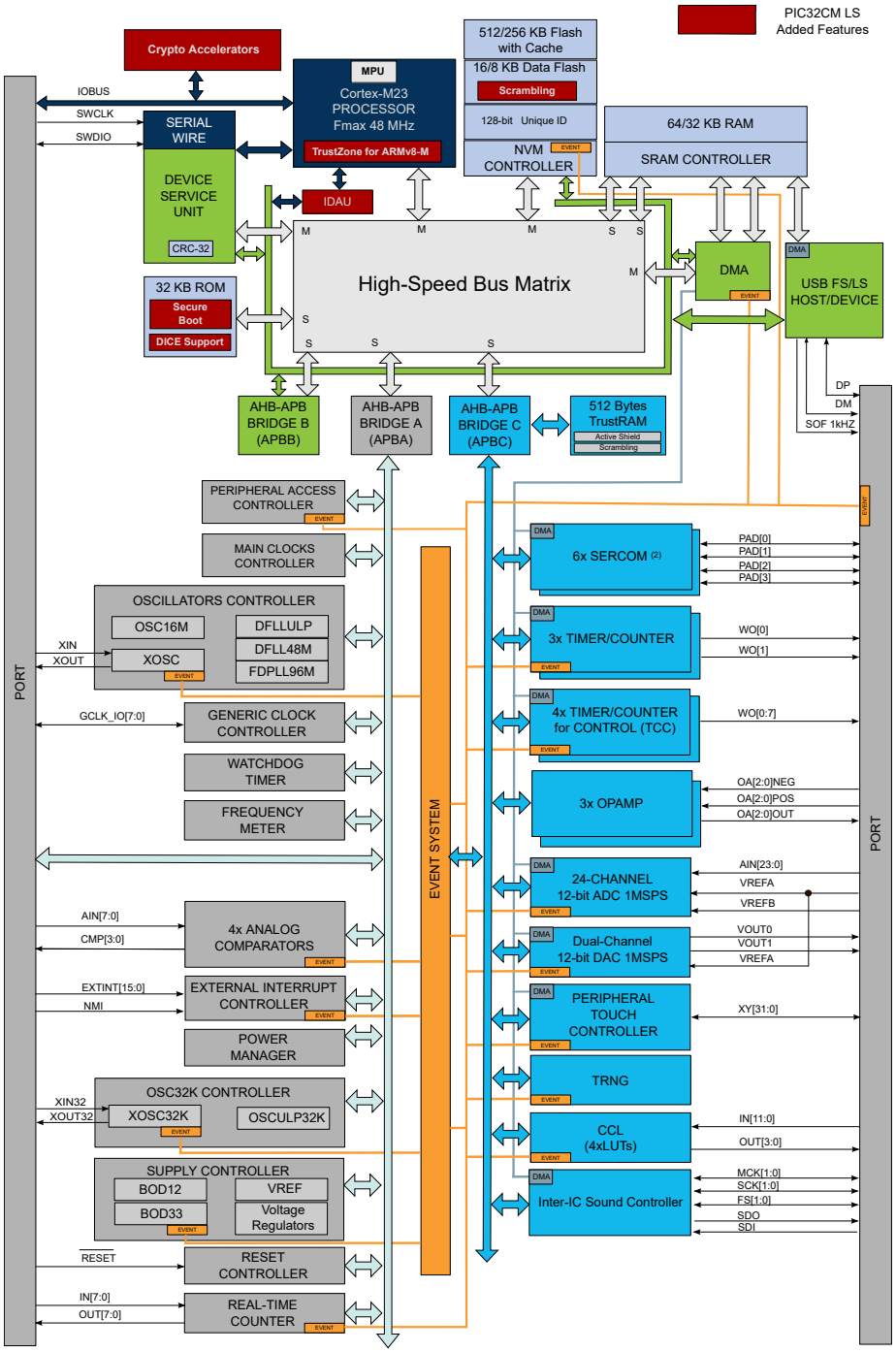
Notes:

1. Devices can be factory programmed with securely key provisioned software. Contact your local Microchip sales office for more information.
2. Packages U5B (48-pin VQFN) and 5LX (64-pin VQFN) are with wettable flanks.
3. PROTO ordering code extension is only applicable for PIC32CM LS60 Family. This enables prototyping the ATECC608B application use case and developing the custom provisioning data-set. PROTO is an ordering code extension and will not be printed onto the package marking.

3. Block Diagram

The following figure illustrates the PIC32CM LE00/LS00/LS60 block diagram.

Figure 3-1. PIC32CM LE00/LS00/LS60 Block Diagram



Notes:

1. Number of peripheral instances, channels, input/output pins can differ between the different packages.
2. SERCOM1 is reserved to ATECC608B interconnection for the PIC32CM LS60 family.

4. Pinout and Packaging

Each pin is controlled by the I/O Pin Controller (PORT) as a general purpose I/O and alternatively can be assigned to one of the peripheral functions: A, B, C, D, E, G, H, I, J, or K.

The following tables describe the peripheral signals multiplexed to the PORT I/O pins for each package.

⚠ CAUTION I/Os for SERCOM and I^2S peripherals are grouped into I/O sets, listed in their 'IOSET' column. For these peripherals, it is mandatory to use I/Os that belong to the same I/O set. The peripheral's timings are not guaranteed when I/Os from different I/O sets are mixed.

The column "Reset State" indicates the reset state of the line with mnemonics:

- "I/O" or "Peripheral Function" indicates whether the I/O pin resets in I/O mode or in peripheral function mode.
- "I"/"O"/"Hi-Z" indicates whether the I/O is configured as an input, output, or is tri-stated.
- "PU"/"PD" indicates whether pull up, pull down, or nothing is enabled.

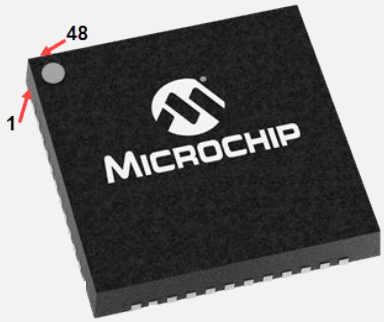
Note: The schematic checklist chapter provides the user with the requirements regarding the different pin connections that must be considered before starting any new board design as well as information on the minimum hardware resources required to quickly develop an application.

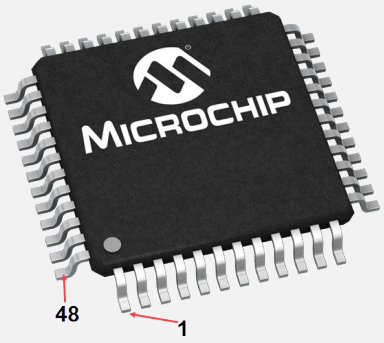
4.1 48-pin VQFN and 48-pin TQFP

48-PIN VQFN and TQFP (Top View)(1)(2)

PIC32CM2532(LE00/LS00)048

PIC32CM5164(LE00/LS00/LS60)048





Notes:

1. The 48-pin VQFN package is with wettable flanks.
2. The 48-pin TQFP package is only available for PIC32CM LE00/LS00.

Table 4-1. 48-pin VQFN/48-pin TQFP I/O PINMUX

Pin		A	B[1,3]						C[4][5]		D[4][5]		E[2]	G	H	I	J[2]	K			
VQFN48 / TQFP48	Pin Name	EIC	REF	ADC	AC	PTC	DAC	OPAMP	SERCOM	IOSET	SERCOM ALT	IOSET	TC / TCC	RTC / USB / Debug	AC / GCLK	CCL	TCC	I2S	IOSET	Power Rail	Reset State
1	PA00 / XIN32	EXTINT[0]				XY[0]		OANEG[1]			SERCOM1/ PAD[0]	6	TCC2/ WO[0]							AVDD	I/O, Hi-Z
2	PA01 / XOUT32	EXTINT[1]				XY[1]		OAPOS[1]			SERCOM1/ PAD[1]	6	TCC2/ WO[1]							AVDD	I/O, Hi-Z
3	PA02[3][6]	EXTINT[2]		AIN[0]		XY[2]	VOUT0	OANEG[0]			SERCOM0/ PAD[2]	3					TCC3/ WO[0]			AVDD	I/O, Hi-Z
4	PA03	EXTINT[3]	VREFA	AIN[1]		XY[3]		OANEG[2]			SERCOM0/ PAD[3]	3					TCC3/ WO[1]			AVDD	I/O, Hi-Z
5	AVSS																				
6	AVDD																				
7	PB08	EXTINT[8]		AIN[18]		XY[20]					SERCOM3/ PAD[0]	4	TC0/ WO[0]			IN[8]	TCC3/ WO[6]			AVDD	I/O, Hi-Z
8	PB09	EXTINT[9]		AIN[19]		XY[21]					SERCOM3/ PAD[1]	4	TC0/ WO[1]			OUT[2]	TCC3/ WO[7]			AVDD	I/O, Hi-Z
9	PA04	EXTINT[4]	VREFB	AIN[2]	AIN[0]			OAOUT[2]			SERCOM0/ PAD[0]	3	TCC0/ WO[0]			IN[0]	TCC3/ WO[2]			AVDD	I/O, Hi-Z
10	PA05	EXTINT[5]		AIN[3]	AIN[1]	XY[4]		OAPOS[2]			SERCOM0/ PAD[1]	3	TCC0/ WO[1]			IN[1]	TCC3/ WO[3]			AVDD	I/O, Hi-Z
11	PA06	EXTINT[6]		AIN[4]	AIN[2]	XY[5]		OAPOS[0]			SERCOM0/ PAD[2]	3	TCC1/ WO[0]			IN[2]	TCC3/ WO[4]			AVDD	I/O, Hi-Z
12	PA07[3]	EXTINT[7]		AIN[5]	AIN[3]		VOUT1	OAOUT[0]			SERCOM0/ PAD[3]	3	TCC1/ WO[1]			OUT[0]	TCC3/ WO[5]			AVDD	I/O, Hi-Z
13	PA08	NMI		AIN[6]		XY[6]			SERCOM1/ PAD[0]	1	SERCOM2/ PAD[0]	2	TCC0/ WO[0]	RTC/ IN[0]		IN[3]	TCC1/ WO[2]	SDI	1	VDD	I/O, Hi-Z
14	PA09	EXTINT[0]		AIN[7]		XY[7]			SERCOM1/ PAD[1]	1	SERCOM2/ PAD[1]	2	TCC0/ WO[1]	RTC/ IN[1]		IN[4]	TCC1/ WO[3]	MCK[0]	1,2	VDD	I/O, Hi-Z
15	PA10	EXTINT[1]		AIN[8]		XY[8]			SERCOM1/ PAD[2]	1, 6	SERCOM2/ PAD[2]	2	TCC1/ WO[0]		GCLK/ IO[4]	IN[5]	TCC0/ WO[2]	SCK[0]	1	VDD	I/O, Hi-Z
16	PA11	EXTINT[2]		AIN[9]		XY[9]			SERCOM1/ PAD[3]	1, 6	SERCOM2/ PAD[3]	2	TCC1/ WO[1]		GCLK/ IO[3]	OUT[1]	TCC0/ WO[3]	FS[0]	2	VDD	I/O, Hi-Z
17	AVSSPLL																				
18	VDD																				
19	VSS																				
20	VDDPLL																				
21	PA12	EXTINT[12]				XY[24]			SERCOM2/ PAD[0]	1	SERCOM4/ PAD[0]	2	TCC2/ WO[0]		AC/ CMP[0]		TCC0/ WO[6]			VDD	I/O, Hi-Z

.....continued

Pin	Pin Name	A	B(1,3)						C(4)(5)		D(4)(5)		E(2)	G	H	I	J(2)	K		Power Rail	Reset State
VQFN48 / TQFP48		EIC	REF	ADC	AC	PTC	DAC	OPAMP	SERCOM	IOSET	SERCOM ALT	IOSET	TC / TCC	RTC / USB / Debug	AC / GCLK	CCL	TCC	I2S	IOSET		
22	PA13	EXTINT[13]				XY[25]			SERCOM2/ PAD[1]	1	SERCOM4/ PAD[1]	2	TCC2/ WO[1]		AC/ CMP[1]		TCC0/ WO[7]			VDD	I/O, Hi-Z
23	PA14 / XIN	EXTINT[3]				XY[10]			SERCOM2/ PAD[2]	1	SERCOM0/ PAD[2]	4	TC0/ WO[0]		GCLK/ IO[0]		TCC0/ WO[4]			VDD	I/O, Hi-Z
24	PA15 / XOUT	EXTINT[4]				XY[11]			SERCOM2/ PAD[3]	1	SERCOM0/ PAD[3]	4	TC0/ WO[1]		GCLK/ IO[1]		TCC0/ WO[5]			VDD	I/O, Hi-Z
25	PA16	EXTINT[5]				XY[12]			SERCOM1/ PAD[0]	3	SERCOM0/ PAD[0]	4	TCC2/ WO[0]	RTC/ IN[2]	GCLK/ IO[2]	IN[0]	TCC0/ WO[6]			VDD	I/O, Hi-Z
26	PA17	EXTINT[6]				XY[13]			SERCOM1/ PAD[1]	3	SERCOM0/ PAD[1]	4	TCC2/ WO[1]	RTC/ IN[3]	GCLK/ IO[3]	IN[1]	TCC0/ WO[7]			VDD	I/O, Hi-Z
27	PA18	EXTINT[7]				XY[14]			SERCOM1/ PAD[2]	3	SERCOM0/ PAD[2]	4	TC2/ WO[0]	RTC/ OUT[0]	AC/ CMP[0]	IN[2]	TCC0/ WO[2]	FS[0]	1	VDD	I/O, Hi-Z
28	PA19	EXTINT[0]				XY[15]			SERCOM1/ PAD[3]	3	SERCOM0/ PAD[3]	4	TC2/ WO[1]	RTC/ OUT[1]	AC/ CMP[1]	OUT[0]	TCC0/ WO[3]	SDO	1,2	VDD	I/O, Hi-Z
29	PA20	EXTINT[4]				XY[22]			SERCOM3/ PAD[2]	3	SERCOM2/ PAD[2]	3	TC1/ WO[0]		GCLK/ IO[4]	IN[5]	TCC0/ WO[6]	SCK[0]	2	VDD	I/O, Hi-Z
30	PA21	EXTINT[5]				XY[23]			SERCOM3/ PAD[3]	3	SERCOM2/ PAD[3]	3			GCLK/ IO[5]		TCC0/ WO[7]	SCK[1]	1,2	VDD	I/O, Hi-Z
31	PA22	EXTINT[1]				XY[16]			SERCOM0/ PAD[0]	1	SERCOM2/ PAD[0]	3	TC0/ WO[0]	RTC/ OUT[2]	GCLK/ IO[2]	IN[6]	TCC0/ WO[4]			VDD	I/O, Hi-Z
32	PA23	EXTINT[2]				XY[17]			SERCOM0/ PAD[1]	1	SERCOM2/ PAD[1]	3	TC0/ WO[1]	RTC/ OUT[3]	GCLK/ IO[1]	IN[7]	TCC0/ WO[5]			VDD	I/O, Hi-Z
33	PA24	EXTINT[3]							SERCOM0/ PAD[2]	1	SERCOM2/ PAD[2]	3	TC1/ WO[0]	USB/ DM	AC/ CMP[2]	IN[8]	TCC1/ WO[2]			VDD	I/O, Hi-Z
34	PA25	EXTINT[4]							SERCOM0/ PAD[3]	1	SERCOM2/ PAD[3]	3	TC1/ WO[1]	USB/ DP	AC/ CMP[3]	OUT[2]	TCC1/ WO[3]			VDD	I/O, Hi-Z
35	VSS																				
36	VDD																				
37	PB22	EXTINT[6]							SERCOM0/ PAD[2]	1	SERCOM5/ PAD[2]	4		USB/ SOF_1KHZ	GCLK/ IO[0]	IN[0]	TCC3/ WO[0]			VDD	I/O, Hi-Z
38	PB23	EXTINT[7]							SERCOM0/ PAD[3]	1	SERCOM5/ PAD[3]	4			GCLK/ IO[1]	OUT[0]	TCC3/ WO[1]			VDD	I/O, Hi-Z
39	VSS																				
40	$\overline{\text{RESET}}$																			VDD	I, PU
41	VDDCORE																				
42	VSSCORE																				
43	VDDOUT																				

.....continued

Pin	Pin Name	A	B[1,3]						C[4][5]		D[4][5]		E[2]	G	H	I	J[2]	K		Power Rail	Reset State
VQFN48 / TQFP48		EIC	REF	ADC	AC	PTC	DAC	OPAMP	SERCOM	IOSET	SERCOM ALT	IOSET	TC / TCC	RTC / USB / Debug	AC / GCLK	CCL	TCC	I2S	IOSET		
44	VDD																				
45	PA30 / SWCLK	EXTINT[6]				XY[18]							TCC1/ WO[0]	SWCLK	GCLK/ IO[0]	IN[3]	TCC3/ WO[4]			VDD	SWCLK, I, PU
46	PA31 / SWDIO	EXTINT[7]				XY[19]							TCC1/ WO[1]			OUT[1]	TCC3/ WO[5]			VDD	I/O, Hi-Z
47	PB02	EXTINT[2]		AIN[10]	AIN[4]				SERCOM3/ PAD[0]	3	SERCOM5/ PAD[0]	4	TC2/ WO[0]			OUT[0]	TCC3/ WO[2]			VDD	I/O, Hi-Z
48	PB03[6]	EXTINT[3]		AIN[11]	AIN[5]				SERCOM3/ PAD[1]	3	SERCOM5/ PAD[1]	4	TC2/ WO[1]				TCC3/ WO[3]			VDD	I/O, Hi-Z

Notes:

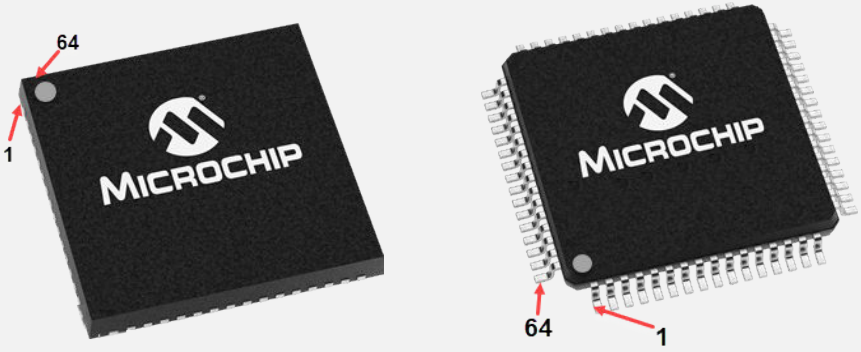
1. All analog pin functions are on the peripheral function B. The peripheral function B must be selected to disable the digital control of the pin.
2. Refer to TCC peripheral chapter for the list of supported features for each peripheral instance.
3. When DAC0 or DAC1 channel is enabled, respective VOUT0 (PA02) or VOUT1 (PA07) pins cannot be used for other purposes (excluding analog inputs).
4. I²C is not supported on all SERCOM pins. Refer to SERCOM I²C peripheral chapter for the list of supported features for each peripheral instance.
5. SERCOM1 signals are not available for the PIC32CM LS60 family as reserved for ATECC608B interconnection.
6. The transition time of the following pins must be greater than 50µs in order to not affect the XOSC32 cycle to cycle jitter.

4.2 64-pin VQFN and 64-pin TQFP

64-PIN VQFN and TQFP (Top View)(1)

PIC32CM2532(LE00/LS00)064

PIC32CM5164(LE00/LS00/LS60)064



Note:

1. The 64-pin VQFN package is with wettable flanks.

Table 4-2. 64-pin VQFN/64-pin TQFP PINMUX

Pin		A	B[1,3]						C[4,5]		D[4,5]		E[2]	G	H	I	J[2]	K			
VQFN64 / TQFP64	Pin Name	EIC	REF	ADC	AC	PTC	DAC	OPAMP	SERCOM	IOSET	SERCOM ALT	IOSET	TC / TCC	RTC / USB / Debug	AC / GCLK	CCL	TCC	I2S	IOSET	Power Rail	Reset State
1	PA00 / XIN32	EXTINT[0]				XY[0]		OANEG[1]			SERCOM1/ PAD[0]	6	TCC2/ WO[0]							AVDD	I/O, Hi-Z
2	PA01 / XOUT32	EXTINT[1]				XY[1]		OAPOS[1]			SERCOM1/ PAD[1]	6	TCC2/ WO[1]							AVDD	I/O, Hi-Z
3	PA02(3)(6)	EXTINT[2]		AIN[0]		XY[2]	VOUT0	OANEG[0]			SERCOM0/ PAD[2]	3					TCC3/ WO[0]			AVDD	I/O, Hi-Z
4	PA03	EXTINT[3]	VREFA	AIN[1]		XY[3]		OANEG[2]			SERCOM0/ PAD[3]	3					TCC3/ WO[1]			AVDD	I/O, Hi-Z
5	PB04	EXTINT[4]		AIN[12]	AIN[6]	XY[26]		OAOUT[1]												AVDD	I/O, Hi-Z
6	PB05	EXTINT[5]		AIN[13]	AIN[7]	XY[27]														AVDD	I/O, Hi-Z
7	AVSS																				
8	AVDD																				
9	PB06	EXTINT[6]		AIN[14]		XY[28]										IN[6]				AVDD	I/O, Hi-Z
10	PB07	EXTINT[7]		AIN[15]		XY[29]										IN[7]				AVDD	I/O, Hi-Z
11	PB08	EXTINT[8]		AIN[18]		XY[20]					SERCOM3/ PAD[0]	4	TC0/ WO[0]			IN[8]	TCC3/ WO[6]			AVDD	I/O, Hi-Z
12	PB09	EXTINT[9]		AIN[19]		XY[21]					SERCOM3/ PAD[1]	4	TC0/ WO[1]			OUT[2]	TCC3/ WO[7]			AVDD	I/O, Hi-Z
13	PA04	EXTINT[4]	VREFB	AIN[2]	AIN[0]			OAOUT[2]			SERCOM0/ PAD[0]	3	TCC0/ WO[0]			IN[0]	TCC3/ WO[2]			AVDD	I/O, Hi-Z
14	PA05	EXTINT[5]		AIN[3]	AIN[1]	XY[4]		OAPOS[2]			SERCOM0/ PAD[1]	3	TCC0/ WO[1]			IN[1]	TCC3/ WO[3]			AVDD	I/O, Hi-Z
15	PA06	EXTINT[6]		AIN[4]	AIN[2]	XY[5]		OAPOS[0]			SERCOM0/ PAD[2]	3	TCC1/ WO[0]			IN[2]	TCC3/ WO[4]			AVDD	I/O, Hi-Z
16	PA07(3)	EXTINT[7]		AIN[5]	AIN[3]		VOUT1	OAOUT[0]			SERCOM0/ PAD[3]	3	TCC1/ WO[1]			OUT[0]	TCC3/ WO[5]			AVDD	I/O, Hi-Z
17	PA08	NMI		AIN[6]		XY[6]			SERCOM1/ PAD[0]	1	SERCOM2/ PAD[0]	2	TCC0/ WO[0]	RTC/IN[0]		IN[3]	TCC1/ WO[2]	SDI	1	VDD	I/O, Hi-Z
18	PA09	EXTINT[0]		AIN[7]		XY[7]			SERCOM1/ PAD[1]	1	SERCOM2/ PAD[1]	2	TCC0/ WO[1]	RTC/IN[1]		IN[4]	TCC1/ WO[3]	MCK[0]	1,2	VDD	I/O, Hi-Z
19	PA10	EXTINT[1]		AIN[8]		XY[8]			SERCOM1/ PAD[2]	1, 6	SERCOM2/ PAD[2]	2	TCC1/ WO[0]		GCLK/ IO[4]	IN[5]	TCC0/ WO[2]	SCK[0]	1	VDD	I/O, Hi-Z
20	PA11	EXTINT[2]		AIN[9]		XY[9]			SERCOM1/ PAD[3]	1, 6	SERCOM2/ PAD[3]	2	TCC1/ WO[1]		GCLK/ IO[3]	OUT[1]	TCC0/ WO[3]	FS[0]	2	VDD	I/O, Hi-Z
21	AVSSPLL																				
22	VDD																				

.....continued

Pin		A	B(1,3)						C(4,5)		D(4,5)		E(2)	G	H	I	J(2)	K			
VQFN64 / TQFP64	Pin Name	EIC	REF	ADC	AC	PTC	DAC	OPAMP	SERCOM	IOSET	SERCOM ALT	IOSET	TC / TCC	RTC / USB / Debug	AC / GCLK	CCL	TCC	I2S	IOSET	Power Rail	Reset State
23	VSS																				
24	VDDPLL																				
25	PB12	EXTINT[12]				XY[30]			SERCOM3/ PAD[0]	1			TC0/ WO[0]		GCLK/ IO[6]		TCC0/ WO[6]	FS[1]	1,2	VDD	I/O, Hi-Z
26	PB13	EXTINT[13]				XY[31]			SERCOM3/ PAD[1]	1			TC0/ WO[1]		GCLK/ IO[7]		TCC0/ WO[7]	MCK[1]	1,2	VDD	I/O, Hi-Z
27	PB14	EXTINT[14]							SERCOM3/ PAD[2]	1, 4			TC1/ WO[0]	RTC/IN[4]	GCLK/ IO[0]	IN[9]				VDD	I/O, Hi-Z
28	PB15	EXTINT[15]							SERCOM3/ PAD[3]	1, 4			TC1/ WO[1]	RTC/IN[5]	GCLK/ IO[1]	IN[10]				VDD	I/O, Hi-Z
29	PA12	EXTINT[12]				XY[24]			SERCOM2/ PAD[0]	1	SERCOM4/ PAD[0]	2	TCC2/ WO[0]		AC/CMP[0]		TCC0/ WO[6]			VDD	I/O, Hi-Z
30	PA13	EXTINT[13]				XY[25]			SERCOM2/ PAD[1]	1	SERCOM4/ PAD[1]	2	TCC2/ WO[1]		AC/CMP[1]		TCC0/ WO[7]			VDD	I/O, Hi-Z
31	PA14 / XIN	EXTINT[3]				XY[10]			SERCOM2/ PAD[2]	1	SERCOM0/ PAD[2]	4	TC0/ WO[0]		GCLK/ IO[0]		TCC0/ WO[4]			VDD	I/O, Hi-Z
32	PA15 / XOUT	EXTINT[4]				XY[11]			SERCOM2/ PAD[3]	1	SERCOM0/ PAD[3]	4	TC0/ WO[1]		GCLK/ IO[1]		TCC0/ WO[5]			VDD	I/O, Hi-Z
33	VSS																				
34	VDD																				
35	PA16	EXTINT[5]				XY[12]			SERCOM1/ PAD[0]	3	SERCOM0/ PAD[0]	4	TCC2/ WO[0]	RTC/IN[2]	GCLK/ IO[2]	IN[0]	TCC0/ WO[6]			VDD	I/O, Hi-Z
36	PA17	EXTINT[6]				XY[13]			SERCOM1/ PAD[1]	3	SERCOM0/ PAD[1]	4	TCC2/ WO[1]	RTC/IN[3]	GCLK/ IO[3]	IN[1]	TCC0/ WO[7]			VDD	I/O, Hi-Z
37	PA18	EXTINT[7]				XY[14]			SERCOM1/ PAD[2]	3	SERCOM0/ PAD[2]	4	TC2/ WO[0]	RTC/OUT[0]	AC/CMP[0]	IN[2]	TCC0/ WO[2]	FS[0]	1	VDD	I/O, Hi-Z
38	PA19	EXTINT[0]				XY[15]			SERCOM1/ PAD[3]	3	SERCOM0/ PAD[3]	4	TC2/ WO[1]	RTC/OUT[1]	AC/CMP[1]	OUT[0]	TCC0/ WO[3]	SDO	1,2	VDD	I/O, Hi-Z
39	PB16	EXTINT[0]							SERCOM5/ PAD[0]	1			TC2/ WO[0]	RTC/OUT[4]	GCLK/ IO[2]	IN[11]	TCC0/ WO[4]	SDI	2	VDD	I/O, Hi-Z
40	PB17	EXTINT[1]							SERCOM5/ PAD[1]	1			TC2/ WO[1]	RTC/OUT[5]	GCLK/ IO[3]	OUT[3]	TCC0/ WO[5]			VDD	I/O, Hi-Z
41	PA20	EXTINT[4]				XY[22]			SERCOM3/ PAD[2]	3	SERCOM2/ PAD[2]	3	TC1/ WO[0]		GCLK/ IO[4]	IN[5]	TCC0/ WO[6]	SCK[0]	2	VDD	I/O, Hi-Z
42	PA21	EXTINT[5]				XY[23]			SERCOM3/ PAD[3]	3	SERCOM2/ PAD[3]	3			GCLK/ IO[5]		TCC0/ WO[7]	SCK[1]	1,2	VDD	I/O, Hi-Z
43	PA22	EXTINT[1]				XY[16]			SERCOM0/ PAD[0]	1	SERCOM2/ PAD[0]	3	TC0/ WO[0]	RTC/OUT[2]	GCLK/ IO[2]	IN[6]	TCC0/ WO[4]			VDD	I/O, Hi-Z

.....continued

Pin		A	B(1,3)						C(4,5)		D(4,5)		E(2)	G	H	I	J(2)	K			
VQFN64 / TQFP64	Pin Name	EIC	REF	ADC	AC	PTC	DAC	OPAMP	SERCOM	IOSET	SERCOM ALT	IOSET	TC / TCC	RTC / USB / Debug	AC / GCLK	CCL	TCC	I2S	IOSET	Power Rail	Reset State
44	PA23	EXTINT[2]				XY[17]			SERCOM0/ PAD[1]	1	SERCOM2/ PAD[1]	3	TC0/ WO[1]	RTC/OUT[3]	GCLK/ IO[1]	IN[7]	TCC0/ WO[5]			VDD	I/O, Hi-Z
45	PA24	EXTINT[3]							SERCOM0/ PAD[2]	1	SERCOM2/ PAD[2]	3	TC1/ WO[0]	USB/DM	AC/CMP[2]	IN[8]	TCC1/ WO[2]			VDD	I/O, Hi-Z
46	PA25	EXTINT[4]							SERCOM0/ PAD[3]	1	SERCOM2/ PAD[3]	3	TC1/ WO[1]	USB/DP	AC/CMP[3]	OUT[2]	TCC1/ WO[3]			VDD	I/O, Hi-Z
47	VSS																				
48	VDD																				
49	PB22	EXTINT[6]							SERCOM0/ PAD[2]	1	SERCOM5/ PAD[2]	4		USB/ SOF_1KHZ	GCLK/ IO[0]	IN[0]	TCC3/ WO[0]			VDD	I/O, Hi-Z
50	PB23	EXTINT[7]							SERCOM0/ PAD[3]	1	SERCOM5/ PAD[3]	4			GCLK/ IO[1]	OUT[0]	TCC3/ WO[1]			VDD	I/O, Hi-Z
51	VSS																				
52	RESET																			VDD	I, PU
53	VDDCORE																				
54	VSSCORE																				
55	VDDOUT																				
56	VDD																				
57	PA30 / SWCLK	EXTINT[6]				XY[18]					SERCOM1/ PAD[2]	4	TCC1/ WO[0]	SWCLK	GCLK/ IO[0]	IN[3]	TCC3/ WO[4]			VDD	SWCLK, I, PU
58	PA31 / SWDIO	EXTINT[7]				XY[19]					SERCOM1/ PAD[3]	4	TCC1/ WO[1]			OUT[1]	TCC3/ WO[5]			VDD	I/O, Hi-Z
59	PB30	EXTINT[14]							SERCOM1/ PAD[0]	4	SERCOM5/ PAD[0]	3	TCC0/ WO[0]				TCC1/ WO[2]			VDD	I/O, Hi-Z
60	PB31	EXTINT[15]							SERCOM1/ PAD[1]	4	SERCOM5/ PAD[1]	3	TCC0/ WO[1]				TCC1/ WO[3]			VDD	I/O, Hi-Z
61	PB00	EXTINT[0]		AIN[16]					SERCOM3/ PAD[2]	3	SERCOM5/ PAD[2]	3, 4				IN[1]				VDD	I/O, Hi-Z
62	PB01	EXTINT[1]		AIN[17]					SERCOM3/ PAD[3]	3	SERCOM5/ PAD[3]	3, 4				IN[2]				VDD	I/O, Hi-Z
63	PB02	EXTINT[2]		AIN[10]	AIN[4]				SERCOM3/ PAD[0]	3	SERCOM5/ PAD[0]	4	TC2/ WO[0]			OUT[0]	TCC3/ WO[2]			VDD	I/O, Hi-Z
64	PB03(6)	EXTINT[3]		AIN[11]	AIN[5]				SERCOM3/ PAD[1]	3	SERCOM5/ PAD[1]	4	TC2/ WO[1]				TCC3/ WO[3]			VDD	I/O, Hi-Z

.....continued

Pin	Pin Name	A	B(1,3)						C(4,5)		D(4,5)		E(2)	G	H	I	J(2)	K		Power Rail	Reset State
VQFN64 / TQFP64		EIC	REF	ADC	AC	PTC	DAC	OPAMP	SERCOM	IOSET	SERCOM ALT	IOSET	TC / TCC	RTC / USB / Debug	AC / GCLK	CCL	TCC	I2S	IOSET		

Notes:

1. All analog pin functions are on the peripheral function B. The peripheral function B must be selected to disable the digital control of the pin.
2. Refer to TCC peripheral chapter for the list of supported features for each peripheral instance.
3. When DAC0 or DAC1 channel is enabled, respective VOUT0 (PA02) or VOUT1 (PA07) pins cannot be used for other purposes (excluding analog inputs).
4. I²C is not supported on all SERCOM pins. Refer to SERCOM I²C peripheral chapter for the list of supported features for each peripheral instance.
5. SERCOM1 signals are not available for the PIC32CM LS60 family as reserved for ATECC608B interconnection.
6. The transition time of the following pins must be greater than 50µs in order to not affect the XOSC32 cycle to cycle jitter.

4.3 100-pin TQFP

100-PIN TQFP (Top View)

PIC32CM2532(LE00/LS00)100

PIC32CM5164(LE00/LS00/LS60)100

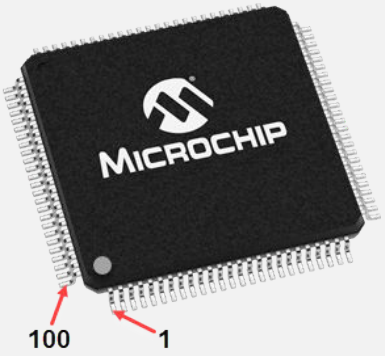


Table 4-3. 100-pin TQFP PINMUX

Pin		A	B (1,3)						C(4,5)		D(4,5)		E(2)	G	H	I	J(2)	K			
TQFP100	Pin Name	EIC	REF	ADC	AC	PTC	DAC	OPAMP	SERCOM	IOSET	SERCOM ALT	IOSET	TC / TCC	RTC / USB / Debug	AC / GCLK	CCL	TCC	I2S	IOSET	Power Rail	Reset State
1	PA00 / XIN32	EXTINT[0]				XY[0]		OANEG[1]			SERCOM1/ PAD[0]	6	TCC2/ WO[0]							AVDD	I/O, Hi-Z
2	PA01 / XOUT32	EXTINT[1]				XY[1]		OAPOS[1]			SERCOM1/ PAD[1]	6	TCC2/ WO[1]							AVDD	I/O, Hi-Z
3	PC00(6)	EXTINT[8]		AIN[20]																AVDD	I/O, Hi-Z
4	PC01	EXTINT[9]		AIN[21]																AVDD	I/O, Hi-Z
5	PC02	EXTINT[10]		AIN[22]																AVDD	I/O, Hi-Z
6	PC03	EXTINT[11]		AIN[23]																AVDD	I/O, Hi-Z
7	PA02(3)	EXTINT[2]		AIN[0]		XY[2]	VOUT0	OANEG[0]			SERCOM0/ PAD[2]	3					TCC3/ WO[0]			AVDD	I/O, Hi-Z
8	PA03	EXTINT[3]	VREFA	AIN[1]		XY[3]		OANEG[2]			SERCOM0/ PAD[3]	3					TCC3/ WO[1]			AVDD	I/O, Hi-Z
9	PB04	EXTINT[4]		AIN[12]	AIN[6]	XY[26]		OAOUT[1]												AVDD	I/O, Hi-Z
10	PB05	EXTINT[5]		AIN[13]	AIN[7]	XY[27]														AVDD	I/O, Hi-Z
11	AVSS																				
12	AVDD																				
13	PB06	EXTINT[6]		AIN[14]		XY[28]										IN[6]				AVDD	I/O, Hi-Z
14	PB07	EXTINT[7]		AIN[15]		XY[29]										IN[7]				AVDD	I/O, Hi-Z
15	PB08	EXTINT[8]		AIN[18]		XY[20]					SERCOM3/ PAD[0]	4	TC0/ WO[0]			IN[8]	TCC3/ WO[6]			AVDD	I/O, Hi-Z
16	PB09	EXTINT[9]		AIN[19]		XY[21]					SERCOM3/ PAD[1]	4	TC0/ WO[1]			OUT[2]	TCC3/ WO[7]			AVDD	I/O, Hi-Z
17	PA04	EXTINT[4]	VREFB	AIN[2]	AIN[0]			OAOUT[2]			SERCOM0/ PAD[0]	3	TCC0/ WO[0]			IN[0]	TCC3/ WO[2]			AVDD	I/O, Hi-Z
18	PA05	EXTINT[5]		AIN[3]	AIN[1]	XY[4]		OAPOS[2]			SERCOM0/ PAD[1]	3	TCC0/ WO[1]			IN[1]	TCC3/ WO[3]			AVDD	I/O, Hi-Z
19	PA06	EXTINT[6]		AIN[4]	AIN[2]	XY[5]		OAPOS[0]			SERCOM0/ PAD[2]	3	TCC1/ WO[0]			IN[2]	TCC3/ WO[4]			AVDD	I/O, Hi-Z
20	PA07(3)	EXTINT[7]		AIN[5]	AIN[3]		VOUT1	OAOUT[0]			SERCOM0/ PAD[3]	3	TCC1/ WO[1]			OUT[0]	TCC3/ WO[5]			AVDD	I/O, Hi-Z
21	PC05	EXTINT[13]																		AVDD	I/O, Hi-Z
22	PC06	EXTINT[14]																		AVDD	I/O, Hi-Z
23	PC07	EXTINT[15]																		AVDD	I/O, Hi-Z
24	AVSS																				

.....continued

Pin	Pin Name	A	B (1,3)						C(4,5)		D(4,5)		E(2)	G	H	I	J(2)	K		Power Rail	Reset State
TQFP100		EIC	REF	ADC	AC	PTC	DAC	OPAMP	SERCOM	IOSET	SERCOM ALT	IOSET	TC / TCC	RTC / USB / Debug	AC / GCLK	CCL	TCC	I2S	IOSET		
25	AVDD																				
26	PA08	NMI		AIN[6]		XY[6]			SERCOM1/ PAD[0]	1	SERCOM2/ PAD[0]	2	TCC0/ WO[0]	RTC/IN[0]		IN[3]	TCC1/ WO[2]	SDI	1	VDD	I/O, Hi-Z
27	PA09	EXTINT[0]		AIN[7]		XY[7]			SERCOM1/ PAD[1]	1	SERCOM2/ PAD[1]	2	TCC0/ WO[1]	RTC/IN[1]		IN[4]	TCC1/ WO[3]	MCK[0]	1,2	VDD	I/O, Hi-Z
28	PA10	EXTINT[1]		AIN[8]		XY[8]			SERCOM1/ PAD[2]	1, 6	SERCOM2/ PAD[2]	2	TCC1/ WO[0]		GCLK/ IO[4]	IN[5]	TCC0/ WO[2]	SCK[0]	1	VDD	I/O, Hi-Z
29	PA11	EXTINT[2]		AIN[9]		XY[9]			SERCOM1/ PAD[3]	1, 6	SERCOM2/ PAD[3]	2	TCC1/ WO[1]		GCLK/ IO[3]	OUT[1]	TCC0/ WO[3]	FS[0]	2	VDD	I/O, Hi-Z
30	PC08	EXTINT[0]																		VDD	I/O, Hi-Z
31	PC09	EXTINT[1]																		VDD	I/O, Hi-Z
32	PC10	EXTINT[2]							SERCOM1/ PAD[2]	2										VDD	I/O, Hi-Z
33	PC11	EXTINT[3]							SERCOM1/ PAD[3]	2										VDD	I/O, Hi-Z
34	PC12	EXTINT[4]							SERCOM1/ PAD[0]	2										VDD	I/O, Hi-Z
35	PC13	EXTINT[5]							SERCOM1/ PAD[1]	2										VDD	I/O, Hi-Z
36	AVSSPLL																				
37	VDD																				
38	VSS																				
39	VDDPLL																				
40	PB12	EXTINT[12]				XY[30]			SERCOM3/ PAD[0]	1			TC0/ WO[0]		GCLK/ IO[6]		TCC0/ WO[6]	FS[1]	1,2	VDD	I/O, Hi-Z
41	PB13	EXTINT[13]				XY[31]			SERCOM3/ PAD[1]	1			TC0/ WO[1]		GCLK/ IO[7]		TCC0/ WO[7]	MCK[1]	1,2	VDD	I/O, Hi-Z
42	PB14	EXTINT[14]							SERCOM3/ PAD[2]	1, 4			TC1/ WO[0]	RTC/IN[4]	GCLK/ IO[0]	IN[9]				VDD	I/O, Hi-Z
43	PB15	EXTINT[15]							SERCOM3/ PAD[3]	1, 4			TC1/ WO[1]	RTC/IN[5]	GCLK/ IO[1]	IN[10]				VDD	I/O, Hi-Z
44	PC14	EXTINT[6]												RTC/IN[6]						VDD	I/O, Hi-Z
45	PC15	EXTINT[7]												RTC/IN[7]						VDD	I/O, Hi-Z
46	PA12	EXTINT[12]				XY[24]			SERCOM2/ PAD[0]	1	SERCOM4/ PAD[0]	2	TCC2/ WO[0]		AC/CMP[0]		TCC0/ WO[6]			VDD	I/O, Hi-Z
47	PA13	EXTINT[13]				XY[25]			SERCOM2/ PAD[1]	1	SERCOM4/ PAD[1]	2	TCC2/ WO[1]		AC/CMP[1]		TCC0/ WO[7]			VDD	I/O, Hi-Z

.....continued

Pin	Pin Name	A	B (1,3)						C(4,5)		D(4,5)		E(2)	G	H	I	J(2)	K		Power Rail	Reset State
TQFP100		EIC	REF	ADC	AC	PTC	DAC	OPAMP	SERCOM	IOSET	SERCOM ALT	IOSET	TC / TCC	RTC / USB / Debug	AC / GCLK	CCL	TCC	I2S	IOSET		
48	PA14 / XIN	EXTINT[3]				XY[10]			SERCOM2/ PAD[2]	1	SERCOM0/ PAD[2]	4	TC0/ WO[0]		GCLK/ IO[0]		TCC0/ WO[4]			VDD	I/O, Hi-Z
49	PA15 / XOUT	EXTINT[4]				XY[11]			SERCOM2/ PAD[3]	1	SERCOM0/ PAD[3]	4	TC0/ WO[1]		GCLK/ IO[1]		TCC0/ WO[5]			VDD	I/O, Hi-Z
50	VSS																				
51	VDD																				
52	PA16	EXTINT[5]				XY[12]			SERCOM1/ PAD[0]	3	SERCOM0/ PAD[0]	4	TCC2/ WO[0]	RTC/IN[2]	GCLK/ IO[2]	IN[0]	TCC0/ WO[6]			VDD	I/O, Hi-Z
53	PA17	EXTINT[6]				XY[13]			SERCOM1/ PAD[1]	3	SERCOM0/ PAD[1]	4	TCC2/ WO[1]	RTC/IN[3]	GCLK/ IO[3]	IN[1]	TCC0/ WO[7]			VDD	I/O, Hi-Z
54	PA18	EXTINT[7]				XY[14]			SERCOM1/ PAD[2]	3	SERCOM0/ PAD[2]	4	TC2/ WO[0]	RTC/OUT[0]	AC/CMP[0]	IN[2]	TCC0/ WO[2]	FS[0]	1	VDD	I/O, Hi-Z
55	PA19	EXTINT[0]				XY[15]			SERCOM1/ PAD[3]	3	SERCOM0/ PAD[3]	4	TC2/ WO[1]	RTC/OUT[1]	AC/CMP[1]	OUT[0]	TCC0/ WO[3]	SDO	1,2	VDD	I/O, Hi-Z
56	PC16	EXTINT[8]																		VDD	I/O, Hi-Z
57	PC17	EXTINT[9]																		VDD	I/O, Hi-Z
58	PC18	EXTINT[10]																		VDD	I/O, Hi-Z
59	PC19	EXTINT[11]																		VDD	I/O, Hi-Z
60	PC20	EXTINT[12]														IN[10]				VDD	I/O, Hi-Z
61	PC21	EXTINT[13]																		VDD	I/O, Hi-Z
62	VSS																				
63	VDD																				
64	PB16	EXTINT[0]							SERCOM5/ PAD[0]	1			TC2/ WO[0]	RTC/OUT[4]	GCLK/ IO[2]	IN[11]	TCC0/ WO[4]	SDI	2	VDD	I/O, Hi-Z
65	PB17	EXTINT[1]							SERCOM5/ PAD[1]	1			TC2/ WO[1]	RTC/OUT[5]	GCLK/ IO[3]	OUT[3]	TCC0/ WO[5]			VDD	I/O, Hi-Z
66	PB18	EXTINT[2]							SERCOM5/ PAD[2]	1	SERCOM3/ PAD[2]	2		RTC/OUT[6]			TCC0/ WO[0]			VDD	I/O, Hi-Z
67	PB19	EXTINT[3]							SERCOM5/ PAD[3]	1	SERCOM3/ PAD[3]	2		RTC/OUT[7]			TCC0/ WO[1]			VDD	I/O, Hi-Z
68	PB20	EXTINT[4]							SERCOM3/ PAD[0]	2	SERCOM5/ PAD[0]	2					TCC0/ WO[2]			VDD	I/O, Hi-Z
69	PB21	EXTINT[5]							SERCOM3/ PAD[1]	2	SERCOM5/ PAD[1]	2					TCC0/ WO[3]			VDD	I/O, Hi-Z
70	PA20	EXTINT[4]				XY[22]			SERCOM3/ PAD[2]	2,3	SERCOM2/ PAD[2]	3	TC1/ WO[0]		GCLK/ IO[4]	IN[5]	TCC0/ WO[6]	SCK[0]	2	VDD	I/O, Hi-Z

.....continued

Pin	Pin Name	A	B (1,3)						C(4,5)		D(4,5)		E(2)	G	H	I	J(2)	K		Power Rail	Reset State
TQFP100		EIC	REF	ADC	AC	PTC	DAC	OPAMP	SERCOM	IOSET	SERCOM ALT	IOSET	TC / TCC	RTC / USB / Debug	AC / GCLK	CCL	TCC	I2S	IOSET		
71	PA21	EXTINT[5]				XY[23]			SERCOM3/ PAD[3]	2,3	SERCOM2/ PAD[3]	3			GCLK/ IO[5]		TCC0/ WO[7]	SCK[1]	1,2	VDD	I/O, Hi-Z
72	PA22	EXTINT[1]				XY[16]			SERCOM0/ PAD[0]	1	SERCOM2/ PAD[0]	3	TC0/ WO[0]	RTC/OUT[2]	GCLK/ IO[2]	IN[6]	TCC0/ WO[4]			VDD	I/O, Hi-Z
73	PA23	EXTINT[2]				XY[17]			SERCOM0/ PAD[1]	1	SERCOM2/ PAD[1]	3	TC0/ WO[1]	RTC/OUT[3]	GCLK/ IO[1]	IN[7]	TCC0/ WO[5]			VDD	I/O, Hi-Z
74	PA24	EXTINT[3]							SERCOM0/ PAD[2]	1	SERCOM2/ PAD[2]	3	TC1/ WO[0]	USB/DM	AC/CMP[2]	IN[8]	TCC1/ WO[2]			VDD	I/O, Hi-Z
75	PA25	EXTINT[4]							SERCOM0/ PAD[3]	1	SERCOM2/ PAD[3]	3	TC1/ WO[1]	USB/DP	AC/CMP[3]	OUT[2]	TCC1/ WO[3]			VDD	I/O, Hi-Z
76	VSS																				
77	VDD																				
78	PB22	EXTINT[6]							SERCOM0/ PAD[2]	1, 2	SERCOM5/ PAD[2]	2,4		USB/ SOF_1KHZ	GCLK/ IO[0]	IN[0]	TCC3/ WO[0]			VDD	I/O, Hi-Z
79	PB23	EXTINT[7]							SERCOM0/ PAD[3]	1, 2	SERCOM5/ PAD[3]	2,4			GCLK/ IO[1]	OUT[0]	TCC3/ WO[1]			VDD	I/O, Hi-Z
80	PB24	EXTINT[8]							SERCOM0/ PAD[0]	2	SERCOM4/ PAD[0]	1			AC/CMP[2]					VDD	I/O, Hi-Z
81	PB25	EXTINT[9]							SERCOM0/ PAD[1]	2	SERCOM4/ PAD[1]	1			AC/CMP[3]					VDD	I/O, Hi-Z
82	PC24	EXTINT[0]							SERCOM0/ PAD[2]	2	SERCOM4/ PAD[2]	1, 2	TC2/ WO[0]							VDD	I/O, Hi-Z
83	PC25	EXTINT[1]							SERCOM0/ PAD[3]	2	SERCOM4/ PAD[3]	1,2	TC2/ WO[1]							VDD	I/O, Hi-Z
84	PC26	EXTINT[2]																		VDD	I/O, Hi-Z
85	PC27	EXTINT[3]									SERCOM1/ PAD[0]	5								VDD	I/O, Hi-Z
86	PC28	EXTINT[4]									SERCOM1/ PAD[1]	5								VDD	I/O, Hi-Z
87	VSS																				
88	RESET																			VDD	I, PU
89	VDDCORE																				
90	VSSCORE																				
91	VDDOUT																				
92	VDD																				
93	PA30 / SWCLK	EXTINT[6]				XY[18]					SERCOM1/ PAD[2]	4, 5	TCC1/ WO[0]	SWCLK	GCLK/ IO[0]	IN[3]	TCC3/ WO[4]			VDD	SWCLK, I, PU

.....continued

Pin	Pin Name	A	B (1,3)						C(4,5)		D(4,5)		E(2)	G	H	I	J(2)	K		Power Rail	Reset State
TQFP100		EIC	REF	ADC	AC	PTC	DAC	OPAMP	SERCOM	IOSET	SERCOM ALT	IOSET	TC / TCC	RTC / USB / Debug	AC / GCLK	CCL	TCC	I2S	IOSET		
94	PA31 / SWDIO	EXTINT[7]				XY[19]					SERCOM1/ PAD[3]	4, 5	TCC1/ WO[1]			OUT[1]	TCC3/ WO[5]			VDD	I/O, Hi-Z
95	PB30	EXTINT[14]							SERCOM1/ PAD[0]	4	SERCOM5/ PAD[0]	3	TCC0/ WO[0]				TCC1/ WO[2]			VDD	I/O, Hi-Z
96	PB31	EXTINT[15]							SERCOM1/ PAD[1]	4	SERCOM5/ PAD[1]	3	TCC0/ WO[1]				TCC1/ WO[3]			VDD	I/O, Hi-Z
97	PB00	EXTINT[0]		AIN[16]					SERCOM3/ PAD[2]	3	SERCOM5/ PAD[2]	3, 4				IN[1]				VDD	I/O, Hi-Z
98	PB01	EXTINT[1]		AIN[17]					SERCOM3/ PAD[3]	3	SERCOM5/ PAD[3]	3, 4				IN[2]				VDD	I/O, Hi-Z
99	PB02	EXTINT[2]		AIN[10]	AIN[4]				SERCOM3/ PAD[0]	3	SERCOM5/ PAD[0]	4	TC2/ WO[0]			OUT[0]	TCC3/ WO[2]			VDD	I/O, Hi-Z
100	PB03(6)	EXTINT[3]		AIN[11]	AIN[5]				SERCOM3/ PAD[1]	3	SERCOM5/ PAD[1]	4	TC2/ WO[1]				TCC3/ WO[3]			VDD	I/O, Hi-Z

Notes:

1. All analog pin functions are on the peripheral function B. The peripheral function B must be selected to disable the digital control of the pin.
2. Refer to TCC peripheral chapter for the list of supported features for each peripheral instance.
3. When DAC0 or DAC1 channel is enabled, respective VOUT0 (PA02) or VOUT1 (PA07) pins cannot be used for other purposes (excluding analog inputs).
4. I²C is not supported on all SERCOM pins. Refer to SERCOM I²C peripheral chapter for the list of supported features for each peripheral instance.
5. SERCOM1 signals are not available for the PIC32CM LS60 family as reserved for ATECC608B interconnection.
6. The transition time of the following pins must be greater than 50µs in order to not affect the XOSC32 cycle to cycle jitter.

5. Signal Descriptions List

The following table provides details on signal names classified by peripherals.

Table 5-1. Signal Descriptions List

Signal Name	Function	Type
Generic Clock Generator (GCLK)		
GCLK_IO[7:0]	Generators Clock Source (Input) or Generic Clock Signal (Output)	Digital I/O
Oscillators Control (OSCCTRL)		
XIN	Crystal Oscillator or External Clock Input	Analog Input (Crystal Oscillator)/Digital Input (External Clock)
XOUT	Crystal Oscillator Output	Analog Output
32 kHz Oscillators Control (OSC32KCTRL)		
XIN32	32.768 kHz Crystal Oscillator or External Clock Input	Analog Input (Crystal Oscillator)/Digital Input (External Clock)
XOUT32	32.768 kHz Crystal Oscillator Output	Analog Output
Universal Serial Bus (USB)		
DP	Differential Data Line + Port	Digital I/O
DM	Differential Data Line - Port	Digital I/O
SOF 1kHz	USB Start of Frame Output	Digital Output
Serial Communication Interface (SERCOMx)		
PAD[3:0]	General SERCOM Pins	Digital I/O
Inter-IC Sound Controller (I^2S)		
MCK[1:0]	Host Clock Pins	Digital I/O
SCK[1:0]	Serial Clock Pins	Digital I/O
FS[1:0]	I^2S Word Select / TDM Frame Sync Pins	Digital I/O
SDO	Serial Data Output for Transmit Serializer	Digital Output
SDI	Serial Data Input for Receive Serializer	Digital Input
Timer Counter (TCx)		
WO[1:0]	Capture Inputs or Waveform Outputs	Digital I/O
Timer Counter for Control (TCCx)		
WO[7:0]	Capture Inputs or Waveform Outputs	Digital I/O
Real Timer Clock (RTC)		
IN[7:0]	Tamper Detection Inputs	Digital Input
OUT[7:0]	Tamper Detection Outputs	Digital Output
Analog Comparators (AC)		
AIN[7:0]	AC Comparator Inputs	Analog Input
CMP[3:0]	AC Comparator Outputs	Digital Output
Analog Digital Converter (ADC)		
AIN[23:0]	ADC Input Channels	Analog Input
VREFA(1)	ADC External Reference Voltage A	Analog Input
VREFB	ADC External Reference Voltage B	Analog Input
Digital Analog Converter (DAC)		

.....continued

Signal Name	Function	Type
VOUT0	DAC Voltage Output 0	Analog Output
VOUT1	DAC Voltage Output 1	Analog Output
VREFA(1)	DAC External Reference Voltage A	Analog Input
Operational Amplifier (OPAMP)		
OA[2:0]NEG	OPAMP Negative Inputs	Analog Input
OA[2:0]POS	OPAMP Positive Inputs	Analog Input
OA[2:0]OUT	OPAMP Outputs	Analog Output
Peripheral Touch Controller (PTC)		
XY[31:0]	X-lines and Y-lines	Digital Output (X-line) /Analog I/O (Y-line)
Custom Control Logic (CCL)		
IN[11:0]	Inputs to lookup table	Digital Input
OUT[3:0]	Outputs from lookup table	Digital Output
External Interrupt Controller (EIC)		
EXTINT[15:0]	External Interrupts Pins	Digital Input
NMI	Non-Maskable Interrupt Pin	Digital Input
General Purpose I/O (PORT)		
PA25-PA00 / PA31-PA30	General Purpose I/O Pin in Port A	Digital I/O
PB09-PB00 / PB25-PB12 / PB31-PB30	General Purpose I/O Pin in Port B	Digital I/O
PC03-PC00 / PC21-PC05 / PC28-PC24	General Purpose I/O Pin in Port C	Digital I/O
Reset Controller (RSTC)		
$\overline{\text{RESET}}$	External Reset Pin (Active Level: LOW)	Digital Input
Debug Service Unit (DSU)		
SWCLK	Serial Wire Clock	Digital Input
SWDIO	Serial Wire Bidirectional Data Pin	Digital I/O

Note:

1. VREFA is shared between the ADC and DAC peripherals.

6. Power Supplies

The PIC32CM LE00/LS00/LS60 have the following power supply pins:

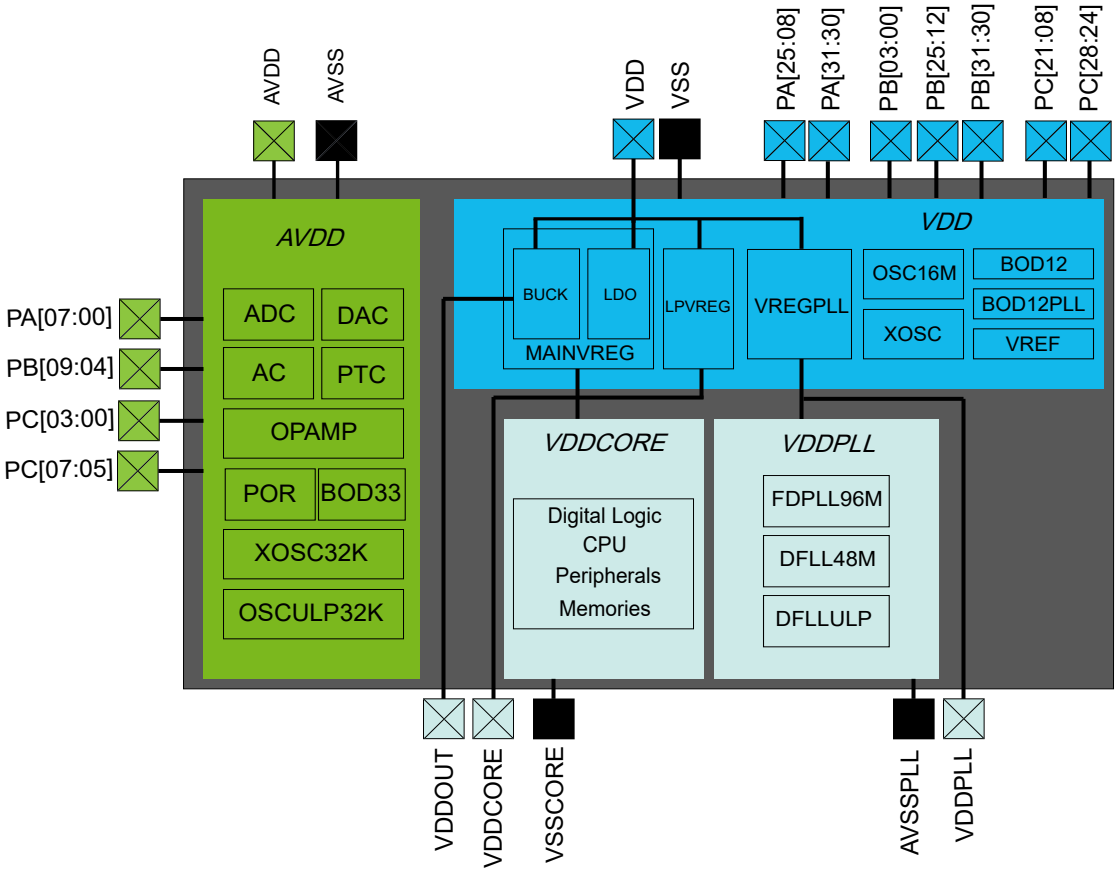
Table 6-1. PIC32CM LE00/LS00/LS60 Power Supplies

Name	Associated Ground	Description	Voltage Range (Electrical Characteristics Parameters Number)
VDD	VSS	Digital Supply Voltage	REG_37
AVDD	AVSS	Analog Supply Voltage	REG_39
VDDCORE	VSSCORE	Core Supply Voltage Output (LDO mode) Core Supply Voltage Input (BUCK mode) **Note:** VDDCORE is not an input for an external power supply.	REG_36
VDDPLL	AVSSPLL	FDPLL96M, DFLL48M and DFLLULP Supply Voltage Output	REG_38
VDDOUT	-	Switching Regulator Mode (BUCK) Output	—

Note: REG_X values: refer to Power Supply Electrical Specifications from the Electrical Characteristics chapter.

Important: The same voltage must be applied to VDD and AVDD. Refer to Power-up Considerations in the Device Start-up chapter for more details.

Figure 6-1. Power Supplies Block Diagram



7. Device Start-up

7.1 Power-up Considerations

After power-up, the device is kept in reset until the power has stabilized throughout the device.

Maximum VDD/AVDD rise and falling rates can be found on the Power Supply Electrical Specifications from Electrical Characteristics chapter.

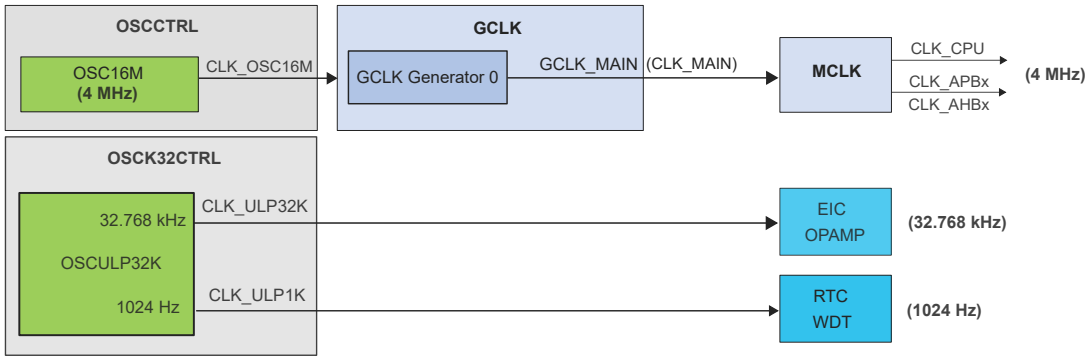
7.2 Clocks after Reset

The device selects the OSC16M oscillator which is enabled by default after reset and configured at 4 MHz.

This 4 MHz clock is also the default time base for the Generic Clock Generator 0 which provides the main clock (CLK_MAIN) to the system through the GCLK_MAIN clock.
Note: Other generic clocks are disabled to optimize the power consumption.

The OSCULP32K is also enabled by default after reset and is used as source for different peripherals requiring a default slow clock source.

Figure 7-1. Clocks Distribution after Reset (Simplified)



7.3 Initial Instructions Fetching

After reset is released, the CPU starts fetching from the Boot ROM.

Unless a debugger is connected and places the Boot ROM in a specific mode called Boot Interactive mode, the CPU will jump to the Flash memory loading the Program Counter (PC) and Stack Pointer (SP) values and start fetching flash user code.

Important: Before jumping to the Flash, the Boot ROM resets the first 4 KB of SRAM and the first 60 bytes of the TrustRAM. The clocks remain unchanged.

In addition, the PIC32CM LS00/LS60 Boot ROM has extra security features, such as device integrity checks, memories and peripherals security attributions, DICE security standard support and Secure Boot that can be executed before jumping to the Flash in Secure state.

Note: PIC32CM LE00/LS00/LS60 Boot Interactive mode allows a debugger to perform several actions on the device, such as NVM areas integrity check, chip erase.

Refer to 13. Boot ROM chapter for more information.

7.4 I/O Pins

After reset, the I/O pins are tri-stated except:

1. PA30 pin: configured in peripheral mode with pull-up enabled (SWCLK peripheral function selected for debugger probe detection support).

7.5 Performance Level Overview

The PIC32CM LE00/LS00/LS60 support two different performance levels: PL0 and PL2.

The default performance level after reset is PL0. This performance level is aiming for the lowest power consumption by limiting logic speeds and CPU frequency. As a consequence, some peripherals and clock sources will work with limited capabilities.

Full device functionality and performance will be ensured with PL2 mode.

Maximum CPU frequency regarding PL0 and PL2 performance level modes can be found in the General Operating Ratings and Thermal Conditions section from the Electrical Characteristics chapter.

8. Product Mapping

Figure 8-1. PIC32CM LE00 Product Mapping

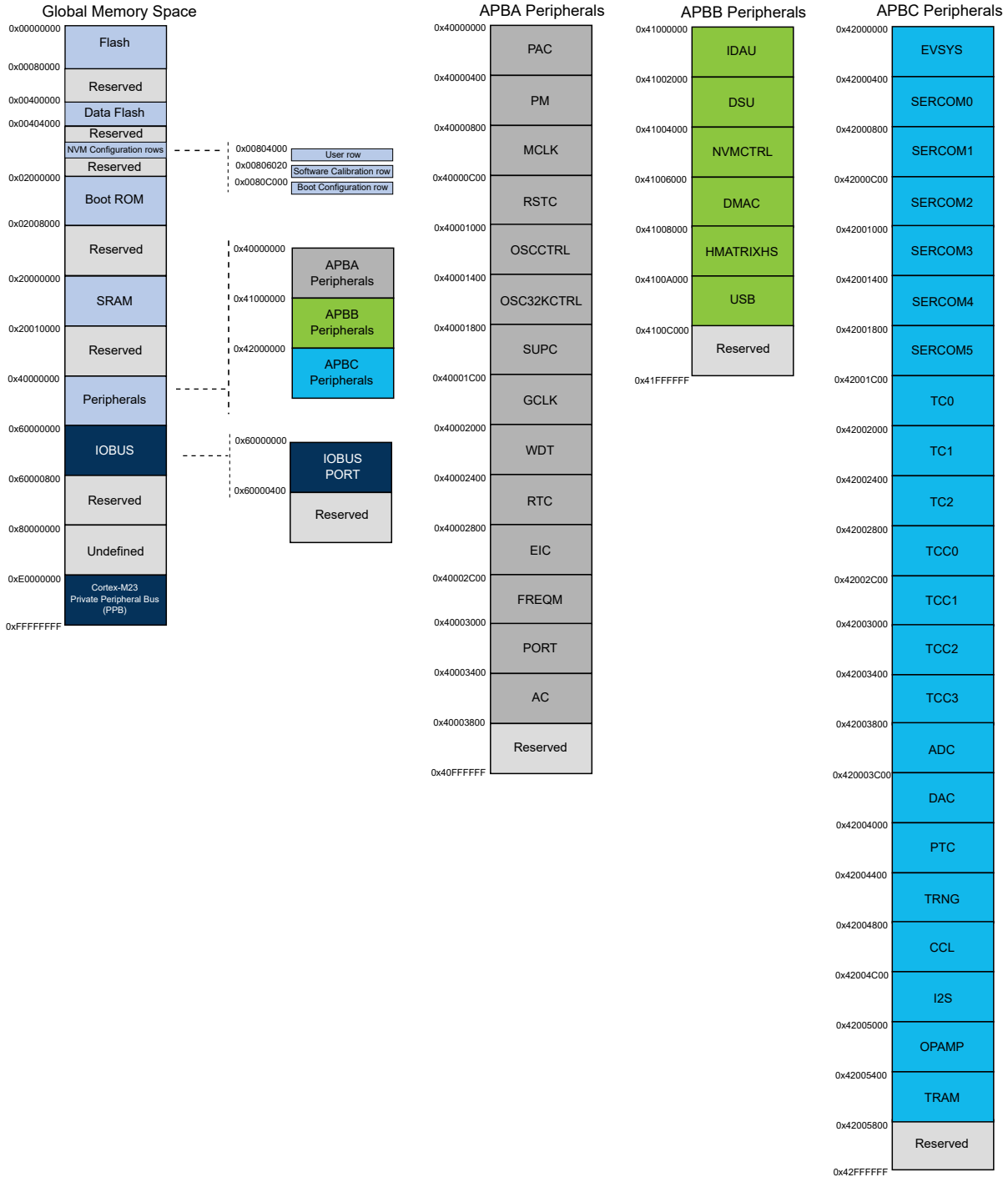
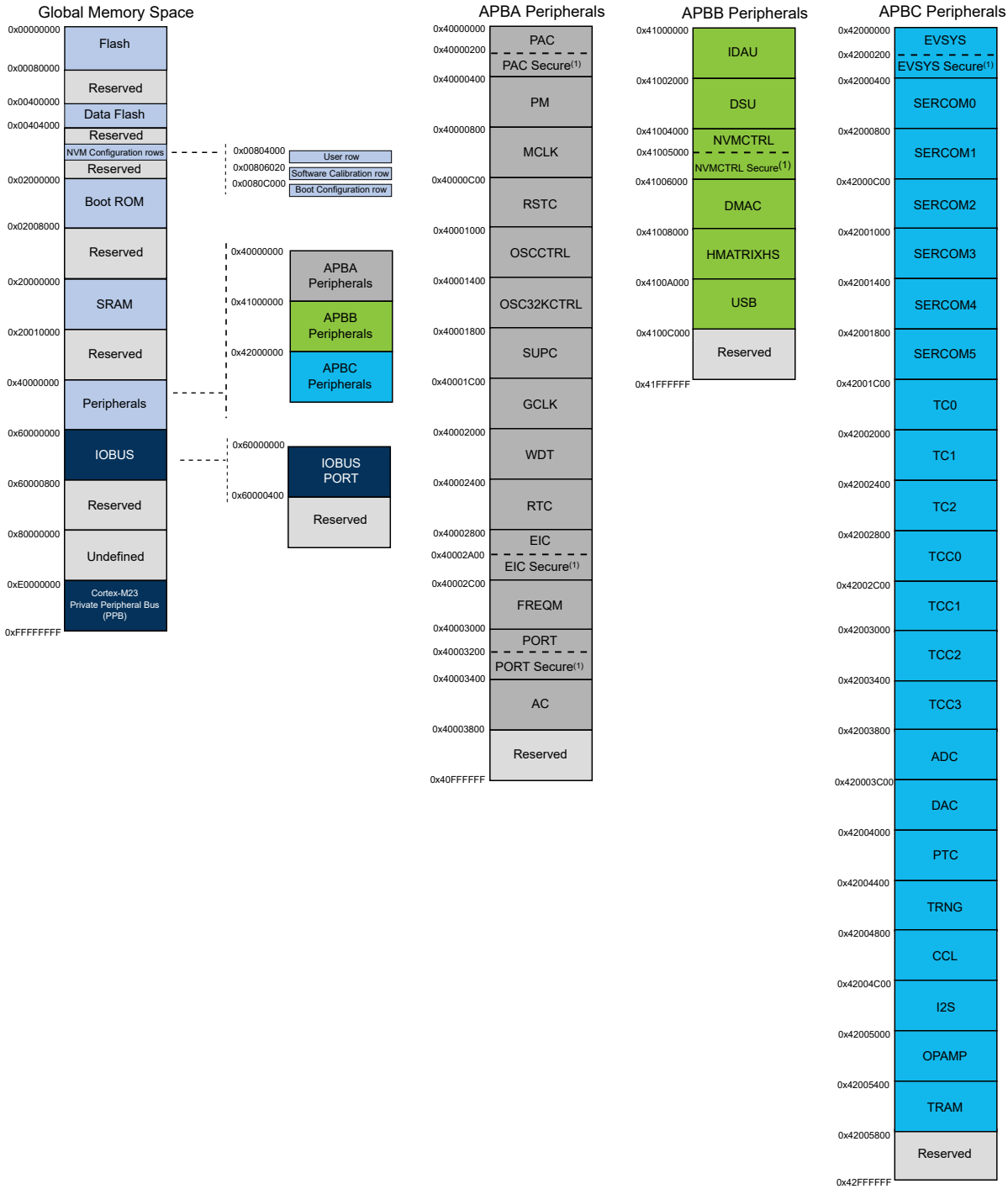


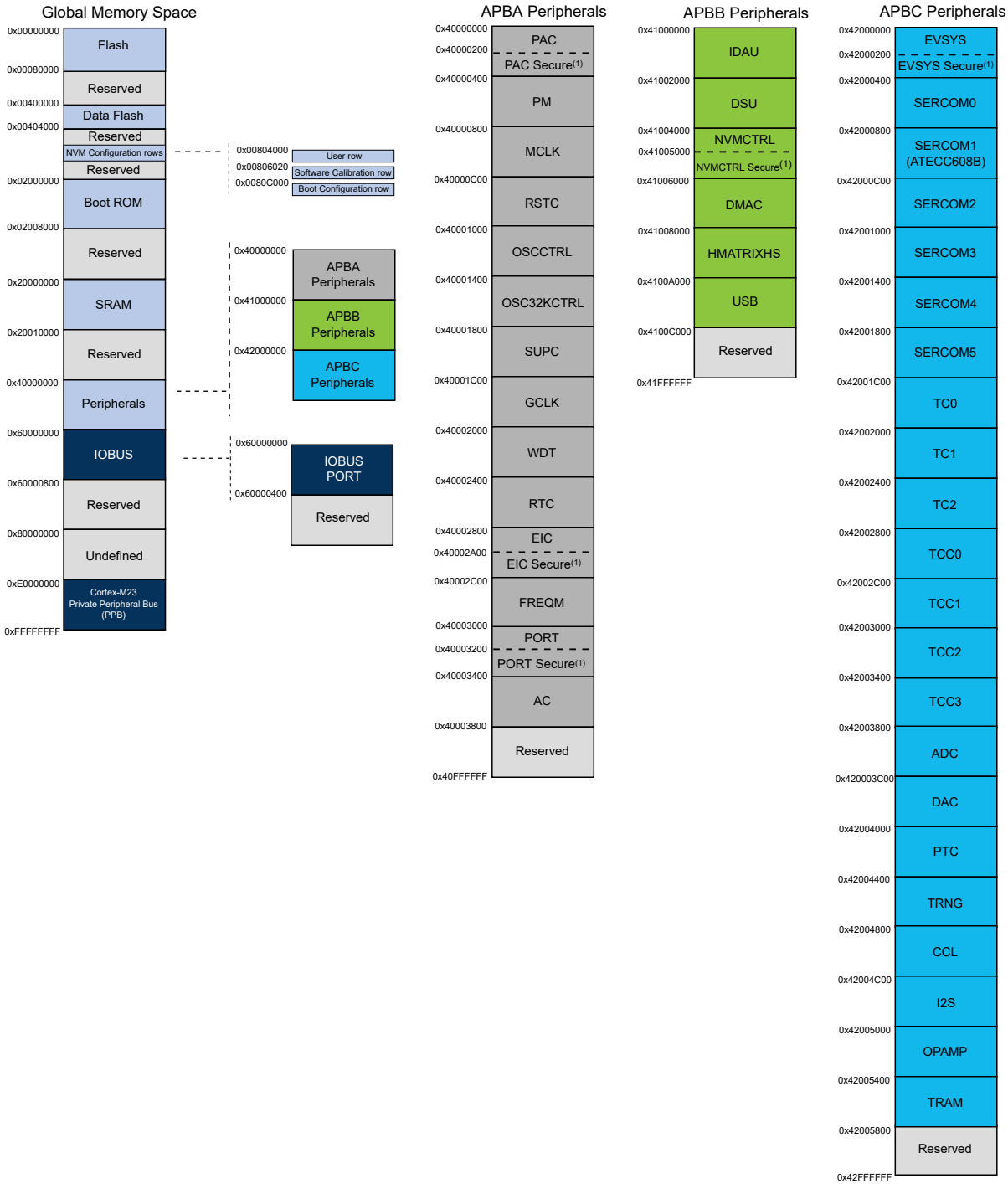
Figure 8-2. PIC32CM LS00 Product Mapping



Note:

1. This peripheral secure memory region will only appear if the peripheral is secured using PAC. Refer to Mix-Secure Peripherals for details.

Figure 8-3. PIC32CM LS60 Product Mapping



Note:

1. This peripheral secure memory region will only appear if the peripheral is secured using PAC. Refer to Mix-Secure Peripherals for details.

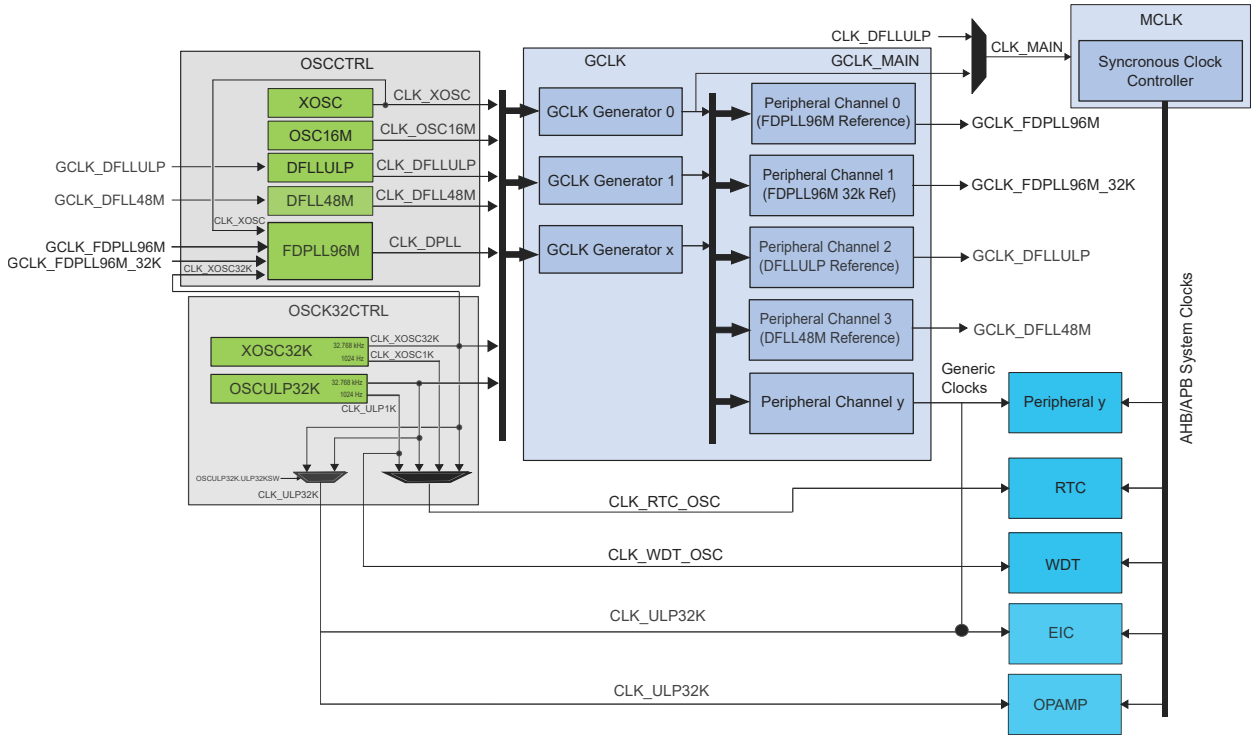
9. Peripherals

9.1 Clocks Distribution

The PIC32CM LE00/LS00/LS60 clock system is composed of:

- *Clock sources (CLK_XXX)*, which are controlled by OSCCTRL and OSC32KCTRL peripherals
- The *Generic Clock Controller (GCLK),* which generates and controls the asynchronous clocks of the device (GCLK_XXX):
  - The Generic Clock Generators are programmable prescalers that can use any of the clock sources as a time base
  - The Peripheral Channels select their generic clock from the different GCLK generators to clock their associated peripherals.
- The *Main Clock Controller (MCLK)*, which generates and controls the synchronous clocks of the device:
  - This includes the CPU, bus clocks (APB, AHB) as well as the user interfaces of the peripherals.

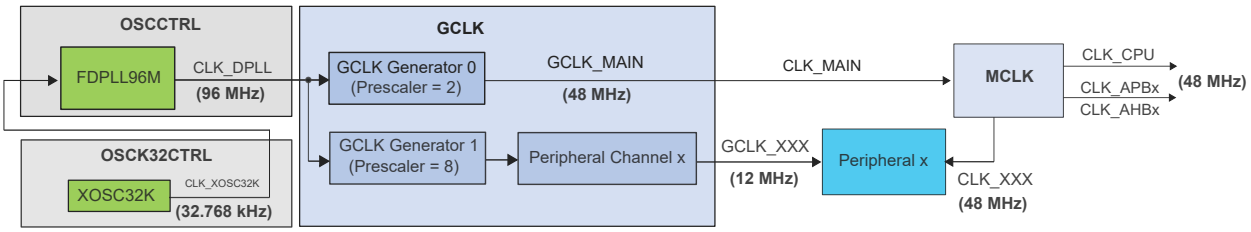
Figure 9-1. Clocks Distribution



Here is an example using the FDPPL96M as clock source where:

- The Generic Clock Generator 0 provides the Main Clock source (48MHz)
- The Generic Clock Generator 1 provides a lower clock to the peripheral (12MHz)

Figure 9-2. 48MHz Clocks System Example (Simplified)



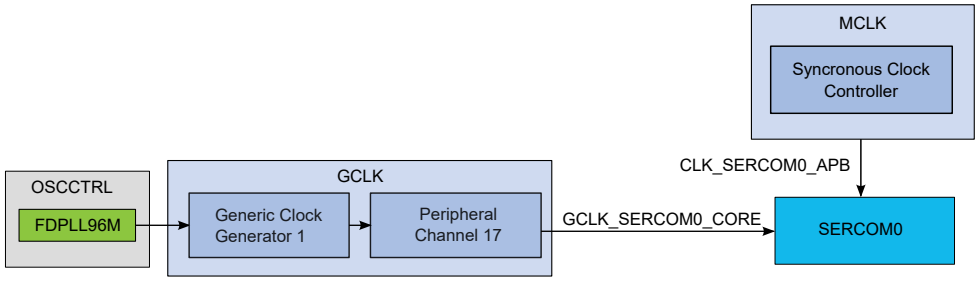
Note: As the CPU and the peripherals can be in different clock domains, that is, they are clocked from different clock sources and with different clock speeds, some peripheral accesses by the CPU need to be synchronized. In this case the peripheral includes a Synchronization Busy (SYNCBUSY) register that can be used to check if a sync operation is in progress.
For a general description, see 9.5.3. Registers Synchronization.

9.2 Enabling a Peripheral

Configuring a peripheral that relies on an asynchronous clock requires the following Generic Clock Controller (GCLK) configuration:

- A clock from the Generic Clock Generator must be configured to use one of the running Clock Sources, and the Generator must be enabled
- The Peripheral Channel that provides the Generic Clock signal to the peripheral must be configured to use a running Generic Clock Generator, and the Generic Clock must be enabled
- A synchronous clock provided by the Main Clock Controller (MCLK) is also required to configure the peripheral user interface

Figure 9-3. SERCOM0 Clock Distribution Example



This results in the following registers configuration:

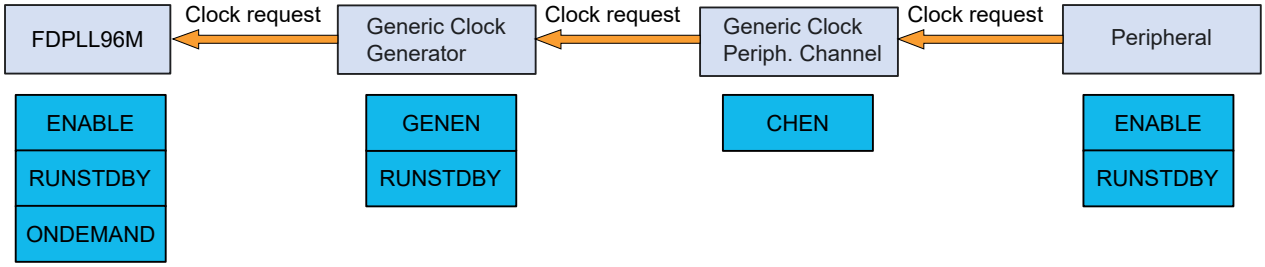
- The source oscillator for a generic clock generator 'n' is selected by writing to the Source bit field in the Generator Control n register (GCLK.GENCTRLn.SRC).
- A Peripheral Channel ‘m’ can be configured to use a specific Generic Clock Generator by writing to the Generic Clock Generator bit field in the respective Peripheral Channel m register (GCLK.PCHCTRLm.GEN)

Note: The Peripheral Channel number, *m*, is fixed for a given peripheral. See the Mapping table in the description of GCLK.PCHCTRLm.

All synchronous peripheral clocks are by default enabled after reset (MCLK.AHBMASK, MCLK.APBxMASK).

9.3 On Demand Clock Requests

Figure 9-4. Clock Request Routing



All clock sources in the system can be run in an on-demand mode: the clock source is in a stopped state unless a peripheral is requesting the clock source. Clock requests propagate from the peripheral through the GCLK, to the clock source. If one or more peripheral is using a clock source, the clock source will be started/kept running. As soon as the clock source is no longer needed and no peripheral has an active request, the clock source will be stopped until requested again.

The clock request can reach the clock source only if the peripheral, the generic clock and the clock from the Generic Clock Generator in-between are enabled. The time taken from a clock request being asserted to the clock source being ready is dependent on the clock source startup time, clock source frequency as well as the divider used in the Generic Clock Generator. The total startup time T_{start} from a clock request until the clock is available for the peripheral is between:

T_{start_max} = Clock source startup time + 2 × clock source periods + 2 × divided clock source periods

T_{start_min} = Clock source startup time + 1 × clock source period + 1 × divided clock source period

The time between the last active clock request stopped and the clock is shut down, T_{stop}, is between:

T_{stop_min} = 1 × divided clock source period + 1 × clock source period

T_{stop_max} = 2 × divided clock source periods + 2 × clock source periods

The On-Demand function can be disabled individually for each clock source by clearing the ONDEMAND bit located in each clock source controller. Consequently, the clock will always run whatever the clock request status is. This has the effect of removing the clock source startup time at the cost of power consumption.

The clock request mechanism can be configured to work in standby mode by setting the RUNSDTBY bits of the modules.

9.4 Peripherals Dependencies

Table 9-1. Peripherals Dependencies

Peripheral name	Base address	NVIC IRQ Index	MCLK AHB/APB Clocks	GCLK Peripheral Channel Index (GCLK.PCHCTRL)	PAC Peripheral Identifier Index (PAC.WRCTRL)	DMA Trigger Source Index (DMAC.CHCTRLB)	Events (EVSYS)		Power Domain (PM.STDBYCFG)
							Users (EVSYS.USER)	Generators (EVSYS.CHANNEL)	
AHB-APB Bridge A (APBA) Peripherals									
PAC	0x40000000	30: ERR	CLK_PAC_AHB CLK_PAC_APB	—	0	—	—	91: ERR	PDSW
PM	0x40000400	0: PLRDY	CLK_PM_APB	—	1	—	—	—	PDAO
MCLK	0x40000800	0: CKRDY	CLK_MCLK_APB	—	2	—	—	—	PDSW
RSTC	0x40000C00	—	CLK_RSTC_APB	—	3	—	—	—	PDAO
OSCCTRL	0x40001000	0: XOSCRDY, XOSCFAIL, OSC16MRDY, DFLLULPRDY, DFLLULPLOCK, DFLLULPNOLOCK, DPLLLCKR, DPLLLCKF, DPLLLTO, DPLLLDRTO, DFLLRDY, DFLLOOB, DFLLLCKF, DFLLLCKC, DFLLRCS	CLK_OSCCTRL_APB	0: GCLK_FDPLL96M 1: GCLK_FDPLL96M_32K 2: GCLK_DFLLULP 3: GCLK_DFLL48M	4	—	0: TUNE	1: CFD	PDSW

.....continued

Peripheral name	Base address	NVIC IRQ Index	MCLK AHB/APB Clocks	GCLK Peripheral Channel Index (GCLK.PCHCTRL)	PAC Peripheral Identifier Index (PAC.WRCTRL)	DMA Trigger Source Index (DMAC.CHCTRLB)	Events (EVSYS)		Power Domain (PM.STDBYCFG)
							Users (EVSYS.USER)	Generators (EVSYS.CHANNEL)	
OSC32KCTRL	0x40001400	0: XOSC32KRDY, CLKFAIL	CLK_OSC32KCTRL_APB	—	5	—	—	2: CFD	PDAO
SUPC	0x40001800	0: BOD33RDY,BOD33DET, B33SRDY, VREGRDY, VCORERDY, ULPVREFRDY,VCORE PLLRDY	CLK_SUPC_APB	—	6	—	—	3: BOD33DET	PDAO
GCLK	0x40001C00	—	CLK_GCLK_APB	—	7	—	—	—	PDSW
WDT	0x40002000	1: EW	CLK_WDT_APB	—	8	—	—	—	PDSW
RTC	0x40002400	2: CMP0-1, TAMPER, OVF, PER0-7, ALARM0	CLK_RTC_APB	—	9	1: TIMESTAMP	1: TAMPEVT	4-11 : PER0-7 12 : ALARM0 12-13 : CMP0-1 14 : TAMPER 15 : OVF 16 : PERD	PDAO
EIC	0x40002800	3: EXTINT0 4: EXTINT1 5: EXTINT2 6: EXTINT3 7: EXTINT4 8: EXTINT5 9: EXTINT6 10: EXTINT7 11: EXTINT8-15, NSCHK NMI	CLK_EIC_APB	4: GCLK_EIC	10	—	—	17-32: EXTINT0-15	PDAO
FREQM	0x40002C00	12: DONE	CLK_FREQM_APB	5: GCLK_FREQM_MSR 6: GCLK_FREQM_REF	11	—	—	—	PDSW
PORT	0x40003000	14: NSCHK	CLK_PORT_APB	—	12	—	3-6 : EVU0-3	—	PDAO
AC	0x40003400	64: COMP0-3, WIN0-1	CLK_AC_APB	29: GCLK_AC	13	—	40-43: COMP0-3	76-79: COMP0-3 80-81: WIN0-1	PDAO
AHB-APB Bridge B (APBB) Peripherals									
IDAU	0x41000000	—	CLK_IDAU_APB	—	32	—	—	—	PDSW
DSU	0x41002000	—	CLK_DSU_AHB CLK_DSU_APB	—	33	2-3: DCC0-1	—	—	PDSW
NVMCTRL	0x41004000	13: DONE, PROGE, LOCKE, NVME, KEYE, NSCHK	CLK_NVMCTRL_AHB CLK_NVMCTRL_APB	—	34	—	2: AUTOW	—	PDSW
DMAC	0x41006000	15: SUSP0, TERR0, TCMPL0 16: SUSP1, TERR1, TCMPL1 17: SUSP2, TERR2, TCMPL2 18: SUSP3, TERR3, TCMPL3 19: SUSP4-15, TERR4-15, TCMPL4-15	CLK_DMAC_AHB	—	35	—	7-14: CH0-7	33-40: CH0-7	PDSW
HMATRIXHS	0x41008000	—	CLK_HMATRIXHS_AHB CLK_HMATRIXHS_APB	—	36	—	—	—	PDSW
USB	0x4100A000	20: EORSM/DNRSM, EORST/RST, LPMNYET/DCONN, LPMSUSP/DDISC, RAMACER, RXSTP/TXSTP, SOF/HSOF, STALL0/STALL, STALL1, SUSPEND, TRCPT0, TRCPT1, TRFAIL0/TRFAIL, TRFAIL1/PERR, UPRSM, WAKEUP	CLK_USB_AHB CLK_USB_APB	7: GCLK_USB	37	—	—	—	PDSW
AHB-APB Bridge C (APBC) Peripherals									

.....continued

Peripheral name	Base address	NVIC IRQ Index	MCLK AHB/APB Clocks	GCLK Peripheral Channel Index (GCLK.PCHCTRL)	PAC Peripheral Identifier Index (PAC.WRCTRL)	DMA Trigger Source Index (DMAC.CHCTRLB)	Events (EVSYS)		Power Domain (PM.STDBYCFG)
							Users (EVSYS.USER)	Generators (EVSYS.CHANNEL)	
EVSYS	0x42000000	21: EVD0, OVR0 22: EVD1, OVR1 23: EVD2, OVR2 24: EVD3, OVR3 25: EVD4, OVR4 26: EVD5, OVR5 27: EVD6, OVR6 28: EVD7, OVR7 29: NSCHK	CLK_EVSYS_APB	8: GCLK_EVSYS_CHANNEL_0 9: GCLK_EVSYS_CHANNEL_1 10: GCLK_EVSYS_CHANNEL_2 11: GCLK_EVSYS_CHANNEL_3 12: GCLK_EVSYS_CHANNEL_4 13: GCLK_EVSYS_CHANNEL_5 14: GCLK_EVSYS_CHANNEL_6 15: GCLK_EVSYS_CHANNEL_7	64	—	—	—	PDSW
SERCOM0(1)	0x42000400	31: bit 0 (DRE, MB, PREC) 32: bit 1 (TXC, SB, AMATCH) 33: bit 2 (RXC, DRDY) 34: bit 3-7 (RXS, TXFE, SSL, CTSIC, RXFF, RXBRK, ERROR)	CLK_SERCOM0_APB	17: GCLK_SERCOM0_CORE 16: GCLK_SERCOM0_SLOW	65	4: RX 5: TX	—	—	PDSW
SERCOM1(1)	0x42000800	35: bit 0 (DRE, MB, PREC) 36: bit 1 (TXC, SB, AMATCH) 37: bit 2 (RXC, DRDY) 38: bit 3-7 (RXS, TXFE, SSL, CTSIC, RXFF, RXBRK, ERROR)	CLK_SERCOM1_APB	18: GCLK_SERCOM1_CORE 16: GCLK_SERCOM1_SLOW	66	6: RX 7: TX	—	—	PDSW
SERCOM2(1)	0x42000C00	39: bit 0 (DRE, MB, PREC) 40: bit 1 (TXC, SB, AMATCH) 41: bit 2 (RXC, DRDY) 42: bit 3-7 (RXS, TXFE, SSL, CTSIC, RXFF, RXBRK, ERROR)	CLK_SERCOM2_APB	19: GCLK_SERCOM2_CORE 16: GCLK_SERCOM2_SLOW	67	8: RX 9: TX	—	—	PDSW
SERCOM3(1)	0x42001000	43: bit 0 (DRE, MB, PREC) 44: bit 1 (TXC, SB, AMATCH) 45: bit 2 (RXC, DRDY) 46: bit 3-7 (RXS, TXFE, SSL, CTSIC, RXFF, RXBRK, ERROR)	CLK_SERCOM3_APB	20: GCLK_SERCOM3_CORE 16: GCLK_SERCOM3_SLOW	68	10: RX 11: TX	—	—	PDSW
SERCOM4(1)	0x42001400	47: bit 0 (DRE, MB, PREC) 48: bit 1 (TXC, SB, AMATCH) 49: bit 2 (RXC, DRDY) 50: bit 3-7 (RXS, TXFE, SSL, CTSIC, RXFF, RXBRK, ERROR)	CLK_SERCOM4_APB	21: GCLK_SERCOM4_CORE 16: GCLK_SERCOM4_SLOW	69	12: RX 13: TX	—	—	PDSW
SERCOM5(1)	0x42001800	51: bit 0 (DRE, MB, PREC) 52: bit 1 (TXC, SB, AMATCH) 53: bit 2 (RXC, DRDY) 54: bit 3-7 (RXS, TXFE, SSL, CTSIC, RXFF, RXBRK, ERROR)	CLK_SERCOM5_APB	22: GCLK_SERCOM5_CORE 16: GCLK_SERCOM5_SLOW	70	14: RX 15: TX	—	—	PDSW
TC0	0x42001C00	55: ERR, MC0, MC1, OVF	CLK_TC0_APB	23: GCLK_TC0_TC1	71	16: OVF 17-18: MC0-1	15: EVU	41: OVF 42-43: MC0-1	PDSW
TC1	0x42002000	56: ERR, MC0, MC1, OVF	CLK_TC1_APB	23: GCLK_TC0_TC1	72	19: OVF 20-21: MC0-1	16: EVU	44: OVF 45-46: MC0-1	PDSW
TC2	0x42002400	57: ERR, MC0, MC1, OVF	CLK_TC2_APB	24: GCLK_TC2	73	22: OVF 23-24: MC0-1	17: EVU	47: OVF 48-49: MC0-1	PDSW
TCC0	0x42002800	58: ERR, MC0-3, OVF, TRG, CNT, DFS, UFS, FAULTA, FAULTB, FAULT0, FAULT1	CLK_TCC0_APB	25: GCLK_TCC0_TCC1	74	25: OVF 26-29: MC0-3	18-19: EVU0-1 20-23: MC0-3	50: TRG 51: CNT 52-55: MC0-3 56: OVF	PDSW
TCC1	0x42002C00	59: ERR, MC0-1, OVF, TRG, CNT, DFS, UFS, FAULTA, FAULTB, FAULT0, FAULT1	CLK_TCC1_APB	25: GCLK_TCC0_TCC1	75	30: OVF 31-32: MC0-1	24-25: EVU0-1 26-27: MC0-1	57: TRG 58: CNT 59-60: MC0-1 61: OVF	PDSW

.....continued

Peripheral name	Base address	NVIC IRQ Index	MCLK AHB/APB Clocks	GCLK Peripheral Channel Index (GCLK.PCHCTRL)	PAC Peripheral Identifier Index (PAC.WRCTRL)	DMA Trigger Source Index (DMAC.CHCTRLB)	Events (EVSYS)		Power Domain (PM.STDBYCFG)
							Users (EVSYS.USER)	Generators (EVSYS.CHANNEL)	
TCC2	0x42003000	60: ERR, MC0-1, OVF, TRG, CNT, DFS, UFS, FAULTA, FAULTB, FAULT0, FAULT1	CLK_TCC2_APB	26: GCLK_TCC2	76	33: OVF 34-35: MC0-1	28-29: EVU0-1 30-31: MC0-1	62: TRG 63: CNT 64-65: MC0-1 66: OVF	PDSW
TCC3	0x42003400	61: ERR, MC0-3, OVF, TRG, CNT, DFS, UFS, FAULTA, FAULTB, FAULT0, FAULT1	CLK_TCC3_APB	27: GCLK_TCC3	77	36: OVF 37-40: MC0-3	32-33: EVU0-1 34-37: MC0-3	67: TRG 68: CNT 69-72: MC0-3 73: OVF	PDSW
ADC	0x42003800	62: OVERRUN, WINMON 63: RESRDY	CLK_ADC_APB	28: GCLK_ADC	78	41: RESRDY	38: START 39 : FLUSH	74: RESRDY 75: WINMON	PDSW
DAC	0x42003C00	65: UNDERRUN0-1 66: EMPTY0-1	CLK_DAC_APB	30: GCLK_DAC	79	42-43: EMPTY0-1	44-45: START0-1	82-83: EMPTY0-1	PDSW
PTC	0x42004000	67: EOC, WCOMP	CLK_PTC_APB	31: GCLK_PTC	80	44 : EOC 45 : SEQ 46 : WCOMP	46 : STCONV 47 : DSEQR	84: EOC 85: WCOMP	PDSW
TRNG	0x42004400	68: DATARDY	CLK_TRNG_APB	—	81	—	—	86 : DATARDY	PDSW
CCL	0x42004800	—	CLK_CCL_APB	32: GCLK_CCL	82	—	48-51 : LUT0-3	87-90 : LUT0-3	PDSW
I2S	0x42004C00	69: RXRDY0-1, TXRDY0-1, RXOR0-1, TXUR0-1	CLK_I2S_APB	33: GCLK_I2S_0 34: GCLK_I2S_1	83	47-48: RX0-1 49-50: TX0-1	—	—	PDSW
OPAMP	0x42005000	—	CLK_OPAMP_APB	—	84	—	—	—	PDSW
TRAM	0x42005400	70: DRP, ERR	CLK_TRAM_AHB	—	85	—	—	—	PDSW

Note:

1. GCLK_SERCOMx_SLOW is only used by SERCOM I²C.

9.5 Registers Description

9.5.1 Registers Properties

Registers can be 8, 16, or 32 bits wide. Atomic 8-bit, 16-bit 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.

PAC Write-Protection Register Property:

Some registers are optionally write-protected by the Peripheral Access Controller (PAC).

PAC write protection is denoted by the "PAC Write-Protection" property in each individual register description.

For more details, refer to the PAC - Peripheral Access Controller.

Read-Synchronized, Write-Synchronized Register Property:

Some registers (or bit fields within a register) require synchronization when read and/or written.

Synchronization is denoted by the "Read-Synchronized" (bits) and "Write-Synchronized" (bits) property in each individual register description.

For more details, refer to Register Synchronization.

Enable-Protected Register Property:

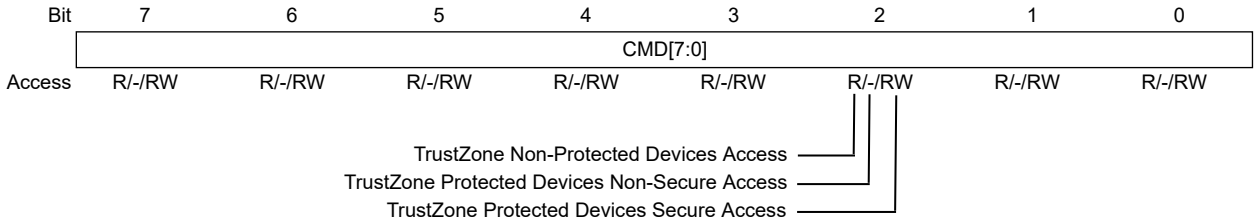
Some registers (or bit fields within a register) can only be written when the peripheral is disabled.

Such protection is denoted by the "Enable-Protected" (bits) property in each individual register description.

Mix-Secure Peripherals Register Property (PIC32CM LS00/LS60 only):

A Mix-Secure Peripheral has different types of registers (Non-Secure, Secure, Write-Secure, Mix-Secure, and Write-Mix-Secure) with different access permissions for each bit field.

The access permissions are specified as shown in the following figure.



For more details, refer to Peripherals Security Attribution.

9.5.2 Registers Access Permissions (PIC32CM LS00/LS60 only)

Each Peripheral has different register access permissions depending on the following:

- Its PAC Security Attribution (Secure or Non-Secure)
- If it is a Mix-Secure Peripheral: PAC, NVMCTRL, PORT, EIC, EVSYS
- If it is an Always Secure Peripheral: IDAU
- If it is an Always Non-Secure Peripheral: DSU

Peripherals excluding Mix-Secure Peripherals case:

- If the peripheral is configured as Non-Secure in the PAC:
  - Secure access and Non-Secure access are granted
- If the peripheral is configured as Secure in the PAC:
  - Secure access is granted
  - Non-Secure access is discarded (Write is ignored, read 0x0) and a PAC error is triggered

Always Secure Peripheral case (IDAU) :

- Secure access is granted
- Non-Secure access is discarded (Write is ignored, read 0x0) and a PAC error is triggered

Always Non-Secure Peripheral case (DSU):

- Secure access and Non-Secure access are granted

Mix-Secure Peripherals case (PAC, NVMCTRL, PORT, EIC, EVSYS):

- If the peripheral is configured as Non-Secure in the PAC:
  - Secure access and Non-Secure access are granted
- If the peripheral is configured as Secure in the PAC:
  - The peripheral register map is duplicated in a Secure and Non-Secure alias which both have different access permissions. Refer to Mix-Secure Peripherals.

For more details, refer to Peripherals Security Attribution.

9.5.3 Registers Synchronization

9.5.3.1 Overview

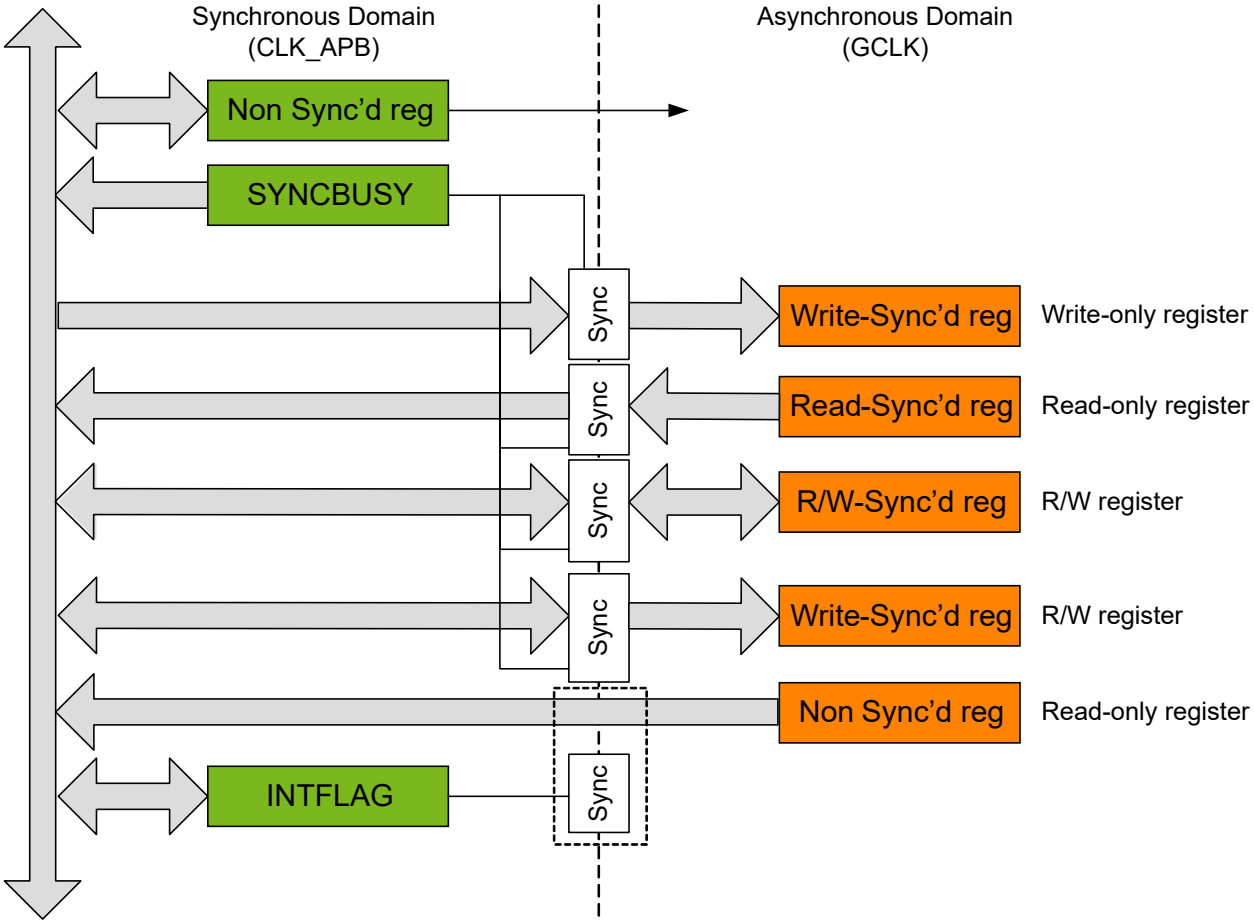
Most of the peripherals are composed of one digital bus interface connected to the APB or AHB bus and running from a corresponding clock in the Main Clock domain, and one peripheral core running from the peripheral Generic Clock (GCLK).

Communication between these clock domains must be synchronized. This mechanism is implemented in hardware, so the synchronization process takes place even if the peripheral generic clock is running from the same clock source and on the same frequency as the bus interface.

As shown in the following figure, each register that requires synchronization has its individual synchronizer and its individual synchronization status bit in the Synchronization Busy register (SYNCBUSY).
Note: For registers requiring both read- and write-synchronization, the corresponding bit in SYNCBUSY is shared.

Synchronization is denoted by the "Read-Synchronized" (bits) and "Write-Synchronized" (bits) property in each individual register description.

Figure 9-5. Register Synchronization Overview



9.5.3.2 General Write Synchronization

Write-Synchronization is triggered by writing to a register in the peripheral clock domain (GCLK). The respective bit in the Synchronization Busy register (SYNCBUSY) will be set when the write-synchronization starts and cleared when the write-synchronization is complete. Refer also to 9.5.3.5. Synchronization Delay.

When write-synchronization is ongoing for a register, any subsequent write attempts to this register will be discarded, and an error will be reported though the Peripheral Access Controller (PAC).

Example:

REGA, REGB are 8-bit core registers. REGC is a 16-bit core register.

Offset	Register
0x00	REGA
0x01	REGB
0x02	REGC
0x03	

Synchronization is per register, so multiple registers can be synchronized in parallel. Consequently, after REGA (8-bit access) was written, REGB (8-bit access) can be written immediately without error.

REGC (16-bit access) can be written without affecting REGA or REGB. If REGC is written to in two consecutive 8-bit accesses without waiting for synchronization, the second write attempt will be discarded and an error is generated through the PAC.

A 32-bit access to offset 0x00 will write all three registers. Note that REGA, REGB and REGC can be updated at different times because of independent write synchronization.

9.5.3.3 General Read Synchronization

Unless otherwise stated, read-synchronized registers are synchronized each time the register value is updated. The respective bit in the Synchronization Busy register (SYNCBUSY) will be set when the read-synchronization starts and cleared when the read-synchronization is complete. Therefore reading a read-synchronized register before its corresponding SYNCBUSY bit is cleared will return the last synchronized value.

9.5.3.4 Completion of Synchronization

In order to check if synchronization is complete, the user can either poll the relevant bits in SYNCBUSY or use the Synchronization Ready interrupt (if available). The Synchronization Ready interrupt flag will be set when all ongoing synchronizations are complete, i.e. when all bits in SYNCBUSY are '0'.

9.5.3.5 Synchronization Delay

The synchronization will delay write and read accesses by a certain amount. This delay *D* is within the range of:

$5 \times P_{GCLK} + 2 \times P_{APB} < D < 6 \times P_{GCLK} + 3 \times P_{APB}$

Where P_{GCLK} is the period of the generic clock and P_{APB} is the period of the peripheral bus clock. A normal peripheral bus register access duration is $2 \times P_{APB}$.

10. Memories

10.1 Embedded Memories

The 32-bit physical memory address space is mapped as follows:

Table 10-1. Memory Sizes

Memory	Base Address	Size (KB)	
		PIC32CM5164	PIC32CM2532 (LE00/LS00 only)
Flash	0x00000000	512	256
Data Flash	0x00400000	16	8
SRAM	0x20000000	64	32
TrustRAM (1)	0x42005600	512 Bytes	512 Bytes
Boot ROM	0x02000000	32	

Note:

1. TrustRAM is accessible through TRAM Controller registers (RAMx).

10.1.1 Flash

The PIC32CM LE00/LS00/LS60 family of devices embed up to 512 KB of internal Flash mapped at address 0x0000 0000. The Flash has a 512-byte (64 lines of 8 bytes) direct-mapped cache which is enabled by default after power-up. The Flash is organized into rows, where each row contains four pages. The Flash has a row-erase and a page-write granularity.

⚠ CAUTION For this Flash technology, a maximum number of eight consecutive writes is allowed per row. Once this number is reached, a row erase is mandatory.

Table 10-2. Flash Memory Parameters

Device	Memory Size [KB]	Number of Rows	Row size [Bytes]	Number of Pages	Page size [Bytes]
PIC32CM5164	512	2048	256	8192	64
PIC32CM2532 (LE00/LS00 only)	256	1024		4096	

The Flash is divided in different regions and each region has a dedicated lock bit preventing from writing and erasing pages on it.

Table 10-3. Flash Lock Regions Parameters

Device	PIC32CM LE00	PIC32CM LS00/LS60
Number of Flash Lock Regions	2	3
Regions Name	Flash (BOOT region) /Flash (APPLICATION region)	Secure + NSC Flash (BOOT region) Secure + NSC Flash (APPLICATION region) Non-Secure Flash (APPLICATION region)

Notes:

1. Refer to the NVM Memory Organization figures in the “NVMCTRL” chapter to get the different regions definition.
2. The regions size is configured by the Boot ROM at device start-up by reading the NVM Boot Configuration Row (BOCOR). Refer to the 13. Boot ROM chapter for more information.

10.1.2 Data Flash

The PIC32CM LE00/LS00/LS60 family of devices embed up to16 KB of internal Data Flash with Write-While-Read (WWR) capability mapped at address 0x0040 0000.

The Data Flash can be programmed or erased while reading the Flash memory. It is not possible to read the Data Flash while writing or erasing the Flash.
Note: The Data Flash memory can be executable, but requires more cycles to be read which may affect system performance.

The Data Flash cannot be cached.

The Data Flash is organized into rows, where each row contains four pages.

The Data Flash has a row-erase and a page-write granularity.

⚠ CAUTION For this Flash technology, a maximum number of 8 consecutive writes is allowed per row. Once this number is reached, a row erase is mandatory.

Table 10-4. Data Flash Memory Parameters

Device	Memory Size [KB]	Number of Rows	Row size [Bytes]	Number of Pages	Page size [Bytes]
PIC32CM5164	16	64	256	256	64
PIC32CM2532 (LE00/LS00 only)	8	32		128	

The Data Flash is divided into one or two regions.

Each region has a dedicated lock bit preventing from writing and erasing pages on it.

Table 10-5. Data Flash Lock Regions Parameters

Device	PIC32CM LE00	PIC32CM LS00/LS60
Number of Data FLASH Lock Regions	1	2
Regions Name	Data Flash	Secure Data Flash/Non-Secure Data Flash

Notes:

1. Refer to the NVM Memory Organization figures in the chapter “NVMCTRL” to obtain the definitions of the different regions definition.
2. The regions size is configured by the Boot ROM at device start-up by reading the NVM Boot Configuration Row (BOCOR). Refer to the chapter 13. Boot ROM for additional information.

10.1.3 SRAM

The PIC32CM LE00/LS00/LS60 family of devices embed up to 64 KB of internal SRAM mapped at address 0x2000 0000.

Table 10-6. SRAM Memory Parameters

Device	Memory Size [KB]
PIC32CM5164	64
PIC32CM2532 (LE00/LS00 only)	32

Important: Before jumping to the Flash, the Boot ROM resets the first 4 KB of SRAM.

SRAM is composed of 16 KB sub-blocks which can be retained or not in Standby Low-Power mode to optimize power consumption.

By default, all sub-blocks are retained, but it is possible to switch them off using the Power Manager (PM).

SRAM retention is guaranteed for Watchdog, External and System Reset resets.

Important: SRAM retention is not guaranteed after Power Supply Resets (POR, BOD12, BOD12PLL and BOD33).

10.1.4 TrustRAM

PIC32CM LE00/LS00/LS60 devices embed an additional 512 bytes TrustRAM with physical protection features.

Important: Before jumping to the Flash, the Boot ROM resets the first 60 bytes of the TrustRAM.

Note: Refer to 30. TrustRAM (TRAM) chapter for more details.

10.1.5 Boot ROM

PIC32CM LE00/LS00/LS60 devices embed 32 KB of internal ROM mapped at address 0x0200 0000.

Note: For additional information, refer to the section 13. Boot ROM chapter.

10.2 NVM Configuration Rows

PIC32CM LE00 and PIC32CM LS00/LS60 have different Non Volatile Memory (NVM) Configuration rows which contain device configuration data that can be used by the system:

Table 10-7. NVM Configuration Rows Mapping

NVM Configuration Rows	Address
User Row (UROW)	0x00804000
Software Calibration Row	0x00806020
Boot Configuration Row (BOCOR)	0x0080C000

10.2.1 NVM User Row (UROW)

The Non Volatile Memory User Row (UROW) contains device configuration data that are automatically read at device power-on.

This row can be updated using the NVMCTRL peripheral.

When writing to the NVM User Row, the new values are not loaded by the other peripherals on the device until a device reset occurs.

The NVM User Row can be read at the address 0x00804000.

PIC32CM LE00/LS00/LS60 have different NVM User Row mappings.

10.2.1.1 PIC32CM LE00 User Row

Table 10-8. PIC32CM LE00 UROW Bitfields Definition

Bit Pos.	Name	Usage	Value(1)	Related Peripheral Register
2:0	SULCK	NVM UnLock Bits (BOOTPROT/BS)	0x7	NVMCTRL.SULCK
5:3	NSULCK	NVM UnLock Bits (ANS, DNS)	0x6	NVMCTRL.NSULCK
6	Reserved	Reserved	All '1's	Reserved
12:7	BOD33_LEVEL	BOD33 threshold level at power-on	0x4	SUPC.BOD33
13	BOD33_DIS	BOD33 Disable at power-on	0x0 (<=> BOD33 enabled)	SUPC.BOD33
15:14	BOD33_ACTION	BOD33 Action at power-on	0x1	SUPC.BOD33
24:16	BOD12/BOD12PLL Calibration Parameters	**DO NOT CHANGE(See Note 1 under Caution)**	**To Read**	Reserved
25	WDT_RUNSTDBY	WDT Runstdby at power-on	0x0	WDT.CTRLA
26	WDT_ENABLE	WDT Enable at power-on	0x0	WDT.CTRLA
27	WDT_ALWAYSON	WDT Always-On at power-on	0x0	WDT.CTRLA
31:28	WDT_PER	WDT Period at power-on	0xB	WDT.CONFIG
35:32	WDT_WINDOW	WDT Window mode time-out at power-on	0xB	WDT.CONFIG
39:36	WDT_EWOFFSET	WDT Early Warning Interrupt Time Offset at power-on	0xB	WDT.EWCTRL
40	WDT_WEN	WDT Timer Window Mode Enable at power-on	0x0	WDT.CTRLA
41	BOD33_HYST	BOD33 Hysteresis configuration at power-on	0x0	SUPC.BOD33
287:42	Reserved	Reserved	Reserved	Reserved

Note:

1. Fresh from factory value or after a ChipErase_ALL command.

⚠ CAUTION

1. BOD12 and BOD12PLL are calibrated in production and their calibration parameters must not be changed to ensure the correct device behavior.

Table 10-9. PIC32CM LE00 UROW Mapping

Offset	Bit Pos.	Name						
0x00	7:0	BOD33_LEVEL	-	NSULCK			SULCK	
0x01	15:8	BOD33_ACTION		BOD33_DIS	BOD33_LEVEL			
0x02	23:16	BOD12/BOD12PLL Calibration Parameters						
0x03	31:24	WDT_PER			WDT_ALWAYSON	WDT_ENABLE	WDT_RUNSTDBY	BOD12/BOD12PLL Calibration Parameters
0x04	39:32	WDT_EWOFFSET			WDT_WINDOW			
0x05	47:40	Reserved					BOD33_HYST	WDT_WEN
0x06-0x23	287:48	Reserved						

10.2.1.2 PIC32CM LS00 User Row

Table 10-10. PIC32CM LS00 UROW Bitfields Definition

Bit Pos.	Name	Usage	Value(1)	Related Peripheral Register
2:0	SULCK	NVM Secure Region UnLock Bits	0x7	NVMCTRL.SULCK
5:3	NSULCK	NVM Non-Secure Region UnLock Bits	0x6	NVMCTRL.NSULCK
6	Reserved	Reserved	All '1's	Reserved
12:7	BOD33_LEVEL	BOD33 threshold level at power-on.	0x4	SUPC.BOD33
13	BOD33_DIS	BOD33 Disable at power-on	0x0 (<=> BOD33 enabled)	SUPC.BOD33
15:14	BOD33_ACTION	BOD33 Action at power-on	0x1	SUPC.BOD33
24:16	BOD12/BOD12PLL Calibration Parameters	**Do not change(See Note 1 under Caution)**	**To Read**	Reserved
25	WDT_RUNSTDBY	WDT Runstdby at power-on	0x0	WDT.CTRLA
26	WDT_ENABLE	WDT Enable at power-on	0x0	WDT.CTRLA
27	WDT_ALWAYSON	WDT Always-On at power-on	0x0	WDT.CTRLA
31:28	WDT_PER	WDT Period at power-on	0xB	WDT.CONFIG
35:32	WDT_WINDOW	WDT Window mode time-out at power-on	0xB	WDT.CONFIG
39:36	WDT_EWOFFSET	WDT Early Warning Interrupt Time Offset at power-on	0xB	WDT.EWCTRL
40	WDT_WEN	WDT Timer Window Mode Enable at power-on	0x0	WDT.CTRLA
41	BOD33_HYST	BOD33 Hysteresis configuration at power-on	0x0	SUPC.BOD33
42	Reserved	Reserved	0x0	Reserved
43	RXN	RAM is eXecute Never	0x0	IDAU.SECCTRL
44	DXN	Data Flash is eXecute Never	0x1	NVMCTRL.SECCTRL
63:45	Reserved	Reserved	All '1's	Reserved
74:64	AS (6)	Secure Flash (AS region) Size = AS*0x100	0x7FF	IDAU.SCFGA
83:75	ANSC	Non-Secure Callable Flash (APPLICATION region) Size = ANSC*0x20	0x000	IDAU.SCFGA
85:84	Reserved	Reserved	All '1's	Reserved
92:86	DS	Secure Data Flash Size = DS*0x100	0x40	IDAU.SCFGA
95:93	Reserved	Reserved	All '1's	Reserved
104:96	RS	Secure SRAM Size = RS*0x80	0x1FF	IDAU.SCFGR
105	Reserved	Reserved	All '1's	Reserved
106	URWEN	User Row Write Enable	0x1	NVMCTRL.SCFGAD
127:107	Reserved	Reserved	All '1's	Reserved
159:128	NONSECA(2)	Peripherals Non-Secure Status Fuses for Bridge A	0x00000000	PAC.NONSECA
191:160	NONSECB(3,4,5)	Peripherals Non-Secure Status Fuses for Bridge B	0x00000000	PAC.NONSECB
223:192	NONSECC	Peripherals Non-Secure Status Fuses for Bridge C	0x00000000	PAC.NONSECC
255:224	CDIROFFSET	DICE CDI SRAM Offset	0x00000000	Boot ROM
287:256	USERCRC	CRC of NVM User Row bits 255:64	0xe87673b6	Boot ROM

Notes:

1. Fresh from factory value or after a ChipErase_ALL command.
2. The PAC Peripheral is always secured if BOCOR.SECCFGLOCK = 1 after exiting the Boot ROM.
3. The IDAU peripheral is always secured regardless of its bit value.
4. The NVMCTRL Peripheral is always secured if BOCOR.SECCFGLOCK = 1 after exiting the Boot ROM
5. The DSU peripheral is always non-secured regardless of its bit value.
6. Secure Flash (AS region) = Secure Flash (APPLICATION region) + Non-Secure Callable Flash (APPLICATION region)

⚠ CAUTION

1. BOD12 and BOD12PLL are calibrated in production and their calibration parameters must not be changed to ensure the correct device behavior.

Table 10-11. PIC32CM LS00 UROW Mapping

Offset	Bit Pos.	Name						
0x00	7:0	BOD33_LEVEL	-	NSULCK			SULCK	
0x01	15:8	BOD33_ACTION		BOD33_DIS	BOD33_LEVEL			
0x02	23:16	BOD12/BOD12PLL Calibration Parameters						
0x03	31:24	WDT_PER			WDT_ALWAYSON	WDT_ENABLE	WDT_RUNSTDBY	BOD12/BOD12PLL Calibration Parameters
0x04	39:32	WDT_EWOFFSET			WDT_WINDOW			
0x05	47:40	Reserved		DXN	RXN	Reserved	BOD33_HYST	WDT_WEN
0x06	55:48	Reserved						
0x07	63:56	Reserved						
0x08	71:64	AS						
0x09	79:72	ANSC				AS		
0x0A	87:80	DS		Reserved		ANSC		
0x0B	95:88	Reserved		DS				
0x0C	103:96	RS						
0x0D	111:104	Reserved				URWEN	Reserved	RS
0x0E-0xF	127:112	Reserved						
0x10-0x13	159:128	NONSECA						
0x14-0x17	191:160	NONSECB						
0x18-0x1B	223:192	NONSECC						
0x1C-0x1F	255:224	CDIROFFSET						
0x20-0x23	287:256	USERCRC						

10.2.1.3 PIC32CM LS60 User Row

Table 10-12. PIC32CM LS60 UROW Bitfields Definition

Bit Pos.	Name	Usage	Value(1)	Related Peripheral Register
2:0	SULCK	NVM Secure Region UnLock Bits	0x7	NVMCTRL.SULCK
5:3	NSULCK	NVM Non-Secure Region UnLock Bits	0x6	NVMCTRL.NSULCK
6	Reserved	Reserved	All '1's	Reserved
12:7	BOD33_LEVEL	BOD33 threshold level at power-on.	0x6	SUPC.BOD33
13	BOD33_DIS	BOD33 Disable at power-on	0x0 (<=> BOD33 enabled)	SUPC.BOD33

.....continued

Bit Pos.	Name	Usage	Value(1)	Related Peripheral Register
15:14	BOD33_ACTION	BOD33 Action at power-on	0x1	SUPC.BOD33
24:16	BOD12/BOD12PLL Calibration Parameters	**Do not change(See Note 1 under Caution)**	**To Read**	Reserved
25	WDT_RUNSTDBY	WDT Runstdby at power-on	0x0	WDT.CTRLA
26	WDT_ENABLE	WDT Enable at power-on	0x0	WDT.CTRLA
27	WDT_ALWAYSON	WDT Always-On at power-on	0x0	WDT.CTRLA
31:28	WDT_PER	WDT Period at power-on	0xB	WDT.CONFIG
35:32	WDT_WINDOW	WDT Window mode time-out at power-on	0xB	WDT.CONFIG
39:36	WDT_EWOFFSET	WDT Early Warning Interrupt Time Offset at power-on	0xB	WDT.EWCTRL
40	WDT_WEN	WDT Timer Window Mode Enable at power-on	0x0	WDT.CTRLA
41	BOD33_HYST	BOD33 Hysteresis configuration at power-on	0x0	SUPC.BOD33
42	Reserved	Reserved	All '1's	Reserved
43	RXN	RAM is eXecute Never	0x0	IDAU.SECCTRL
44	DXN	Data Flash is eXecute Never	0x1	NVMCTRL.SECCTRL
63:45	Reserved	Reserved	All '1's	Reserved
74:64	AS (6)	Secure Flash (AS region) Size = AS*0x100	0x7FF	IDAU.SCFGA
83:75	ANSC	Non-Secure Callable Flash (APPLICATION region) Size = ANSC*0x20	0x000	IDAU.SCFGA
85:84	Reserved	Reserved	All '1's	Reserved
92:86	DS	Secure Data Flash Size = DS*0x100	0x40	IDAU.SCFGA
95:93	Reserved	Reserved	All '1's	Reserved
104:96	RS	Secure SRAM Size = RS*0x80	0x1FF	IDAU.SCFGR
105	Reserved	Reserved	All '1's	Reserved
106	URWEN	User Row Write Enable	0x1	NVMCTRL.SCFGAD
127:107	Reserved	Reserved	All '1's	Reserved
159:128	NONSECA(2)	Peripherals Non-Secure Status Fuses for Bridge A	0x0000_0000	PAC.NONSECA
191:160	NONSECB(3,4,5)	Peripherals Non-Secure Status Fuses for Bridge B	0x0000_0000	PAC.NONSECB
223:192	NONSECC	Peripherals Non-Secure Status Fuses for Bridge C	0x0000_0000	PAC.NONSECC
255:224	CDIROFFSET	DICE CDI SRAM Offset	0x00000000	Boot ROM
287:256	USERCRC	CRC of NVM User Row bits 255:64	0xe87673b6	Boot ROM

Notes:

1. Fresh from factory value or after a ChipErase_ALL command.
2. The PAC peripheral is always secured if BOCOR.SECCFGLOCK = 1 after exiting the Boot ROM.
3. The IDAU peripheral is always secured regardless of its bit value.
4. The NVMCTRL peripheral is always secured if BOCOR.SECCFGLOCK = 1 after exiting the Boot ROM
5. The DSU peripheral is always non-secured regardless of its bit value.
6. Secure Flash (AS region) = Secure Flash (APPLICATION region) + Non-Secure Callable Flash (APPLICATION region)

⚠ CAUTION

1. BOD12 and BOD12PLL are calibrated in production and their calibration parameters must not be changed to ensure the correct device behavior.

Table 10-13. PIC32CM LS60 UROW Mapping

Offset	Bit Pos.	Name						
0x00	7:0	BOD33_LEVEL	-	NSULCK			SULCK	
0x01	15:8	BOD33_ACTION		BOD33_DIS	BOD33_LEVEL			
0x02	23:16	BOD12/BOD12PLL Calibration Parameters						
0x03	31:24	WDT_PER			WDT_ALWAYSON	WDT_ENABLE	WDT_RUNSTDBY	BOD12/BOD12PLL Calibration Parameters
0x04	39:32	WDT_EWOFFSET			WDT_WINDOW			
0x05	47:40	Reserved		DXN	RXN	Reserved	BOD33_HYST	WDT_WEN
0x06	55:48	Reserved						
0x07	63:56	Reserved						
0x08	71:64	AS						
0x09	79:72	ANSC				AS		
0x0A	87:80	DS		Reserved		ANSC		
0x0B	95:88	Reserved		DS				
0x0C	103:96	RS						
0x0D	111:104	Reserved				URWEN	Reserved	RS
0x0E-0xF	127:112	Reserved						
0x10-0x13	159:128	NONSECA						
0x14-0x17	191:160	NONSECB						
0x18-0x1B	223:192	NONSECC						
0x1C-0x1F	255:224	CDIROFFSET						
0x20-0x23	287:256	USERCRC						

10.2.2 NVM Software Calibration Row

The NVM Software Calibration Row contains calibration data that can be used by some peripherals, such as the ADC.

Note: Calibration data are determined and written during production and cannot be written.

The NVM Software Calibration Row can be read at address 0x00806020.

Table 10-14. NVM Software Calibration Bitfields Definition

Bit Position	Name	Description
2:0	ADC BIASREFBUF	ADC Bias Reference Buffer Scaling. Must be written to ADC CALIB.BIASREFBUF.
5:3	ADC BIASCOMP	ADC Bias Comparator Scaling. Must be written to ADC CALIB.BIASCOMP.
8:6	DFLLULP Division Factor in PL0	DFLLULP Division Factor in PL0. Must be written to OSCCTRL DFLLULPCTRL.DIV.
11:9	DFLLULP Division Factor in PL2	DFLLULP Division Factor in PL2. Must be written to OSCCTRL DFLLULPCTRL.DIV.
16:12	USB TRANSN	USB pad calibration (NMOS output impedance of DP/DM drivers). Must be written to USB PADCAL.TRANSN
17:21	USB TRANSP	USB pad calibration (PMOS output impedance of DP/DM drivers).. Must be written to USB PADCAL.TRANSP
22:24	USB TRIM	USB pad calibration (DP/DM rise/fall matching). Must be written to USB PADCAL.TRIM
25:30	DFLL48M COARSE	DFLL48M Coarse Calibration. Must be written to OSCCTRL DFLLVAL.COARSE
31	Reserved	Reserved

Table 10-15. NVM Software Calibration Row Mapping

Offset	Bit Pos.	Name							
0x00	7:0	DFLLULP Division Factor in PL0			ADC BIASCOMP			ADC BIASREFBUF	
0x01	15:8	USB TRANSN					DFLLULP Division Factor in PL2		DFLLULP Division Factor in PL0
0x02	23:16	USB TRIM			USB TRANSP				USB TRANSN
0x03	31:24	Reserved	DFLL48M COARSE						USB TRIM
0x04-0x7	63:32	Reserved							

10.2.3 NVM Boot Configuration Row (BOCOR)

The Non-Volatile Memory Boot Configuration Row (BOCOR) contains device configuration data that are automatically read by the Boot ROM program at device startup.

This row can be updated using the NVMCTRL peripheral.

When writing to the NVM Boot Configuration Row, the new values are not loaded by the other peripherals on the device until a device reset occurs.

The NVM Boot Configuration Row can be read at address 0x0080C000.

PIC32CM LE00 and PIC32CM LS00/LS60 have different NVM Boot Configuration Row mappings.

10.2.3.1 PIC32CM LE00 Boot Configuration Row

Table 10-16. PIC32CM LE00 BOCOR Bitfields Definition

Bit Pos.	Name	Usage	Value (1)	Related Peripheral Register
39:0	Reserved	Reserved	Reserved	Reserved
50:40	BOOTPROT	Boot Protection size = BOOTPROT*0x100	0x000	Boot ROM
511:51	Reserved	Reserved	Reserved	Reserved
639:512	CRCKEY	CRC Key	All '1's	Boot ROM
2047:640	Reserved	Reserved	Reserved	Reserved

Note:

1. Fresh from Factory Value or after a ChipErase_ALL command.

Table 10-17. PIC32CM LE00 BOCOR Mapping

Offset	Bit Pos.	Name	
0x00-0x04	39:0	Reserved	
0x05	47:40	BOOTPROT	
0x06	55:48	Reserved	BOOTPROT
0x07-0x0B	95:56	Reserved	
0x0C-0x0F	127:96	Reserved	
0x10-0x3F	511:128	Reserved	
0x40-0x4F	639:512	CRCKEY	
0x50-0xFF	2047:640	Reserved	

10.2.3.2 PIC32CM LS00 Boot Configuration Row

Table 10-18. PIC32CM LS00 BOCOR Bitfields Definition

Bit Pos.	Name	Usage	Value (1)	Related Peripheral Register
18:0	Reserved	Reserved	All 1s	Reserved
27:19	BNSC	Non-Secure Callable Flash (BOOT region) Size = BNSC*0x20	0x000	IDAU.SCFGB

.....continued

Bit Pos.	Name	Usage	Value (1)	Related Peripheral Register
31:28	Reserved	Reserved	All 1s	Reserved
39:32	BOOTOPT	Boot Option	0x00	Boot ROM
50:40	BOOTPROT	Boot Protection size = BOOTPROT*0x100	0x000	IDAU.SCFGB
51	SECCFGLOCK	Security Configuration Lock	0x1	Boot ROM
52	DICEEN	DICE Enable	0x0	Boot ROM
55:53	Reserved	Reserved	All 1s	Reserved
56	BCWEN	Boot Configuration Write Enable	0x1	NVMCTRL.SCFGB
57	BCREN	Boot Configuration Read Enable	0x1	NVMCTRL.SCFGB
63:58	Reserved	Reserved	All 1s	Reserved
95:64	BOCORCRC	Boot Configuration CRC for bit 63:0	0xc0349acc	Boot ROM
127:96	Reserved	Reserved	All 1s	Reserved
255:128	CEKEY0	Chip Erase Key 0	All 1s	Boot ROM
383:256	CEKEY1	Chip Erase Key 1	All 1s	Boot ROM
511:384	CEKEY2	Chip Erase Key 2	All 1s	Boot ROM
639:512	CRCKEY	CRC Key	All 1s	Boot ROM
895:640	BOOTKEY	SHA- or HMAC-based Secure Boot Key	All 1s	Boot ROM
1151:896	UDS (2)	DICE UDS Key	All 1s (unless provisioned)(2)	Boot ROM
1791:1152	Reserved	Reserved	All 1s	Reserved
2047:1792	BOCORHASH	Boot Configuration Row Hash	All 1s	Boot ROM

Notes:

1. Fresh from Factory Value or after a ChipErase_ALL command.
2. UDS must be provisioned if DICE is enabled (BOCOR.DICEEN=1). A ChipErase_ALL (CE2) will reset these bits to All 1s. Refer to DICE section in Boot ROM chapter for more information.

Table 10-19. PIC32CM LS00 BOCOR Mapping

Offset	Bit Pos.	Name							
0x00-0x1	15:0	Reserved							
0x02	23:16	BNSC						Reserved	
0x03	31:24	Reserved						BNSC	
0x04	39:32	BOOTOPT							
0x05	47:40	BOOTPROT							
0x06	55:48	Reserved			DICEEN	SECCFGLOCK	BOOTPROT		
0x07	63:56	Reserved						BCREN	BCWEN
0x08-0x0B	95:64	BOCORCRC							
0x0C-0x0F	127:96	Reserved							
0x10-0x1F	255:128	CEKEY0							
0x20-0x2F	383:256	CEKEY1							
0x30-0x3F	511:384	CEKEY2							
0x40-0x4F	639:512	CRCKEY							
0x50-0x6F	895:640	BOOTKEY							
0x70-0x8F	1151:896	UDS							
0x90-0xDF	1791:1152	Reserved							
0xE0-0xFF	2047:1792	BOCORHASH							

10.2.3.3 PIC32CM LS60 Boot Configuration Row

Table 10-20. PIC32CM LS60 BOCOR Bitfields Definition

Bit Pos.	Name	Usage	Value (1)	Related Peripheral Register
18:0	Reserved	Reserved	All 1s	Reserved
27:19	BNSC	Non-Secure Callable Flash (BOOT region) Size = BNSC*0x20	0x000	IDAU.SCFGB
31:28	Reserved	Reserved	All 1s	Reserved
39:32	BOOTOPT	Boot Option	0x00	Boot ROM
50:40	BOOTPROT	Boot Protection size = BOOTPROT*0x100	0x000	IDAU.SCFGB
51	SECCFGLOCK	Security Configuration Lock	0x1	Boot ROM
52	DICEEN	DICE Enable	0x0	Boot ROM
55:53	Reserved	Reserved	All 1s	Reserved
56	BCWEN	Boot Configuration Write Enable	0x1	NVMCTRL.SCFGB
57	BCREN	Boot Configuration Read Enable	0x1	NVMCTRL.SCFGB
63:58	Reserved	Reserved	All 1s	Reserved
95:64	BOCORCRC	Boot Configuration CRC for bit 63:0	0xc0349acc	Boot ROM
127:96	Reserved	Reserved	All 1s	Reserved
255:128	CEKEY0	Chip Erase Key 0	All 1s	Boot ROM
383:256	CEKEY1	Chip Erase Key 1	All 1s	Boot ROM
511:384	CEKEY2	Chip Erase Key 2	All 1s	Boot ROM
639:512	CRCKEY	CRC Key	All 1s	Boot ROM
895:640	BOOTKEY	SHA- or HMAC-based Secure Boot Key	All 1s	Boot ROM
1151:896	UDS (2)	DICE UDS Key	All 1s (unless provisioned)(2)	Boot ROM
1407:1152	IOPROTKEY(3)	ATECC608B I/O Protection Key	All 1s (unless provisioned)(3)	Boot ROM
1791:1408	Reserved	Reserved	All 1s	Reserved
2047:1792	BOCORHASH	Boot Configuration Row Hash	All 1s	Boot ROM

Notes:

1. Fresh from Factory Value or after a ChipErase_ALL command.
2. UDS must be provisioned if DICE is enabled (BOCOR.DICEEN=1). A ChipErase_ALL (CE2) will reset these bits to All 1s. Refer to DICE section in Boot ROM chapter for more information.
3. IOPROTKEY must be provisioned if ATECC608B-based secure boot is enabled (BOCOR.BOOTOPT > 4). A ChipErase_ALL (CE2) will reset these bits to All 1s.

Table 10-21. PIC32CM LS60 BOCOR Mapping

Offset	Bit Pos.	Name						
0x00-0x1	15:0	Reserved						
0x02	23:16	BNSC					Reserved	
0x03	31:24	Reserved				BNSC		
0x04	39:32	BOOTOPT						
0x05	47:40	BOOTPROT						
0x06	55:48	Reserved		DICEEN	SECCFGLOCK	BOOTPROT		
0x07	63:56	Reserved					BCREN	BCWEN
0x08-0x0B	95:64	BOCORCRC						
0x0C-0x0F	127:96	Reserved						
0x10-0x1F	255:128	CEKEY0						
0x20-0x2F	383:256	CEKEY1						

.....continued

Offset	Bit Pos.	Name
0x30-0x3F	511:384	CEKEY2
0x40-0x4F	639:512	CRCKEY
0x50-0x6F	895:640	BOOTKEY
0x70-0x8F	1151:896	UDS
0x90-0xAF	1407:1152	IOPROTKEY
0xB0-0xDF	1791:1408	Reserved
0xE0-0xFF	2047:1792	BOCORHASH

10.3 Serial Number

Each device has a unique 128-bit serial number which is a chain of four 32-bit words contained at the following addresses of the NVM configuration rows memory space:

- Word 0: 0x0080A00C
- Word 1: 0x0080A040
- Word 2: 0x0080A044
- Word 3: 0x0080A048

Note: The uniqueness of the serial number is only guaranteed when considering all 128 bits.

11. Processor and Architecture

11.1 Cortex-M23 Processor

The PIC32CM LE00/LS00/LS60 implement the Arm® Cortex®-M23 processor, based on the ARMv8-M Baseline Architecture, which is the smallest and most energy efficient Arm processor with Arm TrustZone® security technology.

TrustZone® for ARMv8-M provides hardware-enforced security isolation between trusted and the untrusted resources on a Cortex™-M23 based device, while maintaining the efficient exception handling.

The implemented Arm Cortex-M23 is revision r1p0.

The Arm Cortex-M23 core has the following bus interfaces:

- Single 32-bit AMBA®-5 AHB-Lite system interface that provides connections to peripherals and memories.
- Single 32-bit I/O port bus interfacing to the PORT and Crypto Accelerator peripherals with 1-cycle load and store.

Note: For more information refer to www.arm.com

11.1.1 Cortex-M23 Configuration

The following table provides the configuration for the Arm Cortex-M23 processor.

Table 11-1. PIC32CM LE00/LS00/LS60 Cortex-M23 Configuration

Features	PIC32CM LE00 Implementation	PIC32CM LS00/LS60 Implementation
Memory Protection Unit (MPU)	One MPU with 12 regions	Two MPUs with 12 regions each (one Secure/one Non-Secure)
Security Attribute Unit (SAU)	Absent	Absent
Implementation Defined Attribution Unit (IDAU)	Absent	Present
SysTick timers	One SysTick timer	Two timers (One Secure/One Non-Secure)
Vector Table Offset Register	Present (one Vector table)	Present (two Vector tables)
Reset all registers	Absent	Absent
Multiplier	Fast (one cycle)	Fast (one cycle)
Divider	Fast (17 cycles)	Fast (17 cycles)
Interrupts	71(1)	71(1)
Instruction fetch width	32-bit	32-bit
Single-cycle I/O port	Present	Present
Architectural clock gating present	Present	Present
Data endianness	Little-endian	Little-endian
Halting debug support	Present	Present
Wake-up interrupt controller (WIC)	Absent	Absent
Number of breakpoint comparators	4	4
Number of watchpoint comparators	2	2
Cross Trigger Interface (CTI)	Absent	Absent
Micro Trace Buffer (MTB)	Absent	Absent
Embedded Trace Macrocell (ETM)	Absent	Absent
JTAGnSW debug protocol	Serial-Wire	Serial-Wire
Multi-drop for Serial Wire	Absent	Absent

Note:

1. Refer to Table 11-3 for more information.

For more details, refer to the Arm Cortex-M23 Processor Technical Reference Manual (www.arm.com).

11.1.2 Cortex-M23 Core Peripherals

The processor has the following core peripherals:

- System Timer (SysTick)
  - The System Timer is a 24-bit timer clocked by the core frequency.

Important: On PIC32CM LS00/LS60 devices, there are two System timers, one for Secure state and one for Non-Secure state.

- Nested Vectored Interrupt Controller (NVIC)
  - The NVIC is an embedded interrupt controller that supports low-latency interrupt processing.

Important: On PIC32CM LS00/LS60 devices, there are two Vector tables: the Secure Vector table and the Non-Secure Vector table.

- System Control Block (SCB)
  - The System Control Block (SCB) provides system implementation information and system control that includes configuration, control, and reporting of system exceptions
- Memory Protection Unit (MPU)
  - The MPU improves system reliability by defining the memory attributes for different memory regions.

Important: On PIC32CM LS00/LS60 devices, there are two MPUs: one for the Secure state and one for the Non-Secure state. Each MPU can define memory access permissions and attributes independently.

- Security Attribution Unit (SAU)
  - The SAU improves system security by defining security attributes for different regions.

Important: The SAU is absent from PIC32CM LE00 and PIC32CM LS00/LS60 devices.

For more details, refer to the Arm Cortex-M23 Processor Technical Reference Manual (www.arm.com).

Table 11-2. Cortex-M23 Core Peripherals Address Map

Core Peripherals	Base Address (PIC32CM LE00 and PIC32CM LS00/LS60)	(Non-Secure) Alias Base Address (PIC32CM LS00/LS60 only)
System Timer (SysTick)	0xE000E010	0xE002E010
Nested Vectored Interrupt Controller (NVIC)	0xE000E100	0xE002E100
System Control Block (SCB)	0xE000ED00	0xE002ED00
Memory Protection Unit (MPU)	0xE000ED90	0xE002ED90

11.1.3 Single Cycle I/O Port

The device allows direct access to PORT registers. Accesses to the AMBA® AHB-Lite™ and the single cycle I/O interface can be made concurrently, hence the Cortex-M23 processor can fetch the next instructions while accessing the I/Os. This enables single cycle I/O access to be sustained for as long as necessary.
Note: The Crypto Accelerator peripheral also benefits from this port. Refer to the 12.3. Crypto Acceleration section for more information.

11.2 Nested Vector Interrupt Controller

11.2.1 Overview

The Nested Vectored Interrupt Controller (NVIC) in the PIC32CM LE00/LS00/LS60 supports up to 71 interrupt lines with four different priority levels + 1 Non Maskable Interrupt (NMI) line.

For more details, refer to the Cortex-M23 Technical Reference Manual (www.arm.com).

11.2.2 Interrupt Line Mapping

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

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

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

The interrupt requests for one peripheral are ORed together on system level, generating one interrupt request for each peripheral. An interrupt request will set the corresponding interrupt pending bit in the NVIC interrupt pending registers (SETPEND/CLRPEND bits in the 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-IPR11 provide a priority field for each interrupt.

Table 11-3. Interrupt Line Mapping

Peripheral Source	Peripheral Interrupt(s)	NVIC line
EIC NMI – External Interrupt Controller	NMI	NMI

.....continued

Peripheral Source	Peripheral Interrupt(s)	NVIC line
PM – Power Manager	PLRDY	0
MCLK - Main Clock	CKRDY	
OSCCTRL - Oscillators Controller	XOSCRDY	
	XOSCFAIL	
	OSC16MRDY	
	DFLLULPRDY	
	DFLLULPLOCK	
	DFLLULPNOLOCK	
	DPLLLCKR	
	DPLLLCKF	
	DPLLLTO	
	DPLLLDRTO	
	DFLLRDY	
	DFLLOOB	
	DFLLLCKF	
	DFLLLCKC	
	DFLLRCS	
OSC32KCTRL - 32.768 kHz Oscillators Controller	XOSC32KRDY	
	CLKFAIL	
SUPC - Supply Controller	BOD33RDY	
	BOD33DET	
	B33SRDY	
	VREGRDY	
	VCORERDY	
	ULPVREFRDY	
	VCOREPLLRDY	
WDT – Watchdog Timer	EW	1

.....continued

Peripheral Source	Peripheral Interrupt(s)	NVIC line
RTC – Real Time Counter	CMP0	2
	CMP1	
	PER0	
	PER1	
	PER2	
	PER3	
	PER4	
	PER5	
	PER6	
	PER7	
	OVF	
	TAMPER	
	ALARM0	
EIC – External Interrupt Controller	EXTINT0	3
	EXTINT1	4
	EXTINT2	5
	EXTINT3	6
	EXTINT4	7
	EXTINT5	8
	EXTINT6	9
	EXTINT7	10
	EXTINT8-15	11
	NSCHK(1)	
FREQM - Frequency Meter	DONE	12
NVMCTRL – Non-Volatile Memory Controller	DONE	13
	PROGE	
	LOCKE	
	NVME	
	KEYE	
	NSCHK(1)	
PORT - I/O Pin Controller	NSCHK(1)	14

.....continued

Peripheral Source	Peripheral Interrupt(s)	NVIC line
DMAC - Direct Memory Access Controller	SUSP0	15
	TERR0	
	TCMPL0	
	SUSP1	16
	TERR1	
	TCMPL1	
	SUSP2	17
	TERR2	
	TCMPL2	
	SUSP3	18
	TERR3	
	TCMPL3	
	SUSP4-15	19
	TERR4-15	
	TCMPL4-15	
USB	EORSM/DNRSM	20
	EORST/RST	
	LPMNYET/DCONN	
	LPMSUSP/DDISC	
	RAMACER	
	RXSTP/TXSTP	
	SOF/HSOF	
	STALL0/STALL	
	STALL1	
	SUSPEND	
	TRCPT0	
	TRCPT1	
	TRFAIL0/TRFAIL	
	TRFAIL1/PERR	
	UPRSM	
	WAKEUP	

.....continued

Peripheral Source	Peripheral Interrupt(s)	NVIC line
EVSYS – Event System	EVD0	21
	OVR0	
	EVD1	22
	OVR1	
	EVD2	23
	OVR2	
	EVD3	24
	OVR3	
	EVD4	25
	OVR4	
	EVD5	26
	OVR5	
	EVD6	27
	OVR6	
	EVD7	28
	OVR7	
	NSCHK(1)	29
PAC - Peripheral Access Controller	ERR	30
SERCOM0 – Serial Communication Interface 0 (Interrupt Sources vary depending on SERCOM mode)	Interrupt Bit 0 (DRE, MB, PREC)	31
	Interrupt Bit 1 (TXC, SB, AMATCH)	32
	Interrupt Bit 2 (RXC, DRDY)	33
	Interrupt Bits 3-7 (RXS, TXFE, SSL, CTSIC, RXFF, RXBRK, ERROR)	34
SERCOM1 – Serial Communication Interface 1 (Interrupt Sources vary depending on SERCOM mode)	Interrupt Bit 0 (DRE, MB, PREC)	35
	Interrupt Bit 1 (TXC, SB, AMATCH)	36
	Interrupt Bit 2 (RXC, DRDY)	37
	Interrupt Bit 3-7 (RXS, TXFE, SSL, CTSIC, RXFF, RXBRK, ERROR)	38
SERCOM2 – Serial Communication Interface 2 (Interrupt Sources vary depending on SERCOM mode)	Interrupt Bit 0 (DRE, MB, PREC)	39
	Interrupt Bit 1 (TXC, SB, AMATCH)	40
	Interrupt Bit 2 (RXC, DRDY)	41
	Interrupt Bits 3-7 (RXS, TXFE, SSL, CTSIC, RXFF, RXBRK, ERROR)	42
SERCOM3 – Serial Communication Interface 0 (Interrupt Sources vary depending on SERCOM mode)	Interrupt Bit 0 (DRE, MB, PREC)	43
	Interrupt Bit 1 (TXC, SB, AMATCH)	44
	Interrupt Bit 2 (RXC, DRDY)	45
	Interrupt Bits 3-7 (RXS, TXFE, SSL, CTSIC, RXFF, RXBRK, ERROR)	46

.....continued

Peripheral Source	Peripheral Interrupt(s)	NVIC line
SERCOM4 – Serial Communication Interface 1 (Interrupt Sources vary depending on SERCOM mode)	Interrupt Bit 0 (DRE, MB, PREC)	47
	Interrupt Bit 1 (TXC, SB, AMATCH)	48
	Interrupt Bit 2 (RXC, DRDY)	49
	Interrupt Bit 3-7 (RXS, TXFE, SSL, CTSIC, RXFF, RXBRK, ERROR)	50
SERCOM5 – Serial Communication Interface 2 (Interrupt Sources vary depending on SERCOM mode)	Interrupt Bit 0 (DRE, MB, PREC)	51
	Interrupt Bit 1 (TXC, SB, AMATCH)	52
	Interrupt Bit 2 (RXC, DRDY)	53
	Interrupt Bits 3-7 (RXS, TXFE, SSL, CTSIC, RXFF, RXBRK, ERROR)	54
TC0 – Timer Counter 0	ERR	55
	MC0	
	MC1	
	OVF	
TC1 – Timer Counter 1	ERR	56
	MC0	
	MC1	
	OVF	
TC2 – Timer Counter 2	ERR	57
	MC0	
	MC1	
	OVF	
TCC0 – Timer Counter for Control 0	ERR	58
	MC0-3	
	OVF	
	TRG	
	CNT	
	DFS	
	UFS	
	FAULTA	
	FAULTB	
	FAULT0	
	FAULT1	

.....continued

Peripheral Source	Peripheral Interrupt(s)	NVIC line
TCC1 – Timer Counter for Control 1	ERR	59
	MC0-1	
	OVF	
	TRG	
	CNT	
	DFS	
	UFS	
	FAULTA	
	FAULTB	
	FAULT0	
	FAULT1	
TCC2 – Timer Counter for Control 2	ERR	60
	MC0-1	
	OVF	
	TRG	
	CNT	
	DFS	
	UFS	
	FAULTA	
	FAULTB	
	FAULT0	
	FAULT1	
TCC3 – Timer Counter for Control 3	ERR	61
	MC0-3	
	OVF	
	TRG	
	CNT	
	DFS	
	UFS	
	FAULTA	
	FAULTB	
	FAULT0	
	FAULT1	
ADC – Analog-to-Digital Converter	OVERRUN	62
	WINMON	
	RESRDY	63
AC – Analog Comparator	COMP0-3	64
	WIN0-1	
DAC – Digital-to-Analog Converter	UNDERRUN0-1	65
	EMPTY0-1	66

.....continued		
Peripheral Source	**Peripheral Interrupt(s)**	**NVIC line**
PTC – Peripheral Touch Controller	EOC	67
	WCOMP	
TRNG - True Random Number Generator	DATARDY	68
I2S - Inter-IC Sound Controller	RXRDY0-1	69
	TXRDY0-1	
	RXOR0-1	
	TXUR0-1	
TRAM - TrustRAM	DRP	70
	ERR	

Note:

1. NSCHK interrupt sources will not generate any interrupts for PIC32CM LE00 devices.

11.3 High-Speed Bus System

11.3.1 Features

The High-Speed Bus Matrix has the following features:

- 32-bit data bus
- Allows concurrent accesses from different hosts to different clients
- Operation at a one-to-one clock frequency with the bus hosts

11.3.2 Configuration

There are two Host-to-Client connections to optimize system bandwidth:

- Multi-Client Hosts which are connected through the AHB bus matrix

Table 11-4. AHB Multi-Client Hosts

AHB Multi-Client Hosts
Cortex-M23 Processor
DSU - Device Service Unit
DMAC - Direct Memory Access Controller / Data Access

Table 11-5. AHB Clients

AHB Clients
Flash Memory
AHB-APB Bridge A (APBA)
AHB-APB Bridge B (APBB)
AHB-APB Bridge C (APBC)
SRAM Port 0 - Cortex-M23 Access
SRAM Port 1 - DMAC Access
SRAM Port 2 - DSU Access
Boot ROM

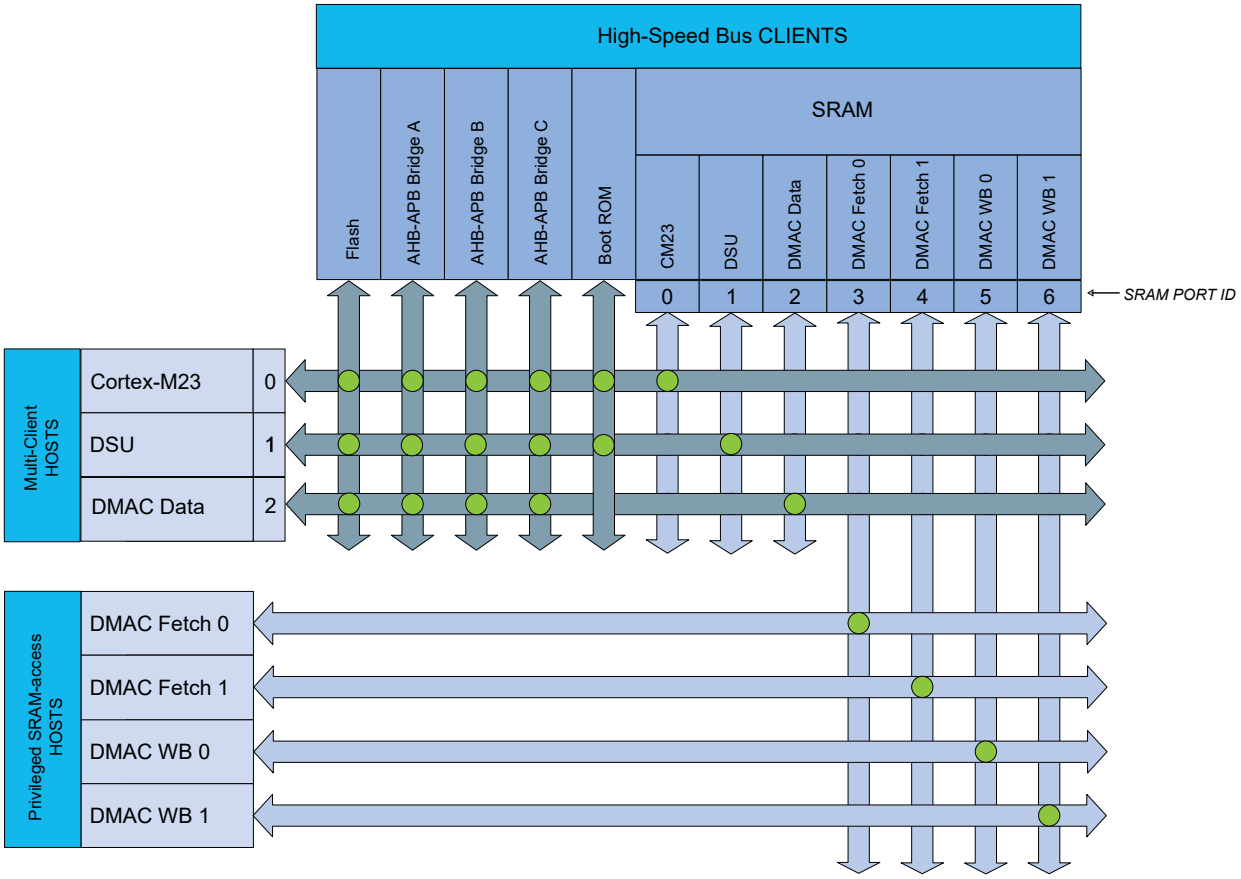
- Privileged SRAM-access Hosts which have a direct access to some dedicated SRAM ports

Table 11-6. Privileged SRAM-access Hosts

Privileged SRAM-access Hosts
DMAC - Fetch 0 Access
DMAC - Fetch 1 Access
DMAC - Write Back 0 Access
DMAC - Write Back 1 Access

Note: Privileged SRAM-access Hosts rely on SRAM quality of service to define priority levels (SRAM Port ID). Refer to 11.4. SRAM Quality of Service for more information.

Figure 11-1. Host-to-Client Access



11.4 SRAM Quality of Service

To ensure that hosts with latency requirements get sufficient priority when accessing RAM, priority levels can be assigned to the hosts for different types of access.

The Quality of Service (QoS) level is independently selected for each host accessing the RAM. For any access to the RAM, the RAM also receives a QoS level. The QoS levels and their corresponding bit values are shown in the following table.

Table 11-7. Quality of Service

Value	Name	Description
0x0	DISABLE	Background (no sensitive operation)

.....continued		
Value	**Name**	**Description**
0x1	LOW	Sensitive Bandwidth
0x2	MEDIUM	Sensitive Latency
0x3	HIGH	Critical Latency

Note: If a host is configured with QoS level DISABLE (0x0) or LOW (0x1), there will be a minimum latency of one cycle to get RAM access.

The priority order for concurrent accesses are decided by two factors:

- As first priority, the QoS level for the host.
- As a second priority, a static priority given by the port ID. The lowest port ID has the highest static priority.

See the tables below for more details.

Table 11-8. SRAM Port Connections QoS

SRAM Port Connection	Port ID	Connection Type	QoS	default QoS
USB - Universal Serial Bus	7	Direct	USB QOSCTRL.CQOS USB QOSCTRL.DQOS	0x3 0x3
DMAC - Direct Memory Access Controller - Write-Back 1 Access	6	Direct	DMAC QOSCTRL.WRBQOS	0x2
DMAC - Direct Memory Access Controller - Write-Back 0 Access	5	Direct	DMAC QOSCTRL.WRBQOS	0x2
DMAC - Direct Memory Access Controller - Fetch 1 Access	4	Direct	DMAC QOSCTRL.FQOS	0x2
DMAC - Direct Memory Access Controller - Fetch 0 Access	3	Direct	DMAC QOSCTRL.FQOS	0x2
DMAC - Direct Memory Access Controller - Data Access	2	Bus Matrix	DMAC QOSCTRL.DQOS	0x2
DSU - Device Service Unit	1	Bus Matrix	DSU CFG.LQOS	0x2
CM23 - Cortex M23 Processor	0	Bus Matrix	0x41008114, bits[1:0](1)	0x3

Note:

1. The CPU QoS level can be written/read, using 32-bit access only.

12. PIC32CM LS00/LS60 Specific Security Features

This chapter provides an overview of the security features which are specific to the PIC32CM LS00/LS60.

12.1 Features

PIC32CM LS00/LS60 Specific Security features can be divided into two main categories.

The first category relates to the Arm TrustZone for ARMv8-M technology features:

- Flexible hardware isolation of memories and peripherals:
  - Up to five regions for the Flash
  - Up to two regions for the Data Flash
  - Up to two regions for the SRAM
  - Individual security attribution (secure or non-secure) for each peripheral using the Peripheral Access Controller (PAC)
  - Mix-Secure peripherals which support both secure and non-secure security attributions
- Three debug access levels allowing:
  - The highest debug level with no restrictions in term of memory and peripheral accesses
  - A restricted debug level with non-secure memory regions access only
  - The lowest debug level where no access is authorized except with a debugger using a Boot ROM-specific mode
- Different chip erase support according to security settings
- Security configurations are fully stored in NVM Configuration rows and safely auto-loaded at start-up during Boot ROM execution using CRC checks
- Security configurations lock capability

Important: Debug access levels transitions as Chip Erase commands support are described in the Boot ROM chapter.

The second category relates to the PIC32CM LS00/LS60-specific security features, which are not related to Arm TrustZone for ARMv8-M technology support:

- Built-in cryptographic accelerator accessible through cryptographic libraries stored in ROM
  - Supporting AES-128/192/256 encryption/decryption, SHA-256 authentication, GCM encryption and authentication
- SHA- or HMAC-based Secure Boot
- ATECC608B CryptoAuthentication™ Device **(PIC32CM LS60 only)**
  - One Permanent Primary P-256 Elliptic Curve Cryptography (ECC) Private Key
  - One Internal Sign Private Key for Key Attestation
  - Three Secondary P-256 ECC Private Keys
  - Signer Public Key from Signer Certificate
  - Public Key Validation Support
  - One Customizable Symmetric Secret Key Slot
  - I/O Protection Key Slot to Protect Communication
  - Secure Boot Enabled with Customizable Secure Boot Public Key
  - ECDH/KDF Key Slot Capable of Being Used with AES Keys and Commands
  - X.509 Compressed Certificate Storage
  - Customizable Certificate Storage Slots
- Device Identity Composition Engine (DICE) security standard support with Unique Device Secret (UDS)

- Secure Pin Multiplexing to isolate on dedicated SERCOM I/O pins a secured communication with external devices from the Non-secure application **(PIC32CM LS00 only)**
- Data Flash Scrambling

The PIC32CM LS00/LS60 has other security features, which are not described in this chapter as they are common to both PIC32CM LE00 and PIC32CM LS00/LS60 such as:

- Up to eight tamper input pins and eight tamper output pins for static and dynamic intrusion detections
- One True Random Number Generator (TRNG)
- Data Flash and TrustRAM rapid tamper, silent access features
- A unique 128-bit serial number

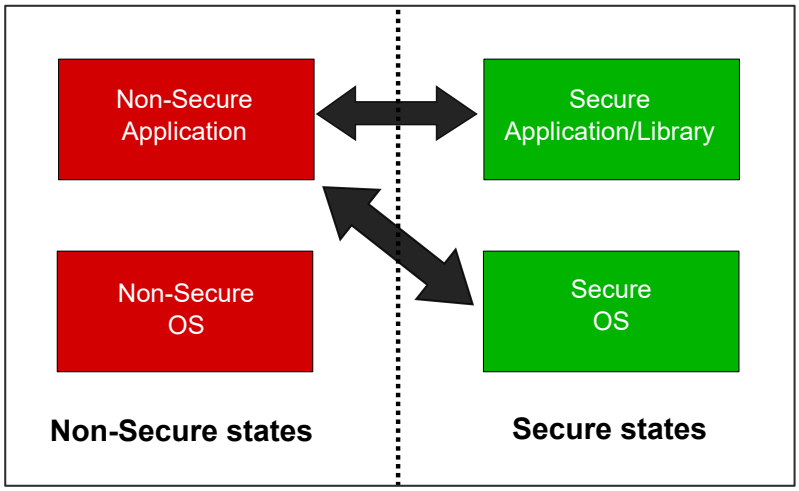
12.2 Arm TrustZone for ARMv8-M

Arm TrustZone for ARMv8-M technology is an optional core extension implemented on PIC32CM LS00/LS60 devices, which enables the system and the software to be partitioned into Secure and Non-Secure domains.

Secure software can access both Secure and Non-Secure memories and resources, while Non-Secure software can only access Non-Secure memories and resources.

Arm TrustZone for ARMv8-M allows Secure and Non-Secure code to run on a single CPU.
Note: The system always starts up in Secure state.

Figure 12-1. TrustZone for ARMv8-M



Important: For more details, refer to *TrustZone Technology for ARMv8-M Architecture*, which is available on the Arm web site (www.arm.com).

The memory space is partitioned into Non-Secure and Secure memory regions:

- **Non-Secure (NS):** Non-Secure addresses are used for memory and peripherals accessible by all software, that is, running on the device.
- **Secure (S):** Secure addresses are used for memory and peripherals accessible only by Secure software or hosts.
- **Non-Secure Callable (NSC):** NSC is a special type of Secure memory location. It allows software to transition from Non-Secure to Secure state.

12.2.1 Memories Security Attribution

12.2.1.1 Flash

The PIC32CM LS00/LS60 Flash can be split in up to five regions:

- The first two regions belongs to the BOOT region and can be configured to support a first-level bootloader for the application.
- The other regions belongs to the APPLICATION region and relate to the application itself.

The BOOT region can be split in up to two sub-regions:

- The Secure (S) sub-region
- The Non-Secure Callable (NSC) sub-region

The APPLICATION region can be split in up to three sub-regions:

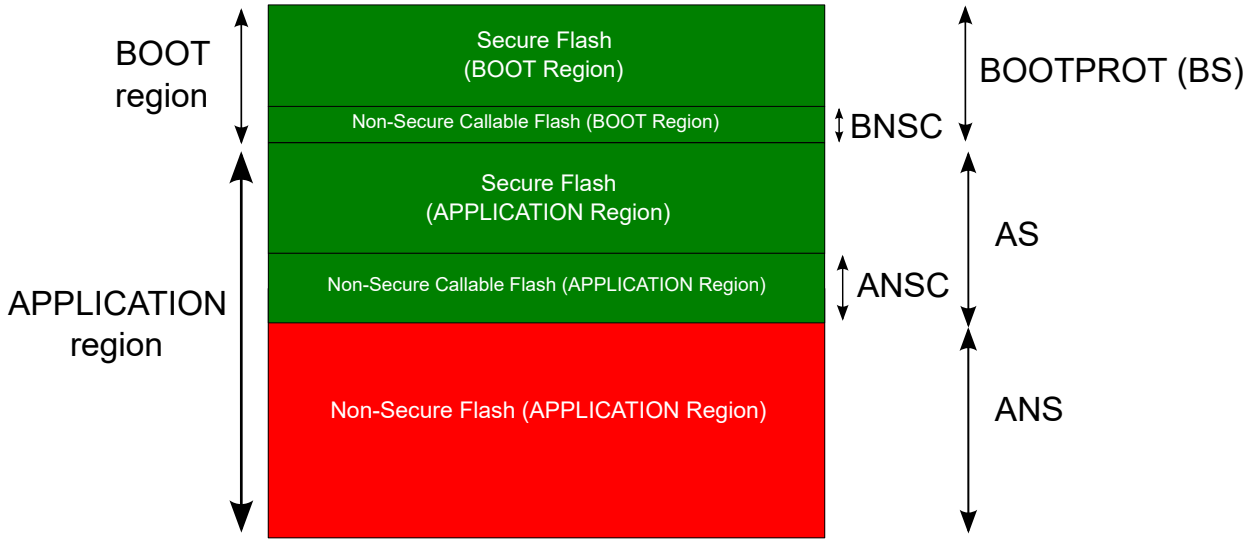
- The Secure (S) sub-region
- The Non-Secure Callable (NSC) sub-region
- The Non-Secure (NS) sub-region

Each sub-region size can be configured using the NVM User (UROW) and Boot Configuration (BOCOR) rows bit fields:

- BOCOR.BOOTPROT corresponds to the size of the (S) + (NSC) sub-regions of the BOOT region
- BOCOR.BNSC corresponds to the size of the (NSC) sub-region of the BOOT region
- UROW.AS corresponds to the size of the (S) + (NSC) sub-regions of the APPLICATION region
- UROW.ANSC corresponds to the size of the (NSC) sub-region of the APPLICATION region

Note: The remaining size of the Flash corresponds to the size of the (NS) sub-region of the APPLICATION region

Figure 12-2. PIC32CM LS00/LS60 Flash Memory Mapping



Important: Refer to "*Memory and Peripheral Security Configurations*" section for more details on the programming of BOOTPROT, BNSC, AS, and ANSC parameters.

12.2.1.2 Data Flash

The PIC32CM LS00/LS60 Data Flash can be split in up to two regions:

- The Secure Data Flash region, with a size defined by the DS NVM bit from the NVM User Row (UROW)
- The Non-Secure (NS) Data Flash region

Figure 12-3. PIC32CM LS00/LS60 Data Flash Memory Mapping

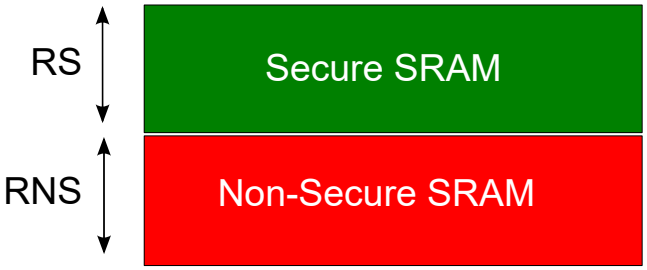
Important: Refer to "*Memory and Peripheral Security Configurations*" section for more details on the programming of DS parameter.

12.2.1.3 SRAM

The PIC32CM LS00/LS60 SRAM can be split in up to two regions:

- The Secure SRAM region, with a size defined by the RS NVM bit from the NVM User Row (UROW)
- The Non-Secure (NS) SRAM region

Figure 12-4. PIC32CM LS00/LS60 SRAM Memory Mapping



Important: Refer to "*Memory and Peripheral Security Configurations*" section for more details on the programming of RS parameter.

12.2.1.4 PIC32CM LS00/LS60 Memory Mapping Configuration Summary

The table below summarizes the mapping of the PIC32CM LS00/LS60 memory regions.

Table 12-1. PIC32CM LS00/LS60 Memory Regions Mapping

Memory region	Base address	Size
Flash BOOT region	0x00000000	BOOTPROT * 256Bytes
Secure Flash (BOOT region)	0x00000000	BOOTPROT*256Bytes - BNSC*32Bytes
Non-Secure Callable Flash (BOOT region)	Contiguous to Secure Flash (BOOT region)	BNSC * 32Bytes
Flash APPLICATION region	BOOTPROT * 256Bytes	Flash size - BOOTPROT*256Bytes
Secure Flash (APPLICATION region)	BOOTPROT * 256Bytes	AS*256Bytes-ANSC*32Bytes
Non-Secure Callable Flash (APPLICATION region)	Contiguous to Secure Flash (APPLICATION region)	ANSC * 32Bytes
Non-Secure Flash (APPLICATION region)	(BOOTPROT+AS) * 256Bytes	Flash size - (BOOTPROT*256Bytes + AS*256Bytes)

.....continued

Memory region	Base address	Size
Secure Data Flash	0x00400000	DS * 256Bytes
Non-Secure Data Flash	Contiguous to Secure Data Flash	2KB - Secure Data Flash size
Secure SRAM	0x20000000	RS * 128Bytes
Non-Secure SRAM	Contiguous to Secure SRAM	SRAM size - Secure SRAM size

Important: Refer to "*Memory and Peripheral Security Configurations*" section for more details on the programming of BOOTPROT, BNSC, AS, ANSC, DS and RS parameters.

Here are the default memories configuration from fresh from factory or after a ChipErase_ALL command.

Figure 12-5. PIC32CM LS00/LS60 Default Memories Mapping - 512KB Flash Case

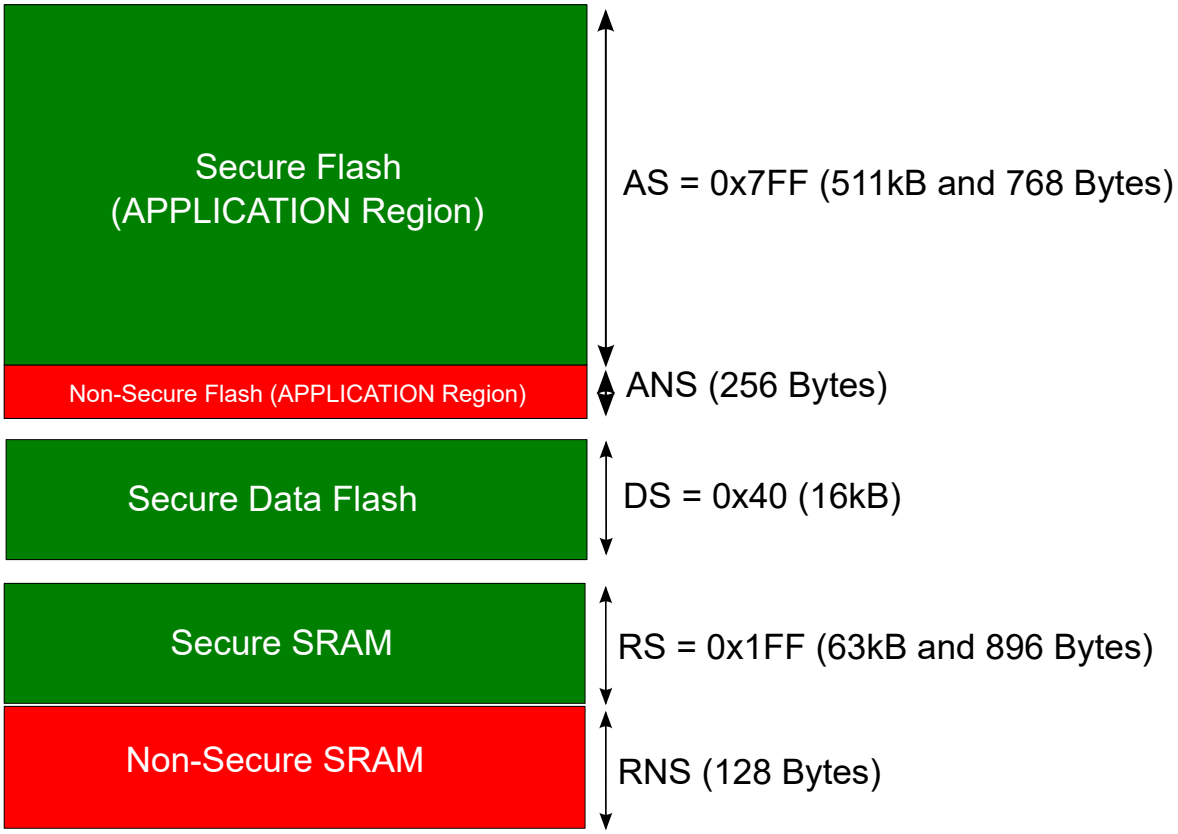
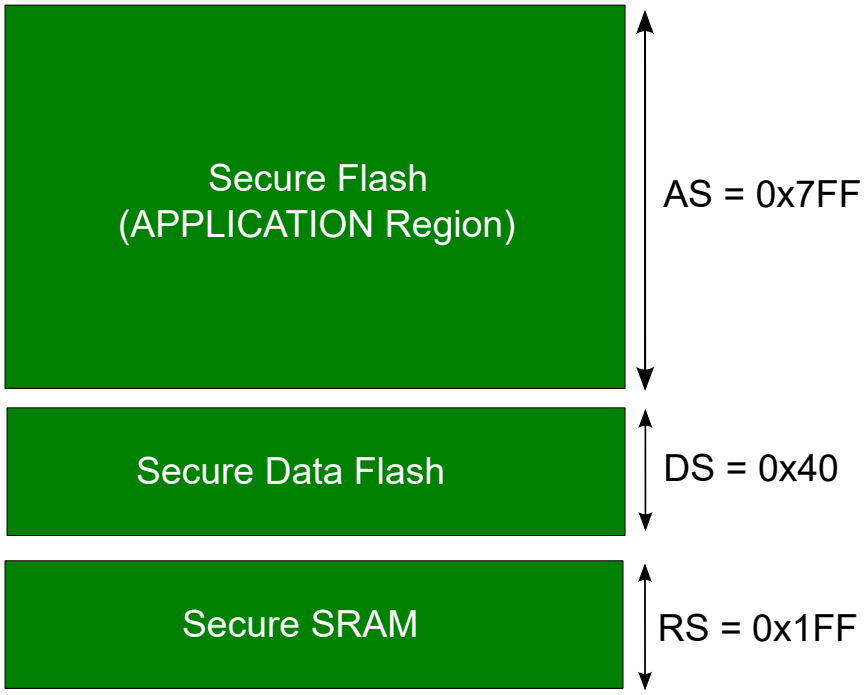


Figure 12-6. PIC32CM LS00 Default Memories Mapping - 256KB Flash Case



Here is a configuration example for a device with 512KB of Flash, 16KB of Data Flash and 64KB of SRAM:

- BOOT region:
  - Flash BOOT region size = 8KB => BOOTPROT = 8192 / 256 = 32 (0x20)
  - Non-Secure Callable Flash (BOOT region) size = 1KB => BNSC = 1024 / 32 = 32 (0x20)
  - Secure Flash (BOOT region) = 8KB - 1KB = 7KB
- APPLICATION region:
  - Flash (APPLICATION region) size = 512KB - 8KB = 504KB
  - Non-Secure Callable Flash (APPLICATION region) size = 1KB => ANSC = 1024 / 32 = 32 (0x20)
  - Secure Flash (APPLICATION region) = 15KB
    - => Secure Flash (APPLICATION region) + Non-Secure Callable Flash (APPLICATION region) size = 16KB
    - => AS * 256 = 16 * 1024 <=> AS = 64 (0x40)
  - Non-Secure Flash (APPLICATION region) size = 504KB - 16KB = 488KB
- Data Flash region:
  - Secure Data Flash size = 8KB => DS = 8192 / 256 = 32 (0x20)
  - Non-Secure Data Flash size = 16KB - 8KB = 8KB
- SRAM region:
  - Secure SRAM size = 32KB => RS = 32768 / 128 = 256 (0x100)
  - Non-Secure SRAM size = 64KB - 32KB = 32KB

12.2.2 Peripherals Security Attribution

Security configurations related to the peripherals are handled by the PAC Controller.

Each peripheral can only be configured either in Secure or in Non-Secure mode except the IDAU peripheral which is always Secured.

The security configuration (Secure or Non-Secure) is propagated to each individual peripheral, thus it is the responsibility of the peripheral to grant or not the access with the following rules:

- If the peripheral is configured as Non-Secure in the PAC:
  - Secure and Non-Secure accesses are granted
- If the peripheral is configured as Secure in the PAC:
  - Secure access is granted
  - Non-Secure access is discarded (Write is ignored, read 0x0), a PAC error is triggered

Important: These rules do not apply to the specific peripherals called Mix-Secure peripherals.

Note: The Secure application will usually provide an API for the Non-Secure application using the Non-Secure Callable region (NSC) to allow the Non-Secure application to request specific resources.

Table 12-2. Peripheral PAC Security Attribution (Excluding Mix-Secure Peripherals)

Mode	Secure Host Access	Non-Secure Host Access
Non-Secure	Read/Write	Read/Write
Secure	Read/Write	Discarded (Write ignored/Read 0x0) PAC Error is generated

Important: Refer to "*Memory and Peripheral Security Configurations*" section for more details on the programming of NONSECA, NONSECB, and NONSECC parameters.

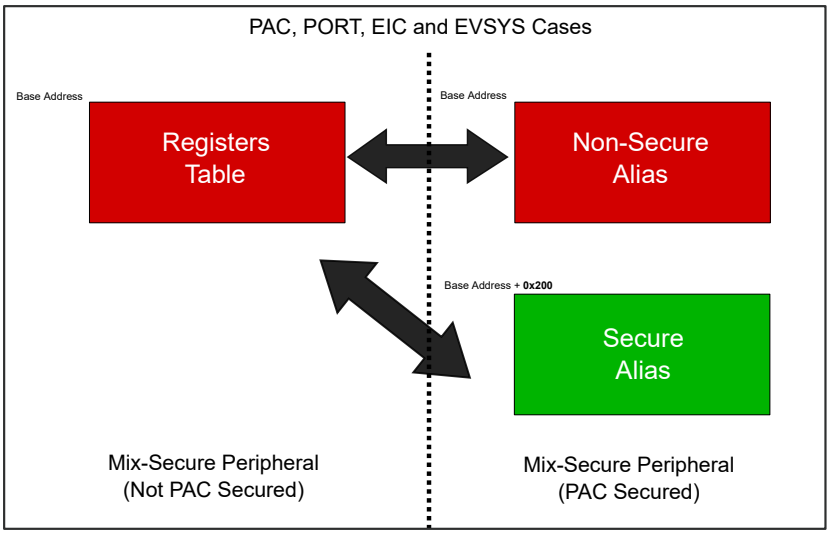
12.2.2.1 Mix-Secure Peripherals

There are five Mix-Secure peripherals that allow internal resources to be shared between the Secure and Non-Secure applications:

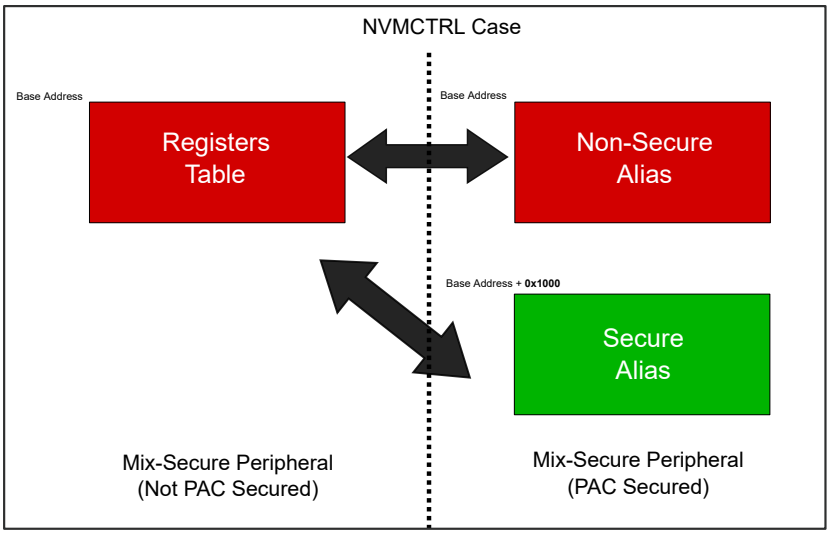
- The PAC controller which manages peripherals security attribution (Secure or Non-Secure).
- The Flash memory controller (NVMCTRL) which supports Secure and Non-Secure Flash regions programming.
- The I/O controller (PORT) which allows to individually allocate each I/O pin to the Secure or Non-Secure applications.
- The External Interrupt Controller (EIC) which allows to individually assign each external interrupt to the Secure or Non-Secure applications.
- The Event System (EVSYS) allows to individually assign each event channel to the Secure or Non-Secure applications.

When a Mix-Secure peripheral is configured as Secure in the PAC, its register map is automatically duplicated in a Secure and Non-Secure alias:

- The Non-Secure alias is at the peripheral base address.
- The Secure alias is located at the peripheral base address:
  - + 0x200 offset for the PAC, EIC, PORT and EVSYS peripherals



  - + 0x1000 offset for the NVMCTRL peripheral.



The Secure alias has the following characteristics:

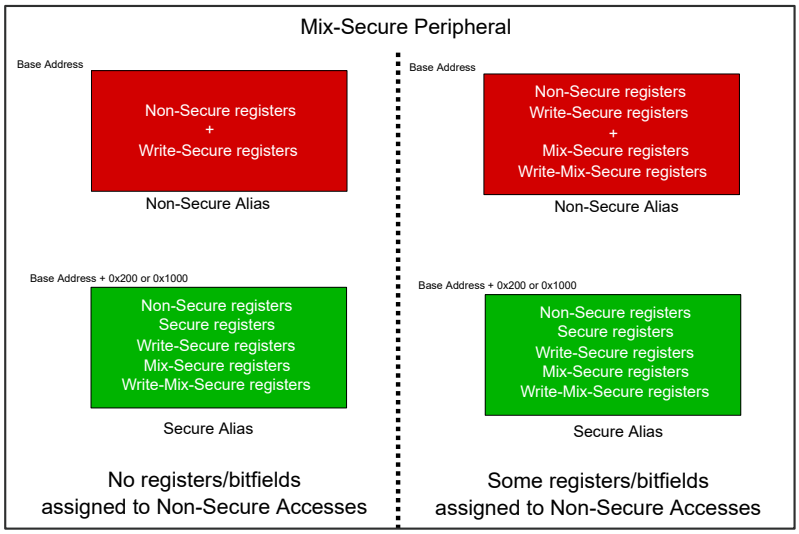
- All of the peripheral registers are available for the Secure application through the Secure alias

- When an internal resource becomes available to the Non-Secure application, the corresponding registers (called Mix-Secure registers) or bitfields in registers are still accessible through this Secure alias by the Secure application
- Non-Secure accesses to this Secure alias are discarded (writing is ignored, reading returns 0x0) and a PAC error is triggered

The Non-Secure alias has the following characteristics:

- Only a restricted set of registers are available for the Non-Secure application through the Non-Secure alias
- It is the responsibility of the Secure application to assign some resources to the Non-Secure application. This is done by setting the corresponding bits in the NONSECx registers of the Mix-Secure peripheral.
  - When an internal resource becomes available for the Non-Secure application, the corresponding registers (called Mix-Secure and Write-Mix-Secure registers) or bitfields in the registers are accessible through the Non-Secure alias by the Non-Secure application
  - Non-Secure accesses to Secure resources (registers, bitfields) are silently discarded (writing is ignored, reading returns 0x0) and no error is generated
- Secure accesses to the Non-Secure alias are silently discarded (writing is ignored, reading returns 0x0) and no error is generated

Figure 12-7. Register Types of Mix-Secure Peripherals



Mix-Secure peripherals have always the following registers:

- NONSEC register is a generic register that tells the Non-Secure application which resources inside a Mix-Secure peripheral can be used
- NSCHK register is a register allowing the Non-Secure application to be notified when the security configuration of a Mix-Secure peripheral is being modified during application execution

Important: It is recommended that the Non-Secure application first copy the content of the NONSEC register inside the NSCHK register, and then enable the NSCHK interrupt flags. Once done, any changes to the NONSEC register by the Secure application will trigger an interrupt so that Non-Secure application can take appropriate actions. This mechanism allows the Secure application to dynamically change the security attribution of a Mix-Secure peripheral and avoid illegal accesses from the Non-Secure application. The interrupt handler must always copy the NONSEC register to the NSCHK register before exiting it.

Mix-Secure peripherals can have the following five registers:

- **Non-Secure**: These registers will always be available in both the Secure and Non-Secure aliases
- **Secure**: These registers will never be available in the Non-Secure alias and always available in the Secure alias
- **Write-Secure**: These are registers than can:

  - Be written or read by the Secure application only in the Secure alias
  - Only read by the Non-Secure application in Non-Secure alias. Write is forbidden.
- **Mix-Secure** registers : These ones are used when a resource can be allocated to either the Secure and Non-Secure alias
  - In some cases, the Mix-Secure properties apply to a bitfield only (like one I/O bit in the PORT peripheral register)
- **Write-Mix-Secure** registers (NVMCTRL peripheral only): These are Mix-Secure registers, which:
  - Can be written or read by the Secure application only in the Secure alias
  - Can only be read by the Non-Secure application in Non-Secure alias except if Non-Secure writes are authorized in NVMCTRL.NONSEC register

Table 12-3. PIC32CM LS00/LS60 Mix-Secure Peripheral Registers Access

Mix-Secure Peripheral Register	Secure Host Access		Non-Secure Host Access	
	Secure Alias	Non-Secure Alias	Secure Alias	Non-Secure Alias
Non-Secure	Read / Write	Discarded (Write ignored / Read 0x0) No Error is generated	Discarded (Write ignored / Read 0x0) PAC Error is generated	Read /Write
Secure				Discarded (Write ignored / Read 0x0) No Error is generated
Write-Secure				Read-only (Write ignored) No Error is generated
Mix-Secure				Read/Write if the resource is available for the Non-Secure Application Discarded if not (Write ignored / Read 0x0) and no error is generated
Write-Mix-Secure				Read /Write if the resource is available for the Non-Secure Application Read-only if not (Write ignored) and no error is generated

12.2.3 Memory and Peripheral Security Configurations

Memory and Peripheral security configurations are stored in different NVM configuration bit fields on two NVM Configuration rows:

- The User Row (UROW)
- The Boot Configuration Row (BOCOR)

Memory and Peripheral security configurations are read from these NVM Configuration rows after each reset during Boot ROM execution and are loaded after Boot ROM verifications into their respective peripheral registers (IDAU, PAC, NVMCTRL and DSU).

Peripheral	Peripheral Register(s)	Security Parameters	Security Configurations
IDAU	SECCTRL, SCFGB, SCFGA, SCFGR	AS, ANSC, DS, RS, RXN BNSC, BOOTPROT	Memories security configurations (Flash, Data Flash and SRAM)
PAC	NONSECA, NONSECB, NONSECC SECLOCKA, SECLOCKB, SECLOCKC	One bit per Peripheral	Peripherals security configuration Peripherals security configuration lock
DSU	STATUSB	DAL	Debug Access Level status
NVMCTRL	SECCTRL, SCFGB, SCFGAD	DXN URWEN, BCREN, BCWEN	Memories security configurations (Data Flash) NVM Configuration rows R/W capability
	SECCTRL	DALUN	Debug Access Level capability

Important: Modifying the security configurations by (re)-programming the different NVM Configuration rows (using the NVMCTRL peripheral) is possible but the changes done on these NVM Configuration rows will always be applied only after a new reset sequence happens (through a new Boot ROM execution).

It is also possible, depending on the SECCFGLOCK NVM bit from BOCOR row, to allow the modifications of the security configurations during the application execution by programming the different IDAU, PAC, and NVMCTRL peripheral registers. This brings an added-value in term of security privileges to the secure software code running out of the Flash Boot region compared to the one running out of the Flash APPLICATION region as it is possible to exit the Boot ROM without locking the security configuration bits.

Therefore, the secure software code of the Flash Boot region will have the responsibility to lock the security configuration before passing control on to the secure software code of the Flash application region.

.

After exiting the Boot ROM:

- If SECCFGLOCK = 1:
  - The security configurations are locked, hence no code (even secure) can change them before next reset sequence.
  - The only way to update the security configurations is to reprogram the NVM Configuration rows then reset the device.
- If SECCFGLOCK = 0:
  - The security configurations can be modified during the application execution.
  - It remains also possible to update the security configurations by reprogramming the NVM Configuration rows then resetting the device.

Note: Refer to the IDAU, NVMCTRL and PAC peripherals for additional information.

⚠ CAUTION

If BOCOR.SECCFGLOCK = 0, to guarantee the security of the overall application, it is critical that the secure software code of the Boot region locks all the IDAU/PAC/NVMCTRL security configuration registers and restore the Debug Access Level configuration:

- IDAU.SECCTRL.SCFGWEN = 0
- NVMCTRL.SECCTRL.SCFGWEN = 0
- PAC.WRCTRL = SECLOCK command for each peripheral (excluding IDAU/DSU which are always locked)
- NVMCTRL.SECCTRL.DALUN = 1 which restores DAL configuration

Refer to the IDAU, NVMCTRL, and PAC peripherals for additional information.

12.3 Crypto Acceleration

12.3.1 Overview

The PIC32CM LS00/LS60 products embed a hardware/software cryptographic accelerator (CRYA) which supports Advanced Encryption Standard (AES) encryption and decryption, Secure Hash Algorithm 2 (SHA-256) authentication, and Galois Counter Mode (GCM) encryption and authentication through a set of APIs.

Note: The CRYA cryptographic accelerator is mapped as a client on the IOBUS port and is driven by the CPU using assembly code (located in ROM).

The Advanced Encryption Standard (AES) is compliant with the American FIPS (Federal Information Processing Standard) Publication 197 specification. The AES operates on a 128-bit block of input data. The key size used for an AES cipher specifies the number of repetitions of transformation rounds that convert the input plaintext, into the final output, called the ciphertext. The AES works on a symmetric-key algorithm, meaning the same key is used for both encrypting and decrypting the data.

The SHA-256 is a cryptographic hash function that creates a 256-bit hash of a data block. The data block is processed in chunks of 512 bits.

The GCM is a mode of operation for AES that combines the CTR (Counter) mode of operation with an authentication hash function.

For detailed algorithm specification, refer to the following standards and specification:

- AES: FIPS Publication 197, Advanced Encryption Standard (AES)
- SHA: FIPS Pub 180-4, The Secure Hash Standard
- GCM: NIST Special Publication 800-38D Recommendation

12.3.2 Features

- Advanced Encryption Standard (AES), compliant with FIPS Publication 197
  - Encryption with 128-bit, 192-bit and 256-bit cryptographic key
  - Decryption with 128-bit, 192-bit and 256-bit cryptographic key
- Secure Hash Algorithm 2 (SHA-256), compliant with FIPS Pub 180-4
  - Accelerates message schedule and inner compression loop
- Galois Counter Mode (GCM) encryption using AES engine and authentication
  - Accelerates the GF(2128) multiplication for AES-GCM hash function

12.3.3 CRYA APIs

The CRYA APIs which are located in a dedicated Boot ROM area. This area is an execute-only area, meaning the CPU cannot do any loads, but can call the APIs. The Boot ROM memory space is a secure area, only the secure application can directly call these APIs.

Table 12-4. CRYA APIs Addresses

CRYA API	Address
AES Encryption	0x02006804
AES Decryption	0x02006808
SHA256 Init	0x02006810
SHA256 Update	0x02006814
SHA256 Final	0x02006818
SHA256 Process (**legacy API**)	0x02006800
GCM Process	0x0200680C

Important: All 8-bit pointers from CRYA API functions must be 32-bit aligned.

12.3.3.1 AES API

The AES software has two function routines to do encryption and decryption on a 128 bit block of input data.

```
/* Definitions for CRYA AES Key Length. */
#define AES_KEY_128_LEN          4
#define AES_KEY_192_LEN          6
#define AES_KEY_256_LEN          8
```

```
/* Type definitions for CRYA AES functions. */
typedef void (*crya_aes_encrypt_t) (const uint8_t *keys, uint32_t key_len, const uint8_t
*src, uint8_t *dst);
typedef void (*crya_aes_decrypt_t) (const uint8_t *keys, uint32_t key_len, const uint8_t
*src, uint8_t *dst);
```

The AES encryption function entry point is located at the Boot ROM address 0x02006804:

```
/* AES encrypt function
* \param keys[in]: A pointer to a 128-bit, 192-bit or 256-bit key
* \param key_len[in]: Number of 32-bit words comprising the key, 4/6/8 for 128-/192-/256-bit
key
* \param src[in]: A pointer to a 128-bit data block to be encrypted
* \param dst[out]: A pointer to a 128-bit encrypted data
*/
#define secure_crya_aes_encrypt ((crya_aes_encrypt_t ) (0x02006804 | 0x1))
```

The AES decryption function entry point is located at the Boot ROM address 0x02006808:

```
/* AES decrypt function
* \param keys[in]: A pointer to a 128-bit, 192-bit or 256-bit key
* \param key_len[in]: Number of 32-bit words comprising the key, 4/6/8 for 128-/192-/256-bit
key
* \param src[in]: A pointer to a 128-bit data block to be decrypted
* \param dst[out]: A pointer to a 128-bit decrypted data
*/
#define secure_crya_aes_decrypt ((crya_aes_decrypt_t ) (0x02006808 | 0x1))
```

12.3.3.2 SHA-256 API

The SHA-256 API can be used to hash variable lengths of input data.

The API is composed of three different functions which must be called in a specific order:

1. SHA-256 Init to initiate a SHA256_CTX structure
2. SHA-256 Update to add a message to be computed in the digest
3. SHA-256 Final to compute the digest

Note: SHA-256 Update can be called several times in the case several messages are to be included in the digest computation.

The SHA-256 structure to define is called SHA56_CTX:

```
/*
*Context structure used by the crya_sha256_init, crya_sha256_update
andcrya_sha256_final functions.
*/
typedef struct {
    /* Digest result of SHA256 */
    uint32_t digest[8];
    /* Length of the message */
    uint64_t length;
    /* Holds the size of the remaining part of data */
    uint32_t remain_size;
    /* Buffer of remaining part of data (512 bits data block) */
    uint8_t remain_ram[64];
    /* RAM buffer of 256 bytes used by crya_sha_process */
    uint32_t process_buf[64];
} SHA256_CTX;
```

The SHA-256 Init function entry point is located at the Boot ROM address 0x02006810:

```
/**
*\brief Type definition for SHA256 initialization function.
*
*\param context[In/Out]: A pointer to a SHA256_CTX structure.
*/
typedef void (*crya_sha256_init_t) (SHA256_CTX *context);
#define crya_sha_init ((crya_sha256_init_t) (0x02006810 | 0x1))
```

The SHA-256 Update function entry point is located at the Boot ROM address 0x02006814:

```
/**
*\brief Type definition for SHA256 update function.
*
*\param context[In/Out]: A pointer to a SHA256_CTX structure.
*\param data[In]:A pointer to a byte-aligned message
*\param length[In]:Size in bytes of the message
*/
typedef void (*crya_sha256_update_t) (SHA256_CTX *context, const unsigned char
*data, size_t length);
#define crya_sha_update ((crya_sha256_update_t) (0x02006814 | 0x1))
```

The SHA-256 Final function entry point is located at the Boot ROM address 0x02006818:

```
/**
*\brief Type definition for SHA256 final function.
*
*\param context[In/Out]: A pointer to a SHA256_CTX structure.
*\param output[Out]:A pointer to the final SHA256 result.
*/
typedef void (*crya_sha256_final_t) (SHA256_CTX *context, unsigned char output[32]);
#define crya_sha256_final ((crya_sha256_final_t) (0x02006818 | 0x1))
```

Code example of using CRYA SHA-256 API:

```
void sha256_hash(SHA256_CTX *ctx, const uint8_t *message, uint32_t length, unsigned char
digest[32])
{
    crya_sha256_init(ctx);
    crya_sha256_update(ctx, message, length);
    crya_sha256_final(ctx, digest);
}
```

Note: crya_sha_update can be called several times in the case several messages are to be included in the digest computation.

12.3.3.2.1 SHA-256 Legacy API

The SHA software function can update the digest value based on the 512-bit data.

It is assumed that the message is already preprocessed properly for the SHA algorithm, so that the SHA software can work directly on 512-bit portions.

The SHA function entry point is located at the Boot ROM address 0x02006800 and has three parameters:

- [In/out]: A pointer to a digest location (digest input and output)
- [In]: A pointer to a 512-bit data block
- [In]: A pointer to a RAM buffer (256B needed for internal algorithm)

The updated digest value is put at first parameter after the function exit.

The API is:

```
/* Type definition for CRYA SHA function. */
typedef void (*crya_sha_process_t) (uint32_t digest_in_out[8], const uint8_t
data[64], uint32_t ram_buf[64]);
```

```
/* CRYA SHA function
* \param digest_in_out[In/out]: A pointer to a digest location (digest input and
output)
* \param data[In]: A pointer to a 512 bit data block
* \param ram_buf[In]: A pointer to a RAM buffer (256B needed for internal algorithm)
*/
#define crya_sha_process ((crya_sha_process_t ) (0x02006800 | 0x1))
```

Code example of using CRYA SHA software:

```
void sha256_process(uint32_t digest[8], const uint8_t data[64])
{
   uint32_t ram_buf[64]; /* 256 bytes needed for message schedule table */

   /* Pointer to CRYA SHA function in ROM */
   static void (*crya_sha_process)(uint32_t digest_in_out[8], const uint8_t data[64],
               uint32_t ram_buf[64]);

   crya_sha_process = (void (*)(uint32_t *, const uint8_t *, uint32_t *))
                      *((uint32_t*)0x02006800);

   crya_sha_process (digest, data, ram_buf);
}
```

12.3.3.3 GCM API

The GCM function entry point is located at the Boot ROM address 0x0200680C:

```
/* Type definition for GF(2^128) multiplication */
typedef void (*crya_gf_mult128_t) (const uint32_t *block1, const uint32_t *block2,
uint32_t *dst);
```

```
/* GF(2^128) multiplication.
*
```

```
* \param block1[In]: A pointer to 128-bit data blocks that are to be multiplied
* \param block2[In]: A pointer to 128-bit data blocks that are to be multiplied
* \param dst[out]: A pointer to a location for storing the result
*/
#define secure_crya_gf_mult128 ((crya_gf_mult128_t ) (0x0200680C | 0x1))
```

12.4 Secure Boot

SHA-256-, HMAC- (PIC32CM LS00/LS60) and ATECC608B-based (PIC32CM LS60) secure boot are supported.

When enabled, secure boot authenticates the user boot loader image in a configurable portion on the Flash (BOOT region) allowing to reset and restart the authentication process in case of a failure.

Refer to 13. Boot ROM for more information.

12.5 Data Flash Scrambling

When Data Flash scrambling is enabled, address and data in the secure portion of the Data Flash are scrambled when written, and de-scrambled when read.

Refer to 29. Non-volatile Memory Controller (NVMCTRL) for more details.

12.6 Secure Pin Multiplexing on SERCOM1 (PIC32CM LS00 only)

The Secure Pin Multiplexing feature can be used on dedicated SERCOM1 I/O pins to isolate a secured communication with external devices from the non-secure application.

This feature is automatically enabled as soon as the security attribution of the SERCOM1 is set to Secured using the PAC peripheral.

When this operation occurs:

- The secured SERCOM1 instances become mapped only on a specific set of I/Os
- All of the alternate SERCOM1 I/O pins of the secured SERCOM1 instance are kept in a Hi-Z configuration
- The PTC cannot enable PTC lines mapped to any of the secured SERCOM1 instance I/O pins
- The CCL I/Os mapped to the secured SERCOM1 instance I/O pins are set to '0'

Refer to the 33. Serial Communication Interface (SERCOM) chapter to obtain the list of pins supporting that feature.

12.7 ATECC608B CryptoAuthentication Device (PIC32CM LS60 only)

The PIC32CM LS60 embeds a pre-provisioned variant of the ATECC608B, called ATECC608B-TFLXTLS.

The ATECC608B-TFLXTLS configuration of the PIC32CM LS60 family is identical to that of the TrustFLEX ATECC608B-TFLXTLS secure element, with the exception of a few device configurations, documented in this chapter, which are unique to the PIC32CM LS60.

Note: Refer to the *ATECC608B-TFLXTLS CryptoAuthentication™* Data Sheet (DS40002138) for additional information.

12.7.1 Interconnect and Configuration

The ATECC608B is connected to the PIC32CM LS60 Microcontroller using a dedicated SERCOM I²C Host peripheral: SERCOM1.

The SERCOM1 signals are not available for the PIC32CM LS60 family as they are reserved for the ATECC608B interconnection. Therefore, no specific configuration is required on the I/O Pin Controller (PORT) peripheral to enable the SERCOM1 functionality for the ATECC608B.

Apart from the PORT configuration which is not required, the SERCOM1 I²C Host peripheral still needs to be configured like any SERCOM peripherals. Refer to the Chapter "I²C Interface" from the *ATECC608B-TFLXTLS CryptoAuthentication™* Data Sheet for additional information.

Important: The I²C address of the secure element is 0xC0.

The ATECC608B SDA/SCL pull-ups are only enabled when the SERCOM1 is configured and enabled in I²C Host mode.

⚠ CAUTION The SERCOM1 must be enabled immediately to remove the extra current consumption which may happen due to the SDA and SCK pins left floating.

12.7.2 Configuration Zone

The following section supersedes the "Device Configuration Information" section of the Chapter "ATECC608B-TFLXTLS Configuration Zone" from the *ATECC608B-TFLXTLS CryptoAuthentication™* Data Sheet.

Device Configuration Information

- The serial number for each device is unique and stored in bytes [0:3, 8:12].
- The default 7-bit I²C address is **0x60**. The I²C address **cannot** be overwritten using the `UpdateExtra` command.
- The I/O levels are set to a fixed reference level. Therefore, the Host processor can operate at a lower voltage than the ATECC608B device.
- The watchdog timer is set to a maximum timeout of 1.3s.
- The use of an I/O Protection key is enabled with the key stored in Slot 6.
- For the ATECC608B, the following individual slots may be uniquely configured to be slot lockable or not: slots 2-6, 8, 10-12, 13 and 15.
- Stored Secure Boot (FullDig) mode of operation is enabled for the ATECC608B.
- Random Nonce is required: Digest for the SecureBoot command is encrypted and the nonce used for the digest encryption uses the ATECC608B random number generator.
- Monotonic counters are available for use by the system and are not attached to any keys.
- The Health Test Failure bit is cleared after any time that a command fails as a result of a health test failure. If the failure symptom was transient, the command may pass when run a second time.

12.7.3 Secure Boot

The PIC32CM LS60 Boot ROM provides support for secure boot using the ATECC608B. The general approach is that the Boot ROM will use the ATECC608B to assist in authenticating and checking the integrity of an application code (usually a boot loader) that is to be subsequently executed. Refer to PIC32CM LS00/LS60 Boot ROM for additional information.

13. Boot ROM

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

The Boot ROM features a Boot Interactive mode, which allows the user to perform several actions on the device, such as NVM areas integrity check and chip erase via a debugger connection.

Unless a debugger is connected and places the Boot ROM in Boot Interactive mode, the CPU will jump to the Flash memory, loading the Program Counter (PC) and Stack Pointer (SP) values, and will start fetching Flash user code.

Important: Before jumping to the Flash, the Boot ROM resets the first 4 KB of SRAM and the first 60 bytes of the TrustRAM. The Clocks remain unchanged.

In addition, the PIC32CM LS00/LS60 Boot ROM has extra security features, such as device integrity checks, memories/peripherals security attributions, and secure boot, which can be executed before jumping to the Flash in Secure state.

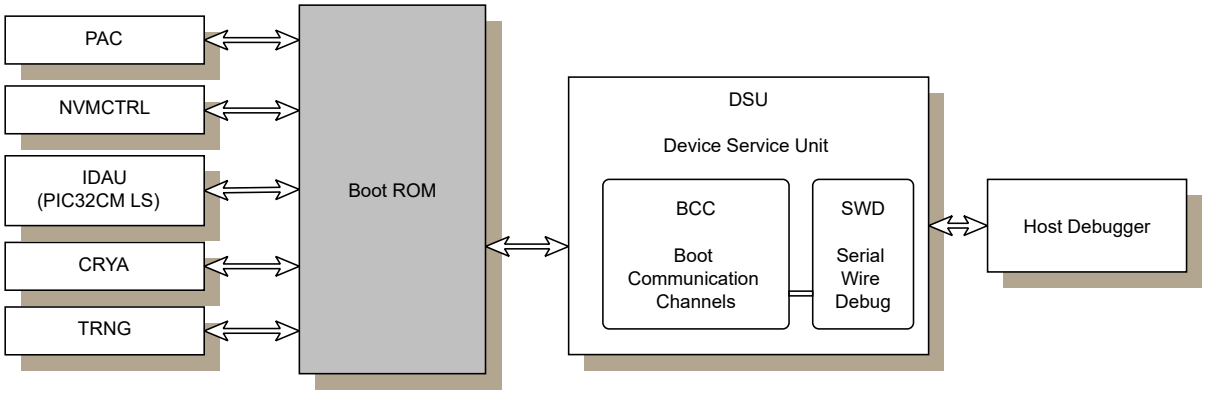
For security reasons, while the Boot ROM is executing, no debug is possible except when entering a specific Boot ROM mode called CPU Park mode.

13.1 Features

- Command interface for the host debugger supporting:
  - Chip erase commands to provide secure transitions between the different Debug Access Levels (DAL)
  - Device integrity check of the NVM memory regions
  - Debugger read access of the NVM Configuration rows
- CPU Park mode to get access for a debugger to the resources of the device depending on Debug Access Level (DAL)
- PIC32CM LS00/LS60 Added features:
  - Device integrity checks
  - Memory and peripheral security attributions from user configuration stored in NVM Configuration rows
  - SHA- or HMAC-based Secure Boot on Secure Flash (BOOT region) and Non-Secure Callable Flash (BOOT region)
  - Device Identity Composition Engine (DICE) security standard support with Unique Device Secret (UDS)
- PIC32CM LS60 Added features:
  - ATECC608B-based Secure Boot on Secure Flash (BOOT region) and Non-Secure Callable Flash (BOOT region)

13.2 Block Diagram

Figure 13-1. Boot ROM Block Diagram



13.3 Boot ROM Dependencies

13.3.1 Clocks

The device selects the OSC16M oscillator which is enabled by default after reset and configured at 4 MHz.

13.3.2 NVM User (UROW) and Boot Configuration (BOCOR) Configuration rows

The Boot ROM reads the different NVM Configuration rows during its execution.

The relevant fuses must be set appropriately by any configuration tools supporting the device in order to operate correctly.

Refer to the 10.2. NVM Configuration Rows section for additional information.

13.3.3 Debug Operations

For security reasons, no debug is possible during the Boot ROM execution except when entering the Boot ROM CPU Park mode.

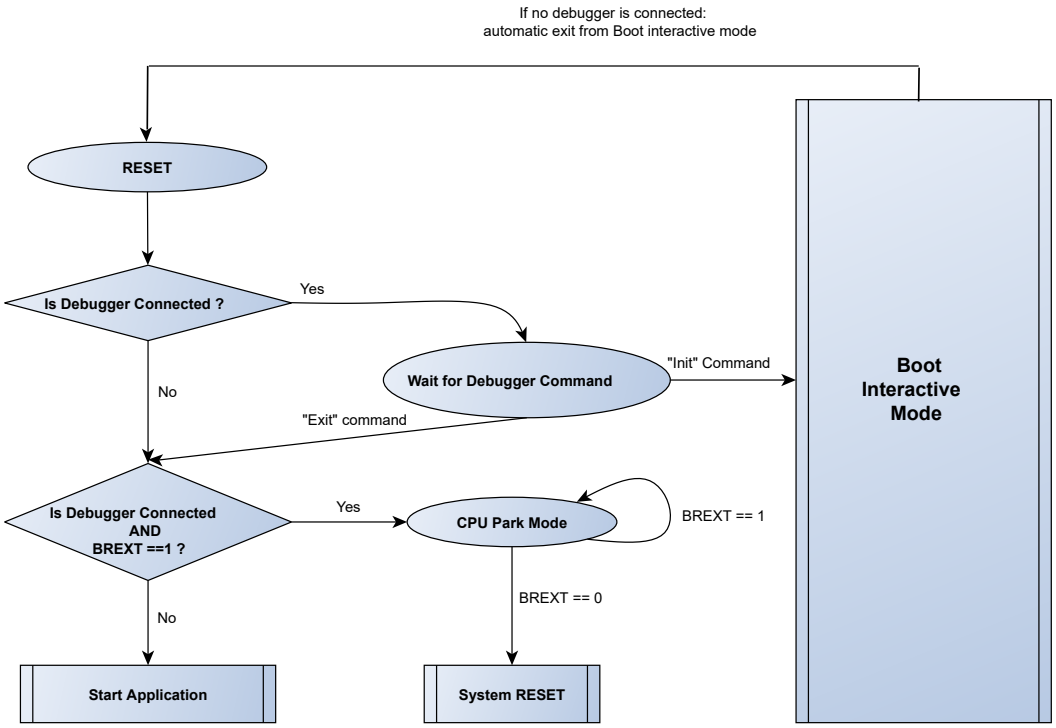
13.4 Functional Description

13.4.1 PIC32CM LE00 Boot ROM

First the PIC32CM LE00 Boot ROM checks if a debugger is present to enter the Boot Interactive mode which allows the user to perform specific tasks through a debugger connection.

Before jumping to the application, the Boot ROM can also enter in a specific mode called CPU Park to allow the debugger to get access to the resources of the device depending on Debug Access Level (DAL).
Note: Boot Interactive and CPU Park modes are described in the later sections of this document..

Figure 13-2. PIC32CM LE00 Boot ROM Flow



13.4.2 PIC32CM LS00/LS60 Boot ROM

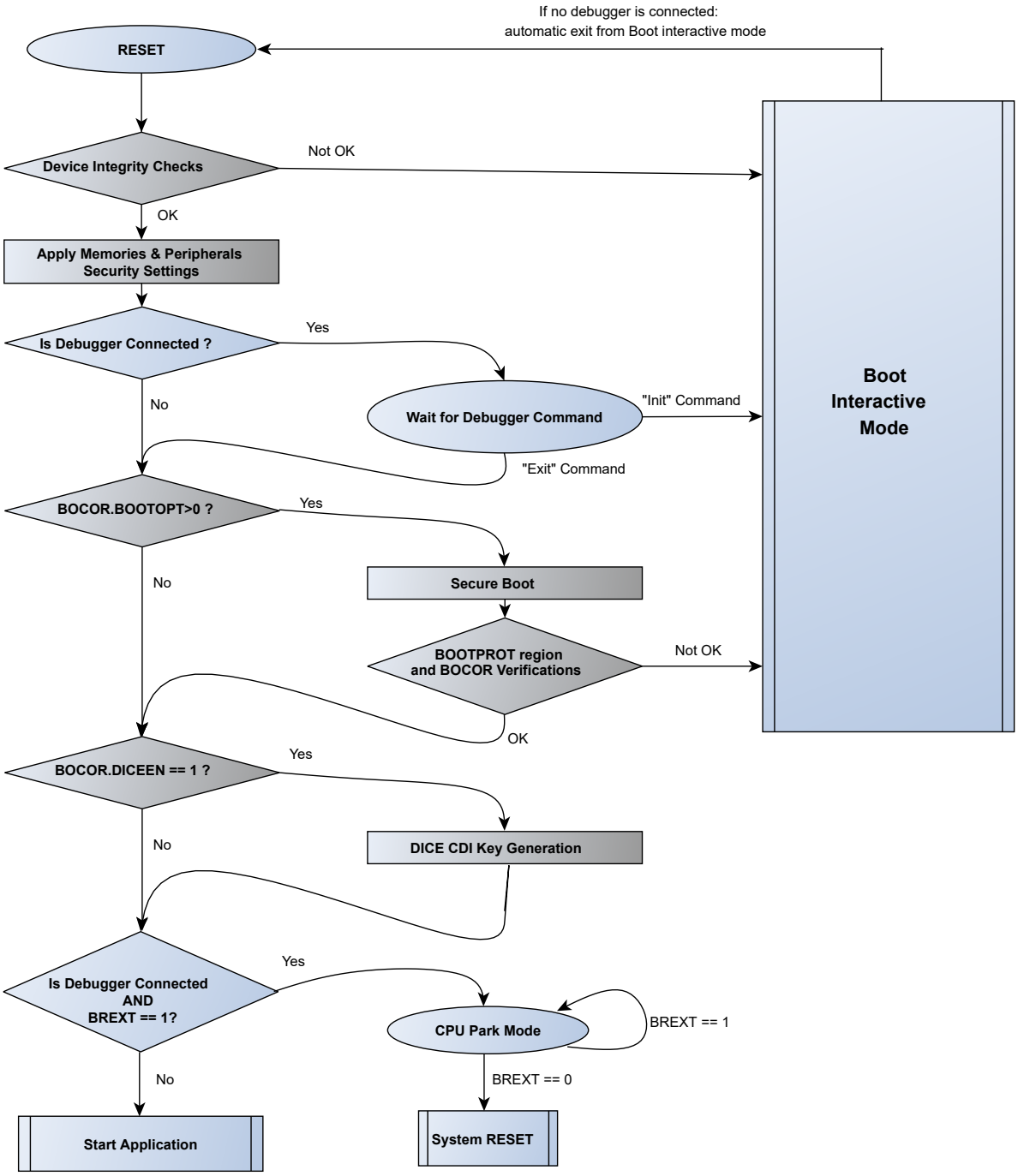
The PIC32CM LS00/LS60 Boot ROM sequence consists in performing several security tasks (integrity checks, memories and peripherals security attribution, Secure Boot, DICE CDI key generation...) before starting the application.

Note: Secure Boot and DICE CDI key generation are independent features and may be enabled simultaneously or without the other.

The Boot ROM first checks if a debugger is present to enter the Boot Interactive mode which allows the user to perform specific tasks via a debugger connection.

Before jumping to the application in Secure state, the Boot ROM can also enter in a specific mode called CPU Park to allow the debugger to get access to the resources of the device depending on Debug Access Level (DAL).
Note: Boot Interactive and CPU Park modes are described later on.

Figure 13-3. PIC32CM LS00/LS60 Boot ROM Flow



13.4.2.1 Device Integrity Checks

For PIC32CM LS00/LS60 devices, the Boot ROM performs security checks on two CRCs:

- The User Row CRC (USERCRC) which is located in the NVM User Row (UROW) at:

UROW Offset	Bit Position	Name
0x20-0x23	287:256	USERCRC

- The Boot Configuration Row CRC (BOCORCRC) which is located in the NVM Boot Configuration Row (BOCOR) at:

BCOR Offset	Bit Position	Name
0x08-0x0B	95:64	BOCORCRC

13.4.2.1.1 User Row CRC (USER CRC)

USERCRC allows to check the following fuses parameters integrity:

- AS, ANSC, DS, RS
- URWEN
- NONSECA, NONSECB, NONSECC
- CDIROFFSET

USERCRC is the CRC of the NVM User row area which starts from 0x00804008 and finish at 0x0080401F (bit 64 to bit 255):

Table 13-1. PIC32CM LS00/LS60 UROW Area Computed in USERCRC

Offset	Bit Pos.	Name							
0x08	71:64	AS							
0x09	79:72	ANSC						AS	
0x0A	87:80	DS				Reserved		ANSC	
0x0B	95:88	Reserved				DS			
0x0C	103:96	RS							
0x0D	111:104	Reserved					URWEN	Reserved	RS
0x0E-0xF	127:112	Reserved							
0x10-0x13	159:128	NONSECA							
0x14-0x17	191:160	NONSECB							
0x18-0x1B	223:192	NONSECC							
0x1C-0x1F	255:224	CDIROFFSET							

13.4.2.1.2 Boot Configuration Row CRC (BOCORCRC)

BOCORCRC allows to check the following fuses parameters integrity:

- BOOTPROT, BNSC, BOOTOPT
- SECCFGLOCK, DICEEN
- BCWEN, BCREN

BOCORCRC is the CRC of the NVM Boot Configuration row area, which starts from 0x0080C000 and finish at 0x00800C007 (bit 0 to bit 63).

Table 13-2. PIC32CM LS00/LS60 BOCOR Area Computed in BOCORCRC

Offset	Bit Pos.	Name							
0x00-0x1	15:0	Reserved							
0x02	23:16	BNSC						Reserved	
0x03	31:24	Reserved					BNSC		
0x04	39:32	BOOTOPT							
0x05	47:40	BOOTPROT							
0x06	55:48	Reserved			DICEEN	SECCFGLOCK	BOOTPROT		
0x07	63:56	Reserved						BCREN	BCWEN

If one of the checks fails, the Boot ROM will report the error to the DSU peripheral and will enter the Boot Interactive mode:

- This will allow, if a debugger is connected, to put the device in the highest debug access level mode (DAL = 2) by issuing a Chip Erase command . Once in that mode, it is possible for a programming tool to reprogram the NVM Configuration Rows.
- When the check fails and no debugger is connected, the part will reset and restart the check sequence again.

Note: Boot Interactive mode is described later in this chapter.

13.4.2.1.3 CRC Computation and Programming

The CRCs needs to be recalculated and updated in their respective NVM Configuration row as soon as a data from any of the checked regions is changed.

Important: USERCRC and BOCORCRC CRCs programming must be done by any programming tool supporting the PIC32CM LS00/LS60 devices.

The algorithm is a CRC-32 module embedded in the DSU peripheral and that uses for both CRC calculation with the following parameters:

- Width = 32 bits
- Polynomial = 0x04C11DB7 (Poly)
- Initial Value = 0xFFFFFFFF (Init)
- Input Data is reflected (RefIn)
- Output Data is reflected (RefOut)
- No XOR is performed on the output CRC (XorOut)

Example: the DSU CRC of 0x31, 0x32, 0x33, 0x34, 0x35, 0x36, 0x37, 0x38, 0x39 is 0x340BC6D9

13.4.2.2 Memories and Peripherals Configurations Initialization

For PIC32CM LS00/LS60 devices, memories and peripherals security attributions are done by reading the different fuses values from the NVM User (UROW) and Boot Configuration (BOCOR) rows.

The Boot ROM is responsible for setting these attributions on the different concerned memory and peripheral controllers:

- Set memory security attribution according to AS, ANSC, DS, RS, BNSC and BOOTPROT fuses
- Set peripherals security attribution according to NONSECA, NONSECB and NONSECC fuses

Important: The Boot ROM does not perform any consistency checks on the configured memory attributions.

13.4.2.3 Secure Boot

Depending on the BOOTOPT fuse value (from BOCOR NVM Configuration row), the following secure boot checks will be performed on:

- The BOOTPROT region which is composed by:
  - The Secure Flash (BOOT region)
  - The Non-Secure Callable Flash (BOOT region)
- The NVM Boot Configuration row (BOCOR)

Different verification methods are supported depending on BOCOR.BOOTOPT value:

Table 13-3. PIC32CM LS00 Secure Boot Verification Methods

BOOTOPT	BOOTPROT region Verification Method	NVM Boot Configuration Row (BOCOR)
0	Secure Boot Disabled	
1	SHA-256	
2	SHA-256 with BOOTKEY(1)	
3	HMAC with BOOTKEY(1)	
Others	Reserved	

Note:

1. BOOTKEY is defined in BOCOR row

Table 13-4. PIC32CM LS60 Secure Boot Verification Methods

BOOTOPT	BOOTPROT region Verification Method	NVM Boot Configuration Row (BOCOR)
0	Secure Boot Disabled	
1	SHA-based Secure Boot	
2	SHA-based Secure Boot with BOOTKEY(1)	
3	HMAC-based Secure Boot with BOOTKEY(1)	
4	ATECC608B-based Secure Boot	SHA
5	ATECC608B-based Secure Boot	SHA with BOOTKEY(1)
6-255	ATECC608B-based Secure Boot	HMAC with BOOTKEY(1)

Note:

1. BOOTKEY is defined in BOCOR row

If the verification fails, the Boot ROM will report the error to the DSU peripheral and will enter the Boot Interactive mode. This will allow, if a debugger is connected, to put the device in the highest debug level access mode (DAL = 2) by issuing a Chip Erase command. Once in that mode, it is possible for a programming tool to reprogram the different memory regions and/or NVM Configuration rows.

When verification fails and no debugger is connected, the part will reset and restart the integrity checks sequences again.

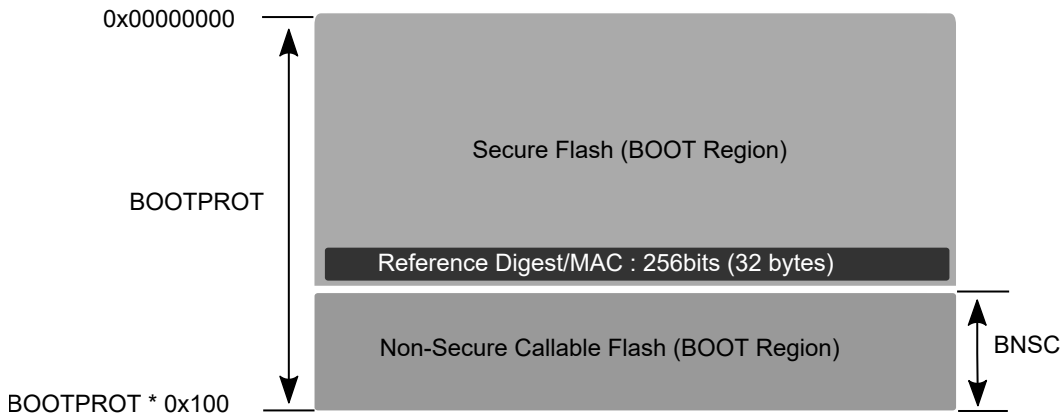
13.4.2.3.1 BOOTPROT Region Verification Using SHA-256 or HMAC

The digest (SHA-256) / MAC (HMAC) is computed on the whole BOOTPROT region, which is composed by the Secure Flash (BOOT region) and the Non-Secure Callable Flash (BOOT region).

The digest/MAC reference value for this area is stored at the end of the Secure Flash (BOOT region), just before the Non-Secure Callable Flash (BOOT region).

Important: The last 256 bits where the digest/MAC is stored are not included in the computation.

Figure 13-4. Reference digest/MAC location



Important: The Flash (APPLICATION region) is not part of the Secure Boot verification. So if an authentication of this memory region is required, it must be handled by the user code itself.

SHA-256 Verification Method
This verification method uses the standard SHA-256 hash algorithm which produces a digest of 256 bits, i.e. 32 bytes that is compared against the value placed at the end of the Secure Flash (BOOT Region).

SHA-256 with BOOTKEY Verification Method
To prevent unauthorized change of the boot loader code, the digest computation can be slightly modified to require a key to produce a valid digest.

When SHA-256 with BOOTKEY is selected, the digest computation (for both BOOTPROT region and NVM BOCOR row) starts by processing the secure boot key (BOOTKEY) data twice, then proceeds with the rest of data.

This secure boot key (BOOTKEY) is located in the NVM Boot Configuration row (BOCOR):

BOCOR Offset	Bit Position	Name
0x50-0x6F	895:640	BOOTKEY

HMAC with BOOTKEY Verification Method
The verification method uses the standard HMAC hash algorithm (defined in FIPS PUB 198-1).

The hash used for HMAC is SHA-256 which produces a MAC of 256 bits, 32 bytes, that is compared against the value placed at the end of the Secure Flash (BOOT Region). The key used is BOOTKEY from BOCOR NVM Configuration row.

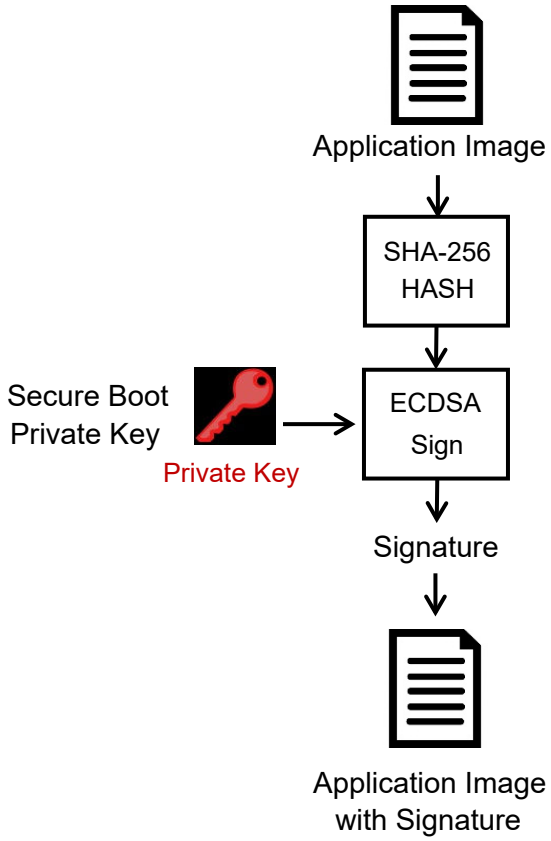
13.4.2.3.2 BOOTPROT Region Verification using the ATECC608B (PIC32CM LS60 only)

When BOOTOPT>0, the Boot ROM uses the ATECC608B to assist in authenticating and checking the integrity of an application code (usually a boot loader) that is to be subsequently executed.

This authentication process is done in two steps:

- An ECDSA signature of the application image must first be generated and stored on the application image
- Secure Boot execution will then authenticate the application code

Figure 13-5. ECDSA Signature Generation

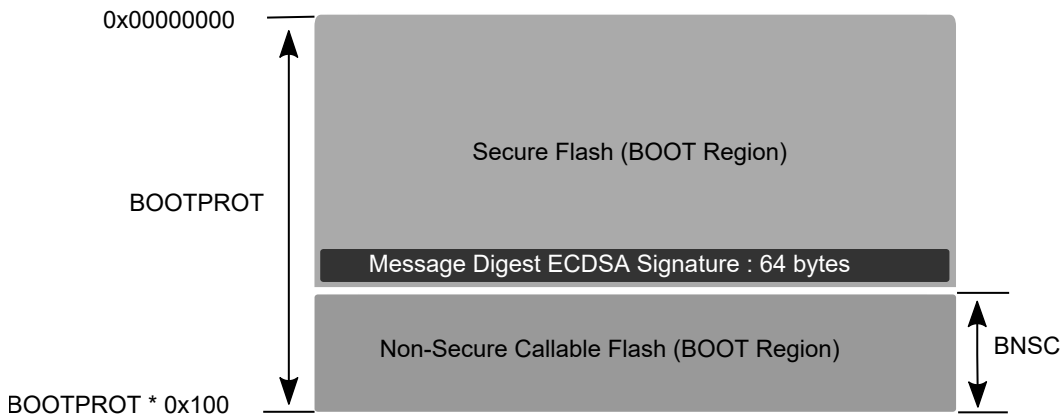


The ECDSA signature (64 bytes) is computed on the overall BOOTPROT region, which is composed by the Secure Flash (BOOT region) and the Non-Secure Callable Flash (BOOT region).

The signature for this area must be stored at the end of the Secure Flash (BOOT region), just before the Non-Secure Callable Flash (BOOT region).

Important: The last 64 bytes where the signature is stored are not included in the computation.

Figure 13-6. ECDSA Signature Location



Important: The Flash (APPLICATION region) is not part of the Secure Boot verification. So, if an authentication of this memory region is required, it must be handled by the user code itself.

ATECC608B-Based Secure Boot Verification Method

This verification method uses the ATECC608B.

The ATECC608B is configured to operate in the Stored Secure Boot (FullDig) mode, where:

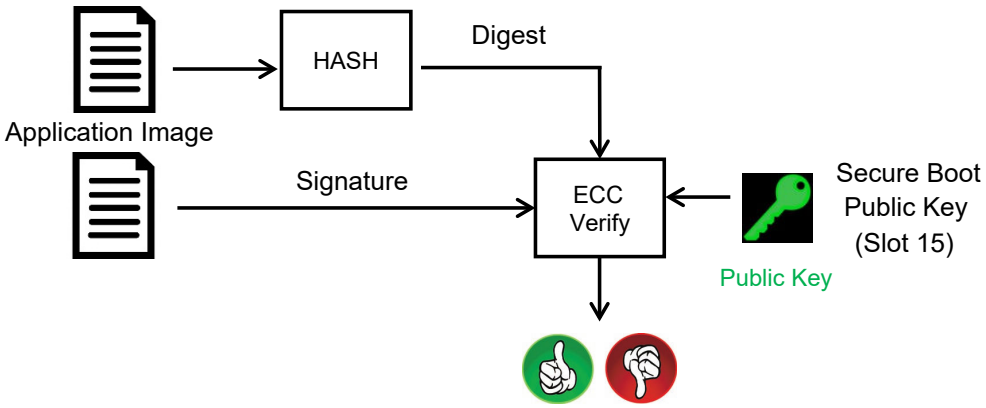
- The digest is stored in Slot 7
- The public key required to verify the SecureBoot is stored in Slot 15
  Note: The public key must be initially provisioned in Slot 15.

The Stored Secure Boot (FullDig) mode executes the ATECC608B SecureBoot command in the following modes:

- FullCopy mode
- FullStore (Digest) mode

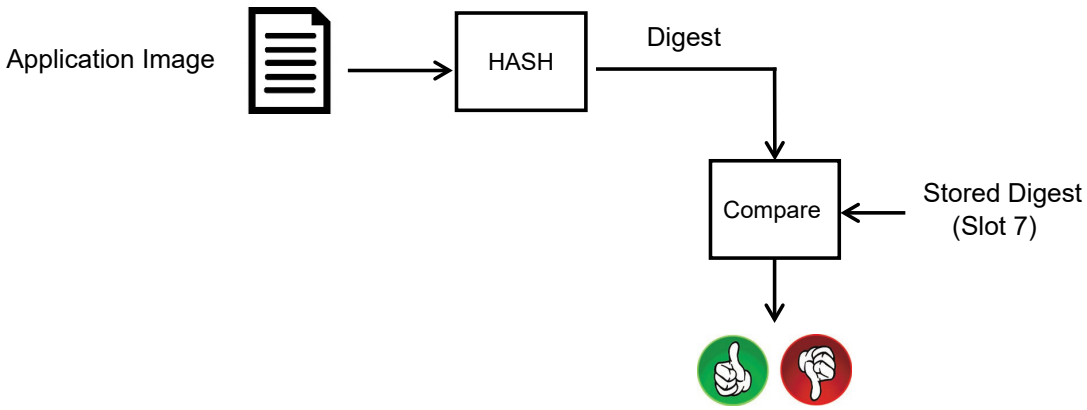
FullCopy mode is executed the first time the code used to authenticate is verified: the Boot ROM passes both the digest and signature to the ATECC608B for verification and stores the digest in the ATECC608B.

Figure 13-7. Secure Boot FullCopy Mode Overview



FullStore (Digest) mode is then executed. In that mode, the Boot ROM passes only the digest to the ATECC608B, which performs a compare with the Digest that has been stored upon the first authentication of the code.

Figure 13-8. Secure Boot FullStore Mode Overview



The communication between the PIC32CM LS60 microcontroller and the ATECC608B is encrypted using an I/O Protection Key. This unique key is initially provisioned in both the ATECC608B Slot 6 and the microcontroller BOCOR NVM Configuration Row (IOPROTKEY).

The ATECC608B Slot 6 is not locked by default, hence if a Full ChipErase is done on the microcontroller (where BOCOR IOPROTKEY will be erased), both the microcontroller and the ATECC608B can be provisioned with a new key value.

⚠ CAUTION If the ATECC608B Slot 6 is locked (i.e. no new I/O Protection Key value can be reprogrammed), it is mandatory for the application to ensure that the microcontroller never executes a Full Chip Erase command or it will break the ATECC608B pairing required for SecureBoot.

Refer to the Chapter “I/O Protection Key” from the *ATECC608B-TFLXTLS CryptoAuthentication*™ Data Sheet to get the ATECC608B commands support this feature.

13.4.2.3.3 NVM Boot Configuration Row (BOCOR) Verification

When BOOTOPT>0, the digest/MAC for the NVM BOCOR row is computed on the whole NVM BOCOR row (excluding BOCORHASH value) and compared with the digest/MAC reference value (BOCORHASH):

BOCOR Offset	Bit Position	Name
0xE0-0xFF	2047:1792	BOCORHASH

13.4.2.4 Device Identity Composition Engine (DICE)

The Device Identifier Composition Engine (DICE) is a standard developed by the Trusted Computing Group (TCG) for implementing attestation in low cost IoT devices.

When enabled using BOCOR.DICEEN fuse, the DICE engine generates the Compound Device Identifier (CDI) at boot time that is based on a stored Unique Device Secret (UDS) key and the digest/MAC of the boot flash image (BOOTPROT region).

BOCOR Offset	Bit Position	Name
0x06	52	DICEEN

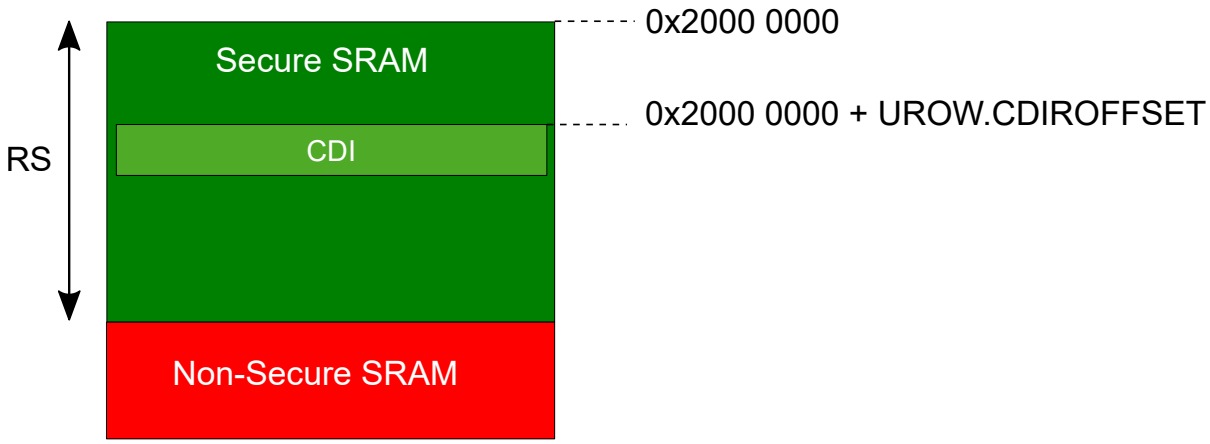
The CDI is written in the SRAM at the offset specified by the UROW.CDIROFFSET fuse, making it available for the boot Flash code for attestation purpose. The boot Flash code can optionally use the CDI to derive other keys for attestation and encryption.

UROW Offset	Bit Position	Name
0x1C-0x1F	255:224	CDIROFFSET

The Boot ROM checks that CDIROFFSET is within the maximum SRAM range; if not, the CDI is not written.

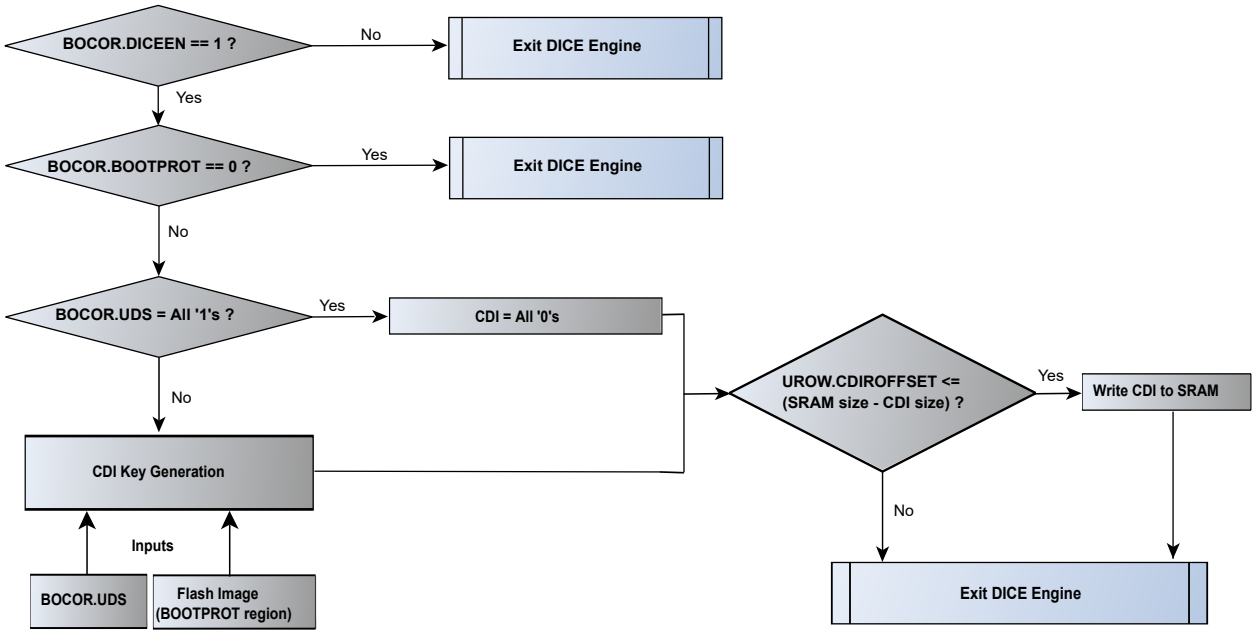
Important: It’s up to the user to ensure the CDI is written in the Secure SRAM region.

Figure 13-9. DICE CDI SRAM location



13.4.2.4.1 DICE CDI Key Generation Flow

Figure 13-10. DICE CDI Key Generation Flow



13.4.2.4.2 Unique Device Secret (UDS)

The Unique Device Secret (UDS) is a 256-bit symmetric key used to generate the CDI. The UDS key is only accessible to the DICE engine and is only used for calculating the CDI at boot time.

Important: The UDS must be accessible only during Boot ROM execution: BOCOR.BCWEN and BOCOR.BCREN must be set both to '0' and BOCOR.SECCFGLOCK must be set to '1' to follow DICE standard.

The Unique Device Secret must be provisioned on BOCOR.UDS fuse. It must be unique for each device and have a strong entropy.

If DICE is enabled but the UDS key is not provisioned (BOCOR.UDS is all 1s), all zeros are written to the CDI output.

Important: Devices can be factory programmed with securely key provisioned software. Contact your local Microchip sales office for more information.

⚠ CAUTION A ChipErase_ALL (CE2) will reset the whole BOCOR including the provisioned UDS.

13.4.2.4.3 Compound Device Identifier (CDI)

The Compound Device Identifier (CDI) is the output of the DICE module that is passed to the boot flash code at a specified memory location in SRAM.

The CDI can be used by the boot flash code directly for attestation or to derive other keys.

The CDI is a 256 bit value calculated as follows:

CDI = HMAC-SHA-256(UDS, H(BOOTPROT region))

Where:

1. The hash function H() calculated over the boot Flash image is SHA-256.
2. The BOOTPROT region is composed by the Secure Flash (BOOT region) and the NSC Flash (BOOT region)

Note: Secure Boot feature requires another digest/MAC

(or signature)

which is stored in the Secure Flash (BOOT region). This digest/MAC is not included in the computation of the BOOTPROT region for CDI generation, to be independent of the Secure Boot operations.

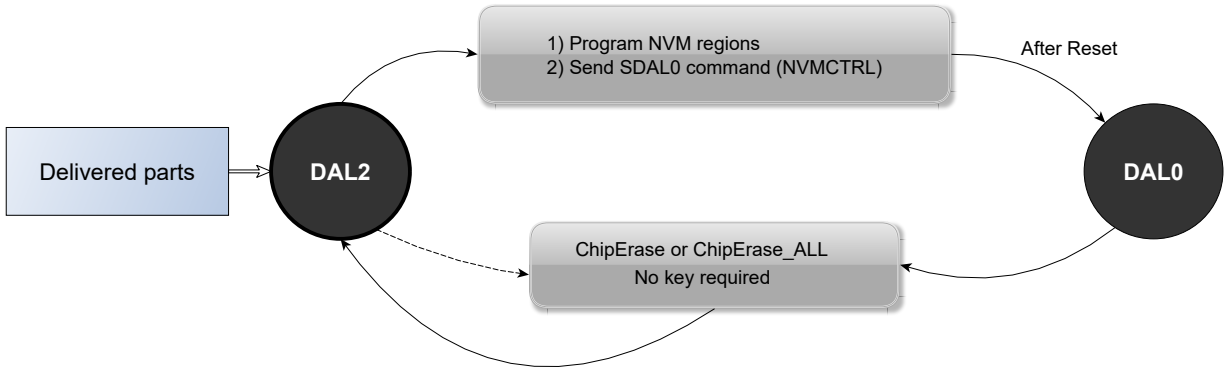
13.4.3 Debug Access Levels

The PIC32CM LE00 has the following two debug access levels (DAL):

- DAL2: Highest debug level access with no restrictions in term of memory and peripheral accesses.
- DAL0: No access is authorized except with a debugger using the Boot ROM Interactive mode.

The possible transitions between each debug access level are described below:

Figure 13-11. PIC32CM LE00 Debug Access Levels Transitions

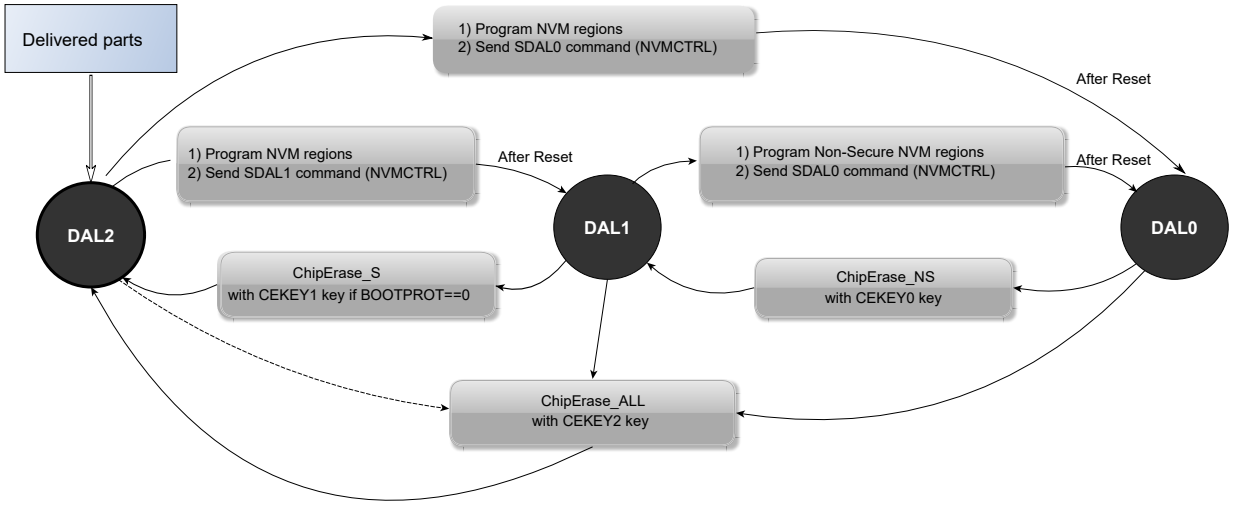


The PIC32CM LS00/LS60 has the following possible debug access levels (DAL):

- DAL2: Highest debug level access with no restrictions in term of memory and peripheral accesses.
- DAL1: Access is limited to the Non-Secure memory regions. Secure memory regions accesses are forbidden.
- DAL0: No access is authorized except with a debugger using the Boot ROM Interactive mode.

The possible transitions between each debug access level are described below:

Figure 13-12. PIC32CM LS00/LS60 Debug Access Levels Transitions



Decreasing the Debug Level Access is done using the NVMCTRL peripheral command from the debugger or the CPU.

Note: Refer to 29. Non-volatile Memory Controller (NVMCTRL) for additional information.

For security reasons, increasing the Debug Level Access is only possible during Boot ROM execution and will be always preceded by a specific chip erase depending on the Debug Access Level.

13.4.4 Chip Erase

The chip erase commands allow to erase memories of the device and provide secure transitions between the different Debug Access Levels.

Important: Chip erase commands are only issued using the Boot ROM Interactive mode (CMD_CE0, CMD_CE1, CMD_CE2 and CMD_CHIPERASE commands).

The Chip erase commands are executed whatever the Flash regions are locked or not. Refer to the Region Unlock Bits in the NVMCTRL chapter.

For PIC32CM LE00, the chip erase command does not require a key.

For PIC32CM LS00/LS60, the chip erase commands are protected with keys (CEKEYx) defined in the NVM BOCOR row.

Note: The chip erase keys can only be read if BOCOR.BCREN = 1.

By default, the devices are delivered with these keys set at “All 1s”.

Important: If a key is set at “All 0s”, the corresponding chip erase command is disabled and it will be impossible for the debugger to use it. If both the ChipErase_ALL (CE2) key is set at "All 0s" and BOCOR.BCWEN = 0, full chip erase is permanently disabled. Depending on Debug Access Levels (DAL0 or DAL1), Microchip’s failure analysis capabilities are limited when this feature is used.

The following table gives the effect of the chip erase commands on the different memories:

Table 13-5. Chip Erase Commands Effects

	PIC32CM LS00/LS60			PIC32CM LE00	
Boot ROM Command	ChipErase_NS (CE0)	ChipErase_S (CE1)	ChipErase_ALL (CE2)	ChipErase (CHIPERASE)	ChipErase_ALL (CHIPERASE_ALL)
Key Requirement	Yes (CEKEY0)	Yes (CEKEY1)	Yes (CEKEY2)	No	
Flash (BOOT region)	No		Yes	No	Yes
Flash (APPLICATION region)	-			Yes	
Data Flash	-			Yes	
Secure Flash (AS region)	No	Yes		-	
Non-Secure Flash (APPLICATION region)	Yes			-	
Secure Data Flash (DS)	No	Yes		-	
Non-Secure Data Flash	Yes			-	
NVM User Row (UROW)	No		Yes	No	Yes
NVM Boot Configuration Row (BOCOR)	No		Yes	No	Yes
Volatile Memories	Yes			Yes	
Debugger Access Level after reset	2 (if DAL was 2) else 1	2 (if DAL was 2 or BOOTPROT==0) else 1	2	2	

Note: Only the ChipErase_ALL (CHIPERASE_ALL or CE2) commands affect rows belonging to the BOOT area (BOOTPROT fuse bits) and the BOCOR row.

13.4.5 Boot ROM Interactive Mode

The interactive mode allows the user to perform several actions on the device during the Boot ROM execution through a debugger connection.

The debugger communicates with the device using the DSU Boot Communication Channels (BCC) through the external address range of the DSU peripheral, regardless of the DAL setting. This communication is bi-directional and allows the debugger to post commands and receive status from the Boot ROM.
Note: Refer to Device Service Unit for additional information on BCC.

13.4.5.1 Enter Interactive Mode (CMD_INIT)

This command allows launching the Boot Interactive command mode of the Boot ROM.

To reach interactive mode, the debugger will trigger a “cold plugging” sequence as described in DSU chapter.

Important: Debugger must not clear the DSU.STATUSA.BREXT bit before clearing the DSU.STATUSA.CRSTEXT bit.

When CRSTEXT is cleared, CPU starts Boot ROM Interactive mode execution. After a small delay (5 ms is advised), the debugger must check if the Boot ROM has not flagged any errors by checking the BCCD1 bit in the DSU.STATUSB register.

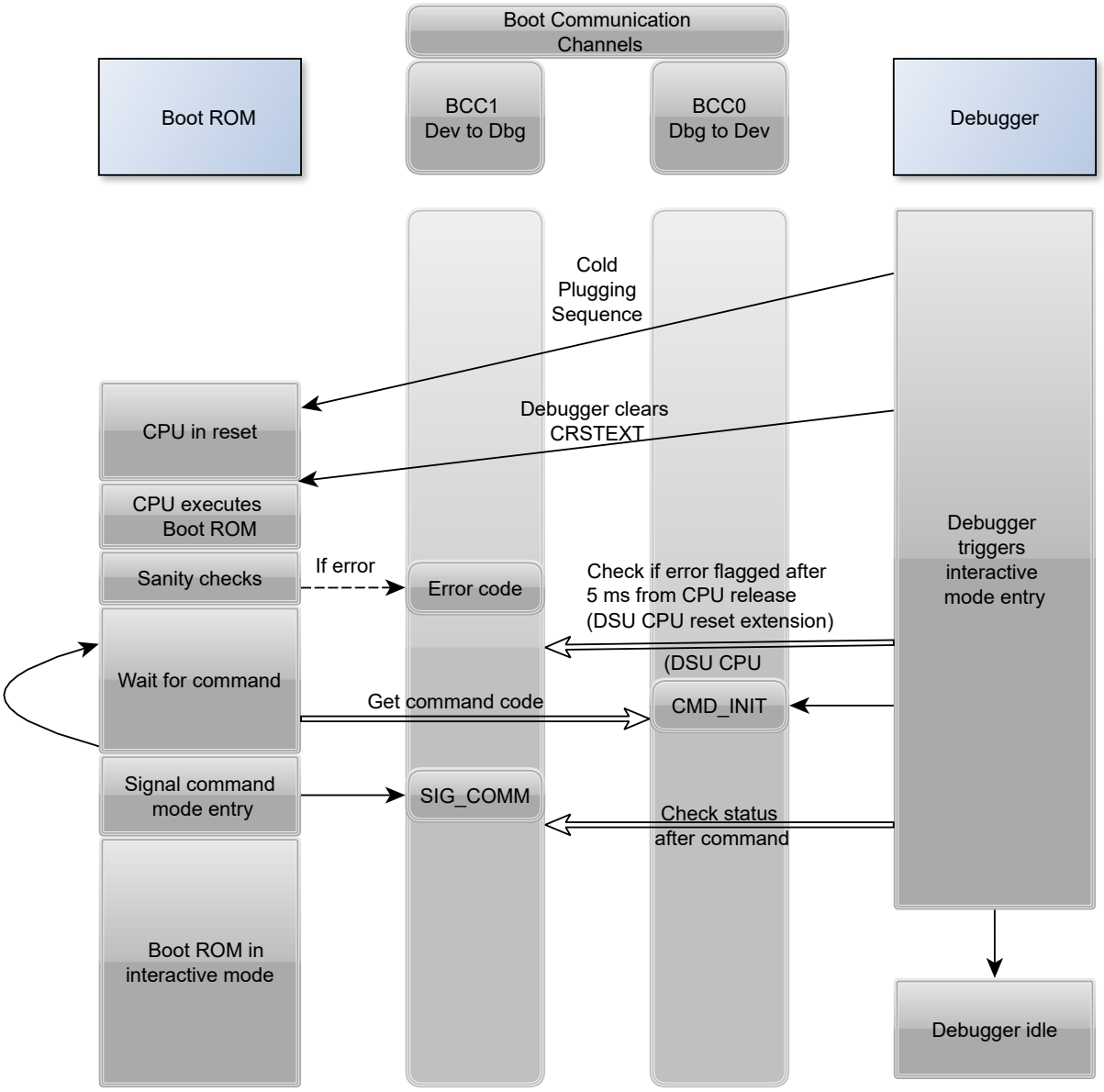
If errors are reported, the debugger can get the error type by checking the DSU.BCC1 register from the DSU external address space.

Note: Errors reported by the Boot ROM in the DSU.BCC1 register are listed later on in the Boot Interactive Mode Status section.

If no error is reported, the debugger writes the CMD_INIT command to the DSU.BCC0 register to request Boot ROM Interactive mode entry. When command is successful, Boot ROM will place the “SIG_COMM” status in the DSU.BCC1 register.

13.4.5.1.1 CMD_INIT

Figure 13-13. CMD_INIT Flow diagram



13.4.5.2 Exit Interactive Mode (CMD_EXIT)

This command allows exiting the Boot Interactive mode.

Exiting the Boot Interactive mode allows to jump to one of the following:

- The application
- The CPU Park mode

13.4.5.2.1 CMD_EXIT

Figure 13-14. CMD_EXIT to APP flow diagram

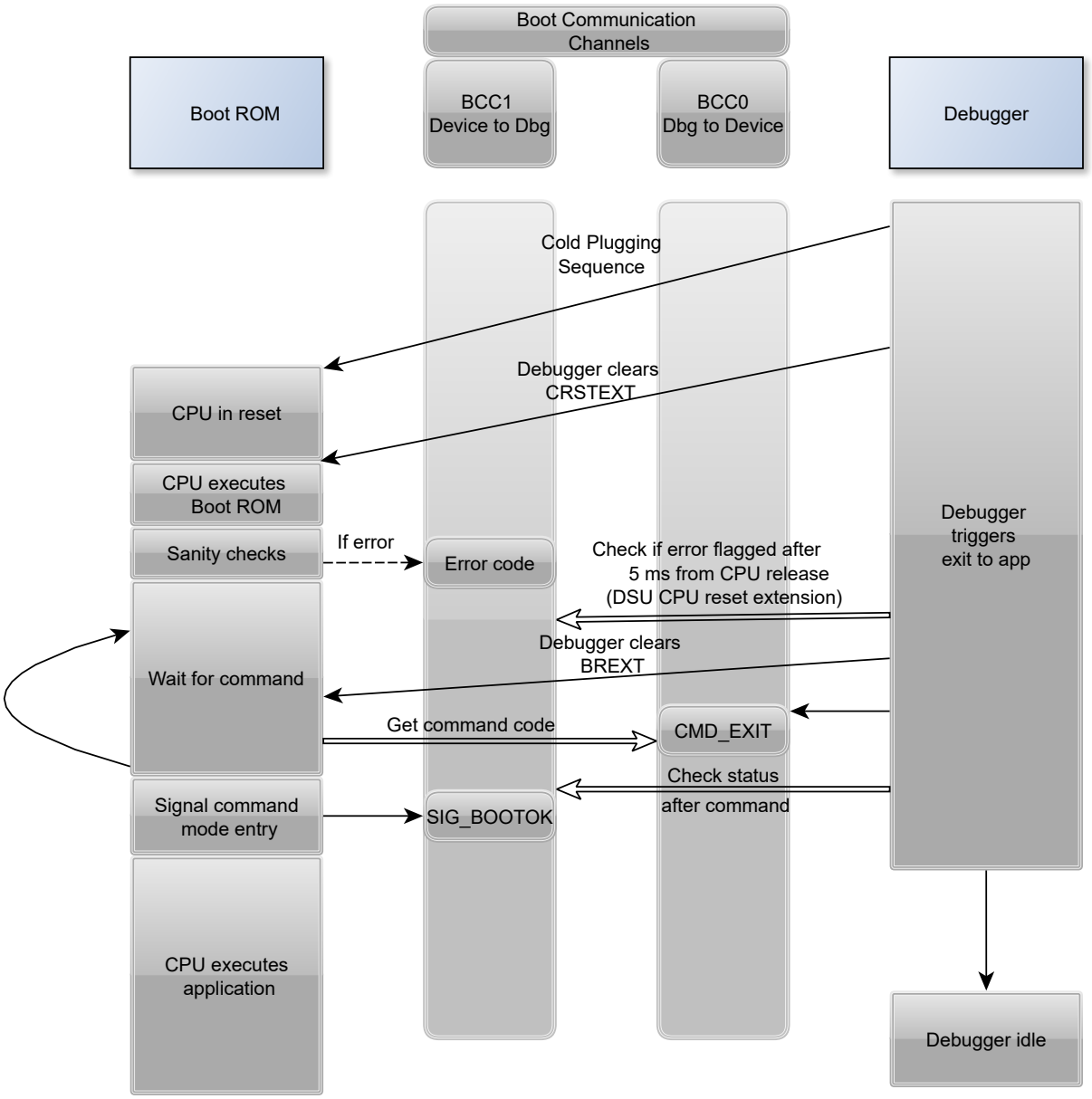
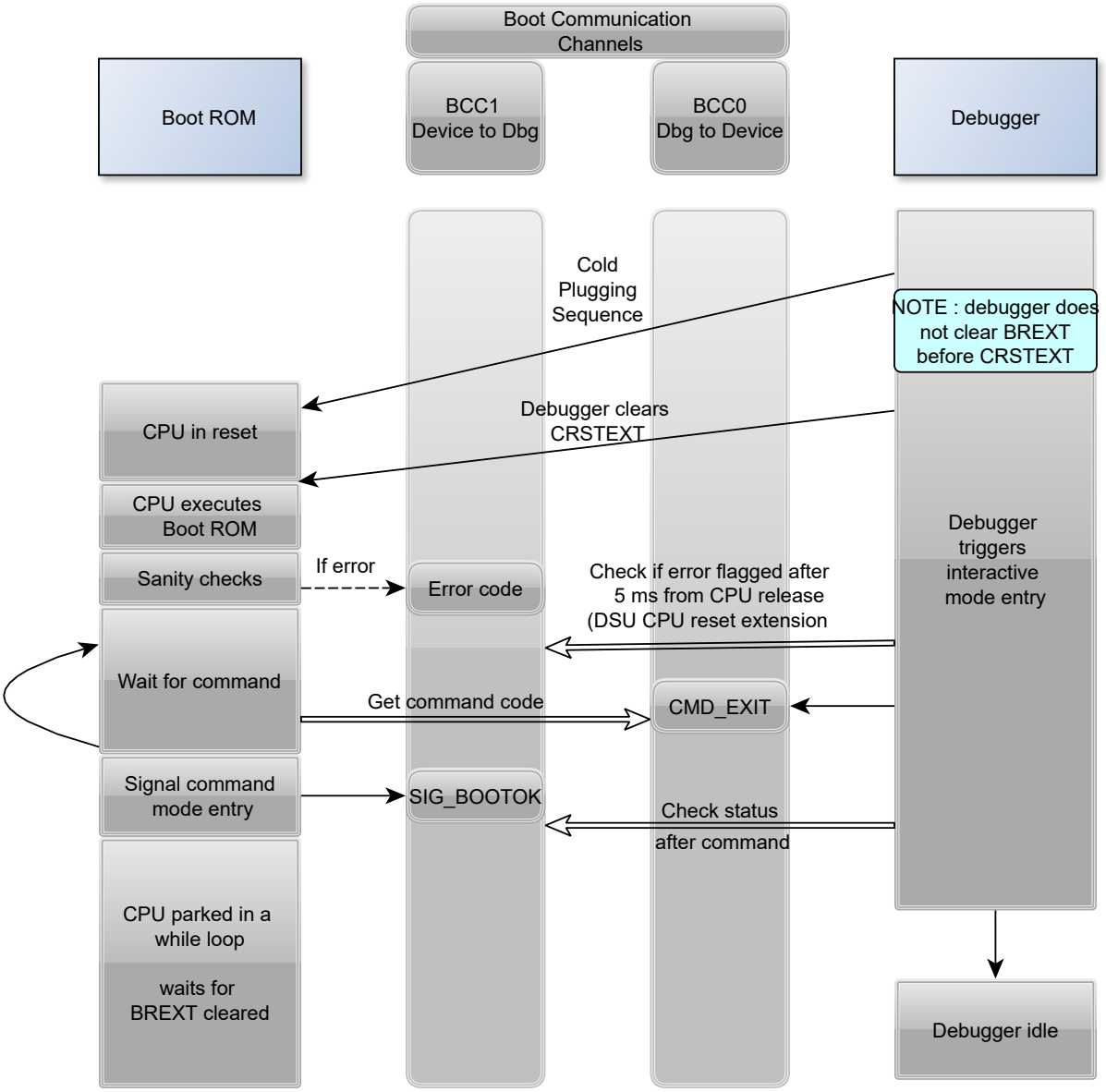


Figure 13-15. CMD_EXIT to Park mode flow diagram



13.4.5.3 System Reset Request (CMD_RESET)

This command allows resetting the system using a system reset request, because the reset is executed immediately after receiving the command, no reply is sent to the debugger.

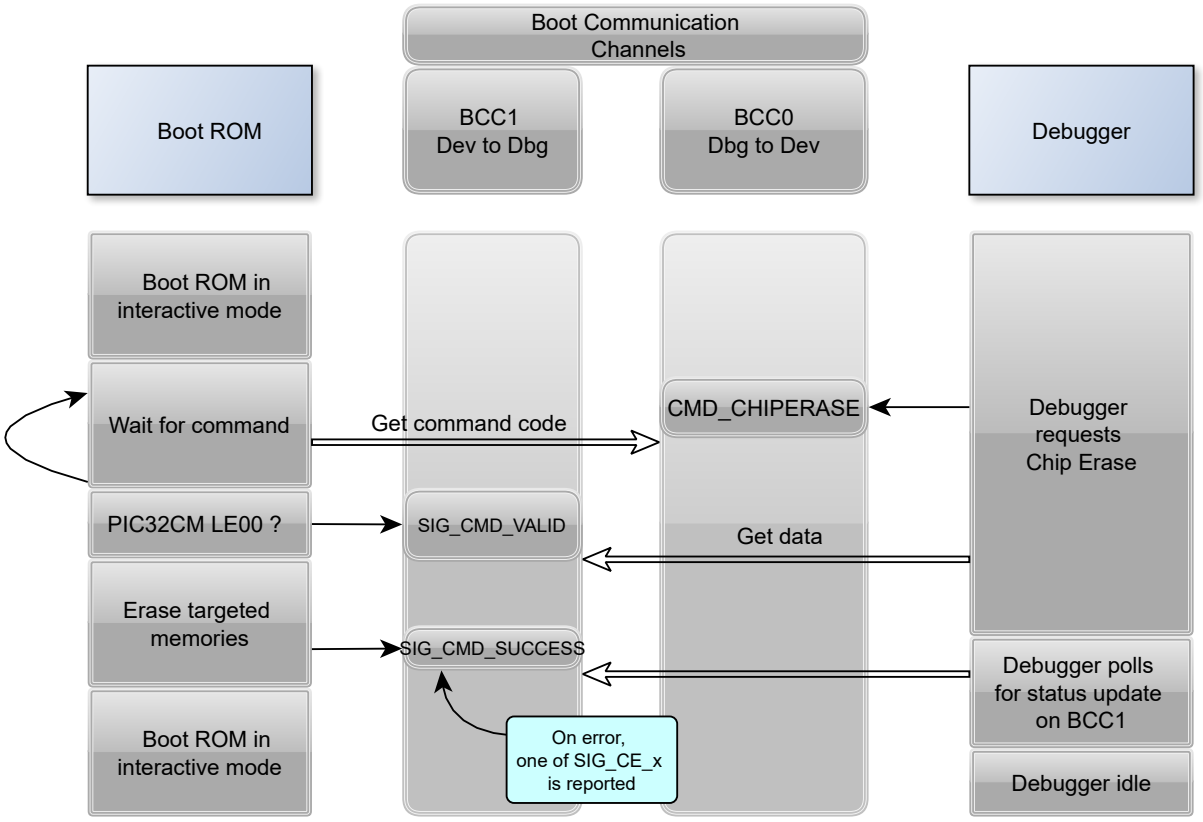
After reset, the CPU executes the Boot ROM code from the beginning.

13.4.5.4 PIC32CM LE00 Chip Erase (CMD_CHIPERASE and CMD_CHIPERASE_ALL)

The CMD_CHIPERASE command erases the entire device except BOOT area, and reverts to Debug Access Level 2.

The CMD_CHIPERASE_ALL command erases the entire device including the BOOT area, and reverts to Debug Access Level 2.

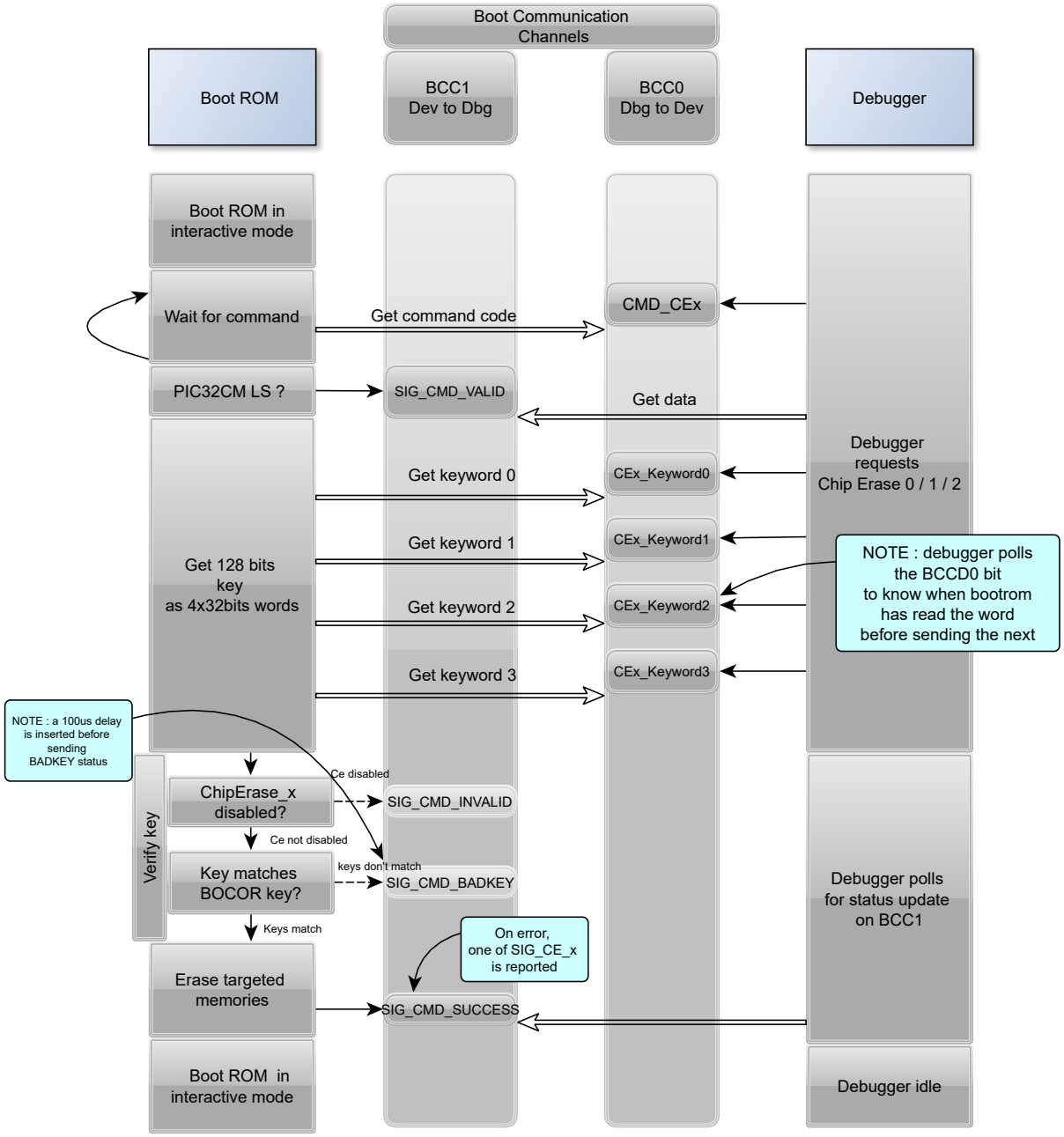
Figure 13-16. CMD_CHIPERASE and CM_CHIPERASE_ALL Flow diagram



13.4.5.5 PIC32CM LS00/LS60 Chip Erase (CMD_CEx)

The CMD_CEx commands are used to erase specific part of the device and to increase the Debug Access Level.

Figure 13-17. CMD_CEx Flow diagram



13.4.5.6 NVM Memory Regions Integrity Checks (CMD_CRC)

The Boot ROM provides a way to check the integrity of the embedded non-volatile memories which may be of interest in case of a failure analysis.

This requires the user to place tables describing the memory area to be checked with their expected CRC values.

Note: During this integrity check process, the debugger sends the CRC table address to the device.

Important: The tables must be programmed by the programming tool in addition to the application binaries.

13.4.5.6.1 CRC Table format

Table 13-6. CRC Table Fields Description

Description	Header	Start Address (1)	Size in bytes (2)	Expected value (3)
Field	HDR	ADDR	SIZE	REFVAL
Offset	0x0	0x4	0x8	0xC
Value	0x43524349	0x00000000	0x100	0xAABBCCDD

Note 1: ADDR must be a multiple of 4 (only ADDR[31:2] are used).

Note 2: SIZE must be a multiple of 4 (only SIZE[31:2] are used).

Note 3: The expected value is the computed CRC32 value of the memory target.

13.4.5.6.2 Requirements

- Each table occupies 16 bytes in memory.
- The table must start at a 16 byte aligned address. (i.e. 0xXXXXXXX0)
- The table must be placed in the same memory region as its target memory range. (i.e. a table placed in the Secure Flash (APPLICATION region) can only target Secure Flash (APPLICATION region) memory addresses). The exceptions to this rule are as follows:
  - For PIC32CM LE00: all non-volatile memories are considered as a single region (e.g. a table located in Data Flash can target Flash)
  - For PIC32CM LS00/LS60: ANSC and BNSC regions are considered to belong to the same region as their “parent” region: AS for ANSC and BOOTPROT for BNSC.

13.4.5.6.3 CRC Command Key

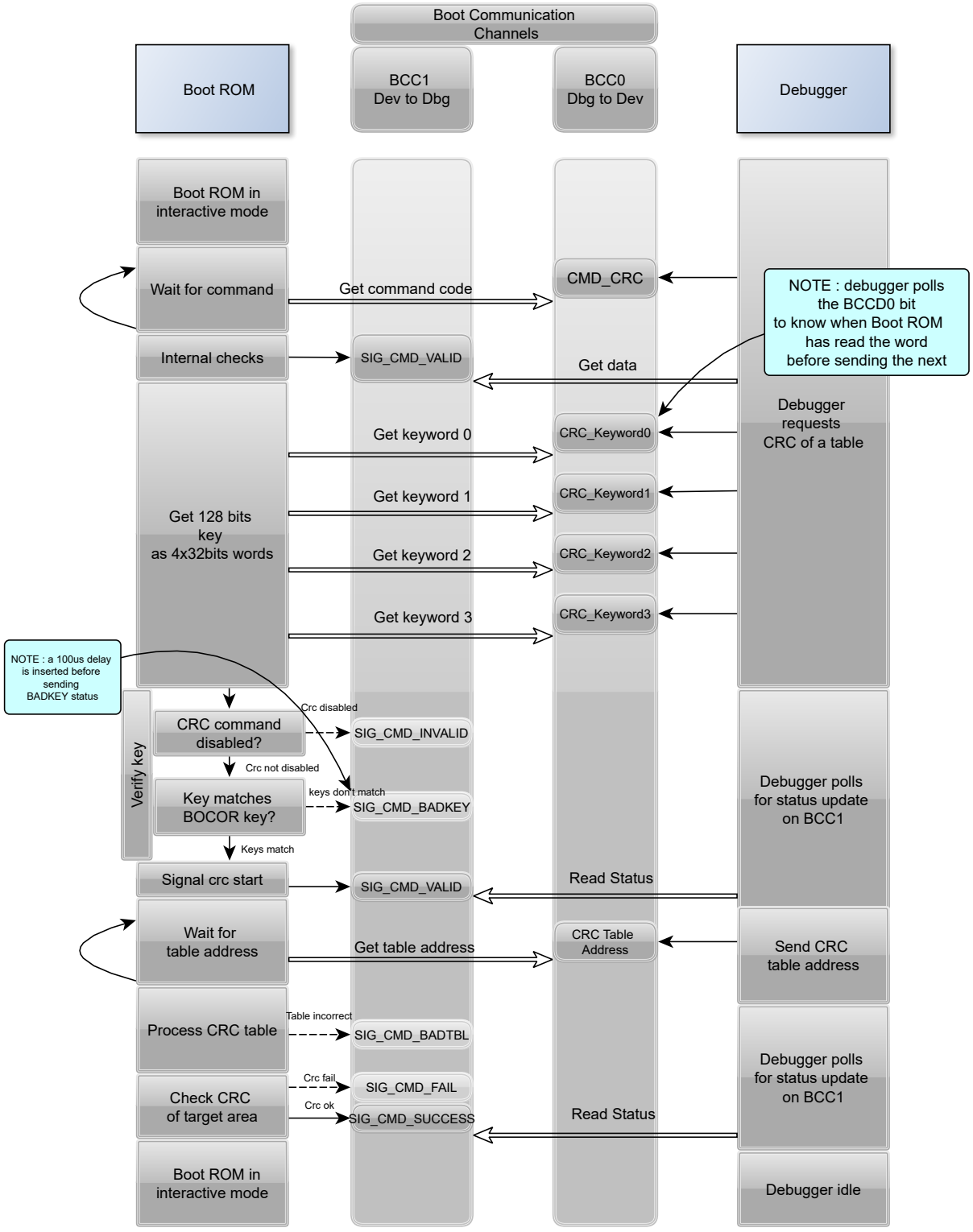
The CRC command (CMD_ CRC) requires an access key (CRCKEY) which is in the NVM BOCOR row at: [0x80C040:0x80C04F]:

BOCOR Offeset	Bit Position	Name
0x40-0x4F	639:512	CRCKEY

Similar to the ChipErase keys, the key can be set to all ‘0s’ to prevent any access to the command.

13.4.5.6.4 CMD_CRC

Figure 13-18. CMD_CRC Flow diagram



13.4.5.7 Random Session Key Generation (CMD_DCEK) - PIC32CM LS00/LS60 only

This command allows using a challenge-response scheme to prevent exposure of the keys in clear text on the debugger communication lines.

The different keys sent by the debugger during the Boot ROM for Chip Erase (CMD_CEx) and CRC (CMD_CRC) commands execution are:

- CRCKEY for CMD_CRC command
- CEKEYx for CMD_CEx commands

Note: The CMD_DCEK command has no effect on the PIC32CM LE00, the key derivation will not be enabled.

The random challenge value is generated using the TRNG of the device. It is generated once the CMD_DCEK is received and communicated to the debugger.

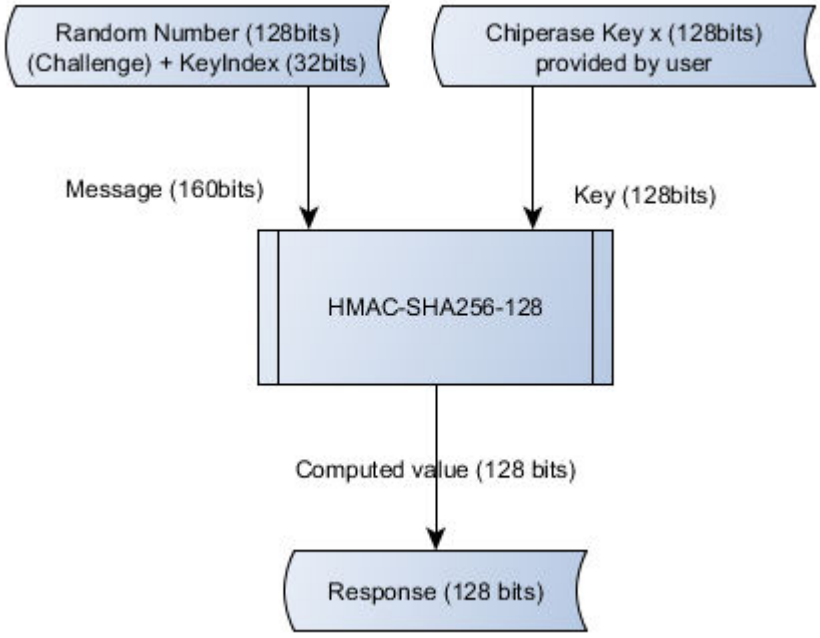
The next CMD_CEx or CMD_CRC commands will expect the key value to be replaced by the computed response corresponding to the challenge.

The challenge value is valid only for the next CMD_CEx/ CMD_CRC command.

Before sending a new CMD_CEx/ CMD_CRC command, a CMD_DCEK shall be used to re-enable the challenge-response scheme a get a new challenge value.

On the debugger side, the response shall be computed using the following algorithm:

Figure 13-19. Debugger Algorithm



Where KeyIndex is:

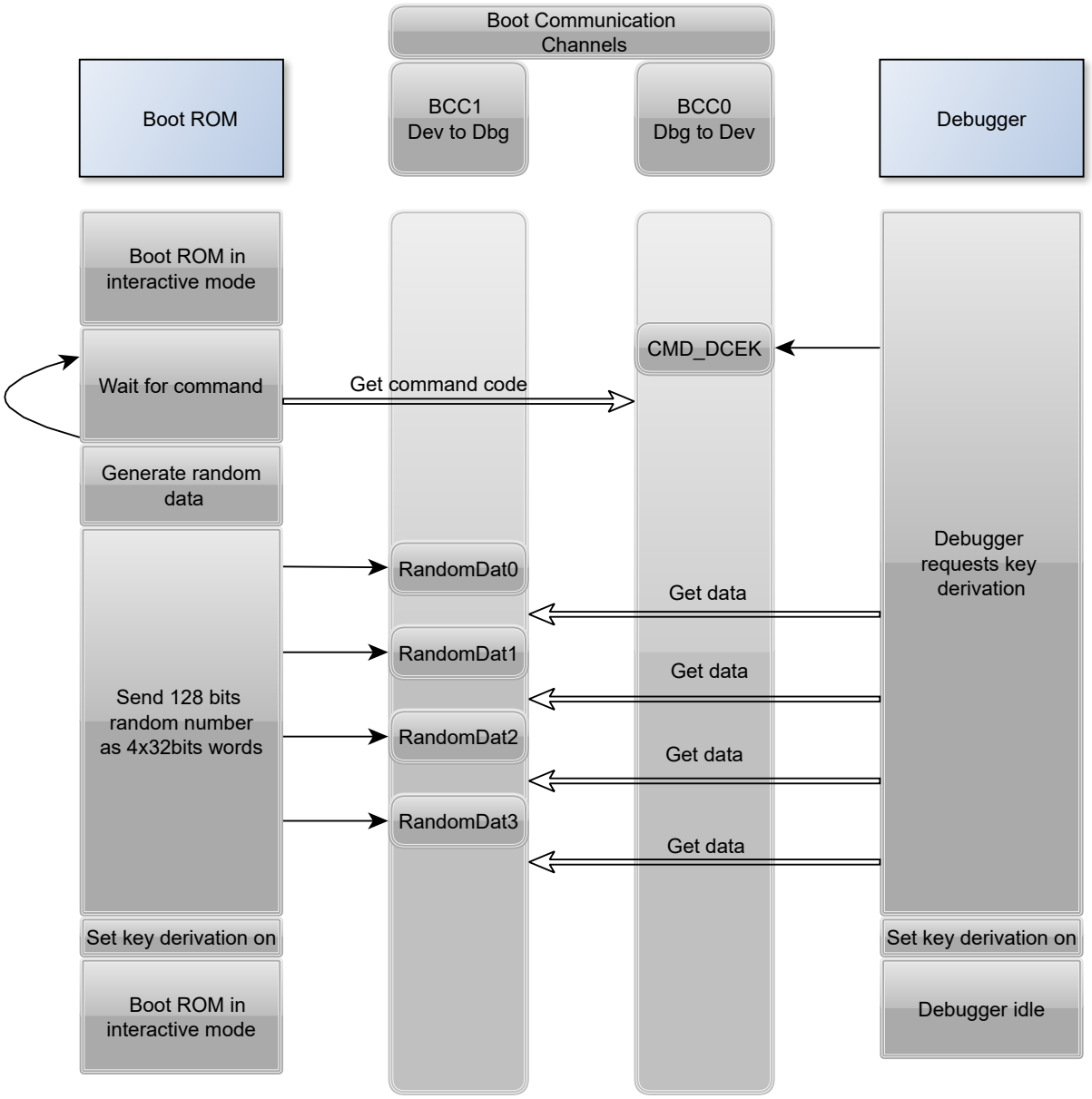
- 0 for ChipErase_NS
- 1 for ChipErase_S
- 2 for ChipErase_ALL
- 3 for CRC Command

Notes:

- HMAC is described in FIPS PUB 198-1.
- The hash used for HMAC is SHA-256.
- The output of the HMAC-SHA-256 is truncated to obtain an HMAC-SHA-256-128 as explained in RFC4868.

13.4.5.7.1 CMD_DCEK (PIC32CM LS00/LS60 only)

Figure 13-20. CMD_DCEK Flow diagram



13.4.5.8 NVM Configuration Rows Content Checks (CMD_RAUX)

The Boot ROM provides a way to check the content of the NVM Configuration rows.

When device is secured (DAL0), the fuse configuration can still be read by the debugger using the Read Auxiliary command (CMD_RAUX).

The following areas are accessible:

Table 13-7. Accessible Memory Range by Read Auxiliary Row Command(1)

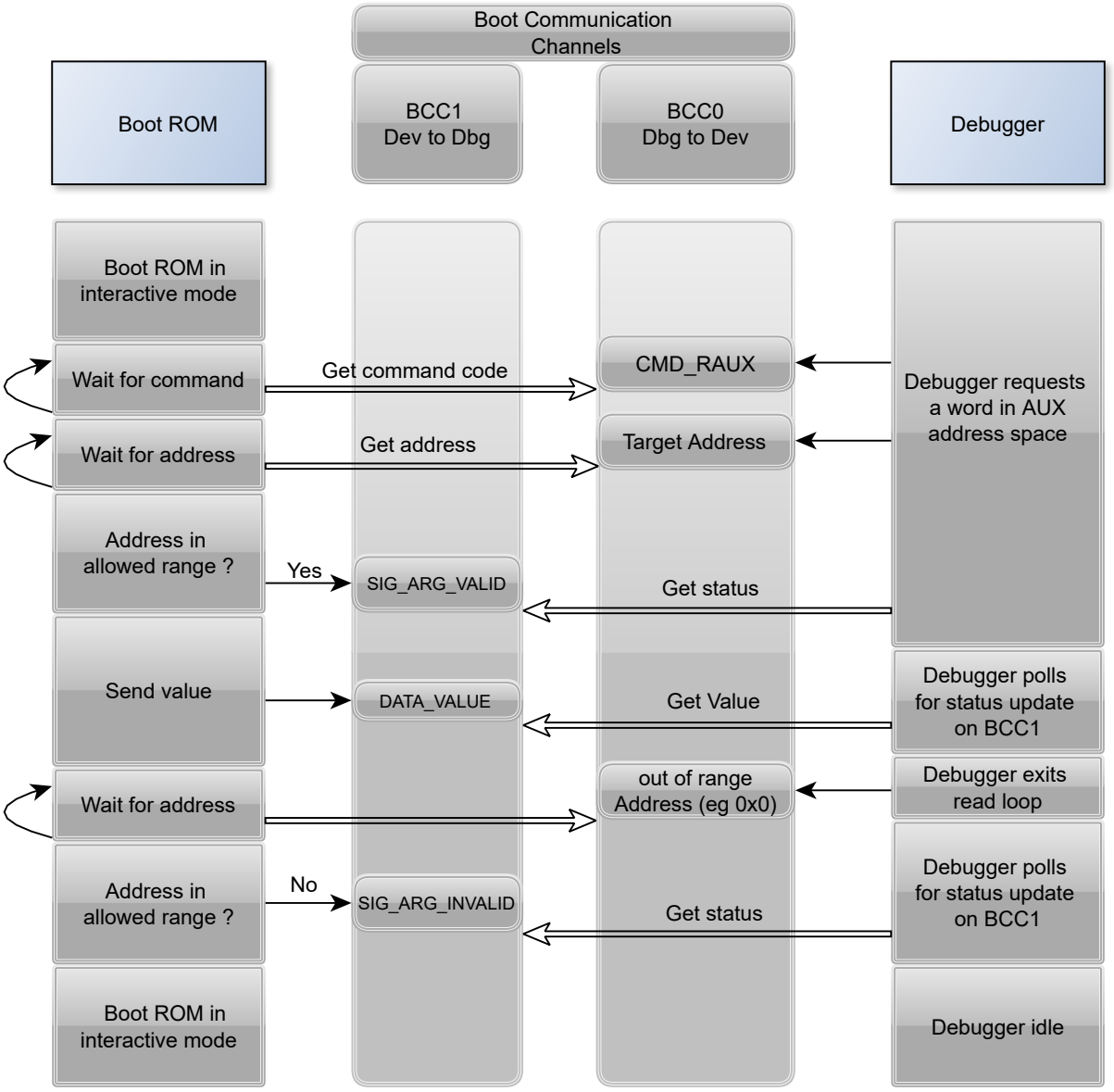
Area	Start address	End address
User row (UROW)	0x00804000	0x008040FF
Software Calibration row	0x00806020	0x00806023

Note:

1. Boot Configuration row (BOCOR) is not accessible by the Read Auxiliary Row command.

13.4.5.8.1 CMD_RAUX

Figure 13-21. CMD_RAUX Flow diagram



Note: After the CMD_RAUX is sent, the debugger can read multiple data, the read loop is exit when an out of range address is sent.

13.4.5.9 Boot Interactive Mode Commands

Table 13-8. Boot Interactive Mode Commands

Command Name	Description	Command prefix	Command
CMD_INIT	Entering Interactive Mode	0x444247	55
CMD_EXIT	Exit Interactive Mode	0x444247	AA
CMD_RESET	System Reset Request	0x444247	52

.....continued

Command Name	Description	Command prefix	Command
CMD_CE0	ChipErase_NS for PIC32CM LS00/LS60	0x444247	E0
CMD_CE1	ChipErase_S for PIC32CM LS00/LS60	0x444247	E1
CMD_CE2	ChipErase_ALL for PIC32CM LS00/LS60	0x444247	E2
CMD_CHIPERASE	ChipErase for PIC32CM LE00	0x444247	E3
CMD_CHIPERASE_ALL	ChipErase_ALL for PIC32CM LE00	0x444247	E4
CMD_CRC	NVM Memory Regions Integrity Checks	0x444247	C0
CMD_DCEK	Random Session Key Generation for PIC32CM LS00/LS60	0x444247	44
CMD_RAUX	NVM Configuration Rows Integrity Checks	0x444247	4C

13.4.5.10 Boot Interactive Mode Status

Table 13-9. Boot Interactive Mode Status

Status Name	Description	Status prefix	Status coding
SIG_NO	No Error	0xEC0000	00
SIG_SAN_FFF	Fresh from factory error	0xEC0000	10
SIG_SAN_UROW	UROW checksum error	0xEC0000	11
SIG_SAN_SECEN	SECEN parameter error	0xEC0000	12
SIG_SAN_BOCOR	BOCOR checksum error	0xEC0000	13
SIG_SAN_BOOTPROT	BOOTPROT parameter error	0xEC0000	14
SIG_SAN_NOSECREG	No secure register parameter error	0xEC0000	15
SIG_COMM	Debugger start communication command	0xEC0000	20
SIG_CMD_SUCCESS	Debugger command success	0xEC0000	21
SIG_CMD_FAIL	Debugger command fail	0xEC0000	22
SIG_CMD_BADKEY	Debugger bad key	0xEC0000	23
SIG_CMD_VALID	Valid command	0xEC0000	24
SIG_CMD_INVALID	Invalid command	0xEC0000	25
SIG_ARG_VALID	Valid argument	0xEC0000	26
SIG_ARG_INVALID	Invalid argument	0xEC0000	27
SIG_CE_CVM	Chip erase error: CVM	0xEC0000	30
SIG_CE_ARRAY_ERASEFAIL	Chip erase error: array erase fail	0xEC0000	31
SIG_CE_ARRAY_NVME	Chip erase error: array NVME	0xEC0000	32
SIG_CE_DATA_ERASEFAIL	Chip erase error: data erase fail	0xEC0000	33
SIG_CE_DATA_NVME	Chip erase error: data NVME	0xEC0000	34
SIG_CE_BCUR	Chip erase error: BOCOR, UROW	0xEC0000	35
SIG_CE_BC	Chip erase error: BC check	0xEC0000	36
SIG_BOOT_OPT	BOOTOPT parameter error	0xEC0000	40
SIG_BOOT_ERR	Boot image hash verify fail	0xEC0000	41
SIG_BOCOR_HASH	BOCOR hash error	0xEC0000	42
SIG_CRC_BADTBL	Bad CRC table	0xEC0000	50
SIG_SECEN0_ERR	PAC or IDAU cfg check failure	0xEC0000	60
SIG_SECEN1_ERR	PAC or IDAU cfg check failure	0xEC0000	61

.....continued

Status Name	Description	Status prefix	Status coding
SIG_EXIT_ERR	Exit: BC or check error	0xEC0000	70
SIG_HARDFAULT	Hardfault error	0xEC0000	F0
SIG_BOOTOK	Boot ROM ok to exit	0xEC0000	39

13.4.6 CPU Park mode

This mode allows the debugger to get access to the resources of the device during Boot ROM execution while the CPU is trapped in a while loop. The debug access level when entering that mode corresponds to the DAL value which is programmed in the device.

Important: This mode is the recommended way to enter a debugging session in a safe way even if it is also possible to launch a debug session when the application is running.

This mode is reached by sending the Exit command (CMD_EXIT) without clearing the DSU.STATUSA.BREXT bit to the Boot ROM.

As soon as the BREXT bit is cleared, the device exits this state and performs a system reset.

At this point, the MPU is still enabled and prevents software execution elsewhere than in Boot ROM region.

If the host needs to run software on the device, MPU must be disabled by accessing the Cortex-M23 MPU CTRL register with the debugger.

14. Implementation Defined Attribution Unit (IDAU)

Important: The IDAU peripheral is only present on PIC32CM LS00/LS60.

The Cortex-M23 provides two ways for handling the transactions from the different hosts (core, debugger, and so on):

- The Secure Attribution Unit (SAU): not present on PIC32CM LS00/LS60.
- The Implementation Defined Attribution Unit (IDAU): chosen for PIC32CM LS00/LS60.

The SAU is a Memory Protection Unit (MPU) like hardware embedded in the core. The role of the SAU is to manage all the Secure and Non-Secure transactions coming from the core and the debugger. However, using the SAU implies that the security configuration must be propagated somewhere else in the MCU architecture for security awareness.

The IDAU is a hardware unit external to the core, which is used to indicate to the processor if a particular memory region is Secure (S), Non-Secure Callable (NSC), or Non-Secure (NS). It can also mark a memory region to be exempted from security checking. The Cortex-M23 checks each access (fetch or data) in the IDAU, which returns the privilege information about that specific address. If the access is not permitted, the CPU enters a HardFault exception.

Table 14-1. IDAU Memory Attribution Definition

Attribute	Description
Non-Secure	Memory can be accessed in Secure or Non-Secure state.
Secure	Memory can only be accessed in Secure state. It cannot be called from Non-Secure state.
Non-Secure callable	Memory can only be accessed in Secure state, but can be called from Non-Secure state.
Exempt	No attribution check will be done, and the operation will take place on the bus

The following table provides the PIC32CM LS00/LS60 memory space security attributions:

Table 14-2. PIC32CM LS00/LS60 Memory Space Security Attributions (IDAU.CTRL.ENABLE == 1) (1)

Memory region	Attribute
Secure Flash (BOOT region)	Secure
Non-Secure Callable Flash (BOOT region)	Non-secure callable
Secure Flash (APPLICATION region)	Secure
Non-Secure Callable Flash (APPLICATION region)	Non-secure callable
Non-Secure Flash (APPLICATION region)	Non-secure
Secure Data Flash	Secure
Non-Secure Data Flash	Non-secure
NVM Rows	Exempt - eXecute Never Secure (R/W access) Non-Secure (Discarded for BOCOR, Read-only for the others)
Boot ROM	Secure Execute-only for CRYA functions
Secure SRAM	Secure
Non-Secure SRAM	Non-secure
Peripherals	Exempt - eXecute Never
IOBUS	Exempt - eXecute Never

.....continued

Memory region	Attribute
Others (Reserved, Undefined...)	Secure

Note:

1. Exempt property relates only to the IDAU: Peripherals, IOBUS as well as NVM rows security is directly done at the peripheral level.

Table 14-3. PIC32CM LS00/LS60 Memory Space Security Attributions (IDAU.CTRL.ENABLE == 0)

Memory region	Attribute
Boot ROM	Secure Execute-only for CRYA functions
Others	Exempt

In addition, the IDAU propagates all the security configurations to:

- The memory controllers: Flash, Data Flash and SRAM embedded memories can be split in sub-regions, which are reserved either for the Secure or for the Non-Secure application.
- The peripheral controllers using the Peripherals Access Controller (PAC) which can allocate each peripheral either to the Secure or to the Non-Secure application, with the exception of the IDAU and DSU which have a fixed security attribute:

Important:

1. The IDAU peripheral is always Secured.
2. The DSU peripheral is always Non-Secured.
3. The PAC and NVMCTRL peripherals are always Secured if BOCOR.SECCFGLOCK = 1 after exiting Boot ROM.

Software can check the privilege state of a memory location by using the Cortex-M23 Test Target instructions: TT, TTT, TTA, and TTAT.

The memory location is referenced using the Cortex-M23 IREGION bit field, which specifies the IDAU region number (Refer to the *ARMv8-M Architecture Reference Manual* for more information).

Table 14-4. PIC32CM LS00/LS60 IDAU Region Number for TT, TTT, TTA and TTAT Cortex-M23 Instructions (IDAU.CTRL.ENABLE == 1)

Memory Region	IDAU Region Number for TTx Instructions (IREGION bits)
Secure Flash (BOOT region)	0x01
Non-Secure Callable Flash (BOOT region)	0x02
Secure Flash (APPLICATION region)	0x03
Non-Secure Callable Flash (APPLICATION region)	0x04
Non-Secure Flash (APPLICATION region)	0x05
Secure Data Flash	0x06
Non-Secure Data Flash	0x07
NVM User Rows	0x00 (invalid)
Boot ROM	0x08
Secure SRAM	0x09
Non-Secure SRAM	0x0A
Peripherals	0x00 (invalid)
IOBUS	0x00 (invalid)

.....continued	
Memory Region	**IDAU Region Number for TTx Instructions (IREGION bits)**
Others (Reserved, Undefined...)	0x00 (invalid)

Table 14-5. PIC32CM LS00/LS60 IDAU Region Number for TT, TTT, TTA and TTAT Cortex-M23 Instructions (IDAU.CTRL.ENABLE == 0)

Memory Region	IDAU Region Number for TTx Instructions (IREGION bits)
Boot ROM	0x08
Others	0x00 (invalid)

14.1 Peripheral Dependencies

Table 14-6. IDAU Configuration Summary

Peripheral name	Base address	MCLK AHB/APB Clocks	PAC Peripheral Identifier Index (PAC.WRCTRL)	Power Domain (PM.STDBYCFG)
IDAU	0x41000000	CLK_IDAU_APB	32	PDSW

14.2 Register Summary

Refer to the Registers Description section for more details on register properties and access permissions.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRL	7:0							ENABLE	
0x01	SECCTRL	7:0						RXN	Reserved	SCFGWEN
0x02 ... 0x03	Reserved									
0x04	SCFGB	31:24		BOOTPROT[10:4]						
		23:16	BOOTPROT[3:0]				BNSC[8:5]			
		15:8	BNSC[4:0]							
		7:0								
0x08	SCFGA	31:24						DS[6:4]		
		23:16	DS[3:0]				ANSC[8:5]			
		15:8	ANSC[4:0]					AS[10:8]		
		7:0	AS[7:0]							
0x0C	SCFGR	15:8								RS[8]
		7:0	RS[7:0]							

14.2.1 Control

Name: CTRL
Offset: 0x00
Reset: 0x02
Property: PAC Write-Protection

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

Bit 1 – ENABLE Enable
The ENABLE bit allows to apply the different IDAU security configurations of the SECCTRL, SCFGB, SCFGA and SCFGR registers.
It is mandatory to disable the IDAU before modifying any of the IDAU security configurations.

Important:
- SECCTRL.SCFGWEN must not be cleared when the IDAU is disabled as it will lock the CTRL register, which will prevent from enabling the IDAU and apply the new security configurations.
- Users must ensure when enabling the IDAU, that the secure software code running out of the Flash BOOT region is still entirely part of that region as no consistency checks are done on the new applied security memory mapping.

Value	Description
0	The IDAU is disabled. The IDAU security configurations of the SECCTRL, SCFGB, SCFGA and SCFGR registers can be updated.
1	The IDAU is enabled. The IDAU security configurations are applied to the system.

14.2.2 Secure Boot Configuration

Name: SCFGB
Offset: 0x04
Reset: 'x' initially determined from NVM Boot Configuration Row (BOCOR) after Reset.
Property: PAC Write-Protection

This register is loaded from BOCOR during Boot ROM execution.

Important: if BOCOR.SECCFGLOCK == 0 after exiting the Boot ROM:

- The secure software code of the Flash BOOT region, before passing control on to the secure software code of the Flash APPLICATION region, must lock the IDAU memory security configurations by clearing the IDAU.SECCTRL.SCFGWEN bit.
- Write accesses (W*) are allowed.

Bit	31	30	29	28	27	26	25	24
		BOOTPROT[10:4]						
Access		R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*
Reset		x	x	x	x	x	x	x

Bit	23	22	21	20	19	18	17	16
	BOOTPROT[3:0]				BNSC[8:5]			
Access	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*
Reset	x	x	x	x	x	x	x	x

Bit	15	14	13	12	11	10	9	8
	BNSC[4:0]							
Access	R/W*	R/W*	R/W*	R/W*	R/W*			
Reset	x	x	x	x	x			

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bits 30:20 – BOOTPROT[10:0] Secure Flash (BOOTPROT region) size
This field defines the size of the Secure Flash (BOOTPROT region) = BOOTPROT*0x100 bytes.
The Secure Flash (BOOTPROT region) is composed by:

1. The Secure (S) Flash BOOT region
2. The Non-Secure Callable (NSC) Flash BOOT region

Bits 19:11 – BNSC[8:0] Non-Secure Callable (NSC) Flash (BNSC region) size
This field defines the size of the Non-Secure Callable (NSC) Flash BOOT region = BNSC*0x20 bytes.

14.2.3 Secure Application Configuration

Name: SCFGA
Offset: 0x08
Reset: x initially determined from NVM User Row after reset
Property: PAC Write-Protection

This register is loaded from UROW during Boot ROM execution.

Important: if BOCOR.SECCFGLOCK = 0 after exiting the Boot ROM:

- The secure software code of the Flash BOOT region, before passing control on to the secure software code of the Flash APPLICATION region, must lock the IDAU memory security configurations by clearing the IDAU.SECCTRL.SCFGWEN bit.
- Write accesses (W*) are allowed.

Bit	31	30	29	28	27	26	25	24
						DS[6:4]		
Access						R/W*	R/W*	R/W*
Reset						x	x	x

Bit	23	22	21	20	19	18	17	16
	DS[3:0]				ANSC[8:5]			
Access	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*
Reset	x	x	x	x	x	x	x	x

Bit	15	14	13	12	11	10	9	8
	ANSC[4:0]					AS[10:8]		
Access	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*
Reset	x	x	x	x	x	x	x	x

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

Bits 26:20 – DS[6:0] Secure Data Flash (DS region) size
This field defines the size of the Secure (S) Data Flash region = DS*0x100 bytes.

Bits 19:11 – ANSC[8:0] Non-Secure Callable (NSC) Flash (ANSC region) size
This field defines the size of the Non-Secure Callable (NSC) Flash APPLICATION region = ANSC*0x20 bytes.

Bits 10:0 – AS[10:0] Secure Flash (AS region) size
This field defines the size of the AS region = AS*0x100 bytes.
The AS region is composed by:

1. The Secure (S) Flash APPLICATION region
2. The Non-Secure Callable (NSC) Flash APPLICATION region

14.2.4 Secure SRAM Configuration

Name: SCFGR
Offset: 0x0C
Reset: 0x0000
Property: PAC Write-Protection

This register is loaded from UROW during Boot ROM execution.

Important: if BOCOR.SECCFGLOCK == 0 after exiting the Boot ROM:

- The secure software code of the Flash BOOT region, before passing control on to the secure software code of the Flash APPLICATION region, must lock the IDAU memory security configurations by clearing the IDAU.SECCTRL.SCFGWEN bit.
- Write accesses (W*) are allowed.

Bit	15	14	13	12	11	10	9	8
								RS[8]
Access								R/W*
Reset								x

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

Bits 8:0 – RS[8:0] Secure SRAM (RS region) Size
This field defines the size of the Secure (S) SRAM region = RS*0x80 bytes.

14.2.5 Security Control

Name: SECCTRL
Offset: 0x01
Reset: x/y initially determined after reset from NVM User Row (UROW) / BOCOR.SECCFGLOCK
Property: PAC Write-Protection

Important: if BOCOR.SECCFGLOCK == 0 after exiting the Boot ROM:

- The secure boot flash code, before exiting, must lock the memory security configurations by clearing the IDAU.SECCTRL.SCFGWEN bit.
- Write accesses (W*) are allowed.

Bit	7	6	5	4	3	2	1	0
						RXN	Reserved	SCFGWEN
Access						R/W*	R	R/W*
Reset						x	0	y

Bit 2 – RXN SRAM eXecute Never
This bit status is loaded from UROW at boot.

Value	Description
0	Execution out of SRAM is authorized.
1	Execution out of SRAM is not authorized.

Bit 1 – Reserved Reserved

Bit 0 – SCFGWEN Security Configuration Write Enable
After Boot ROM execution, this bit is:

- Cleared if BOCOR.SECCFGLOCK == 1
- Set if BOCOR.SECCFGLOCK == 0

Important:

- If SCFGWEN = 1, the secure software code of the Flash BOOT region must clear this bit before passing control on to the secure software code of the Flash APPLICATION region in order to lock the IDAU security configurations.
- SCFGWEN must be enabled only when the IDAU is enabled (CTRL.ENABLE == 1).

Value	Description
0	CTRL, SCFGB, SCFGA, SCFGR and SECCTRL.SCFGWEN cannot be written until the next reset.
1	CTRL, SCFGB, SCFGA, SCFGR and SECCTRL.SCFGWEN can be written.

15. Peripheral Access Controller (PAC)

15.1 Overview

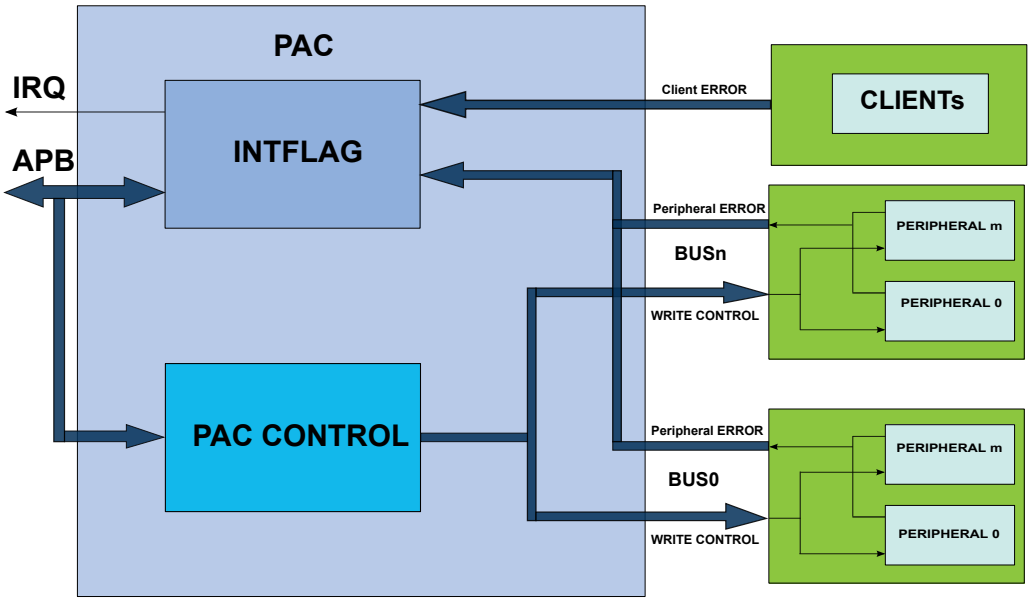
The Peripheral Access Controller (PAC) provides an interface for the locking and unlocking and for managing security attribution of peripheral registers within the device. It reports all violations that could happen when accessing a peripheral; write protected register access, illegal access, enable protected register access when clock synchronization or software reset is on-going. 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-existing address is detected.

15.2 Features

- Manages write protection access and reports access errors for the peripheral modules or bridges
- Manages security attribution for the peripheral modules (PIC32CM LS00)

15.3 Block Diagram

Figure 15-1. PAC Block Diagram



15.4 Peripheral Dependencies

Table 15-1. PAC Configuration Summary

Peripheral name	Base address	NVIC IRQ Index	MCLK AHB/APB Clocks	PAC Peripheral Identifier Index (PAC.WRCTRL)	Events (EVSYS)		Power Domain (PM.STDBYCFG)
					Users (EVSYS.USER)	Generators (EVSYS.CHANNEL)	
PAC	0x40000000	30: ERR	CLK_PAC_AHB CLK_PAC_APB	0	—	91: ERR	PDSW

15.5 Functional Description

15.5.1 Principle of Operation

The Peripheral Access Control module allows the user to set a write protection or security attribution 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, slaves bus errors can be also reported in the cases where reserved area is accessed by the application.

15.5.2 Basic Operation

15.5.2.1 Initialization, Enabling and Resetting

The PAC is always enabled after reset.

Only a hardware reset will reset the PAC module.

15.5.2.2 Operations

The PAC module allows the user to set, clear or lock the write protection status and security attribution of all peripherals on all Peripheral Bridges.

If a peripheral register violation occurs, the Peripheral Interrupt Flag n registers (INTFLAGn) are updated to inform the user on 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. The corresponding Peripheral Non-Secure Status n register (NONSECn) gives the state of the security attribution for all peripherals connected to the corresponding Peripheral Bridge n. Refer to 15.5.2.3. Peripheral Access Errors for details.

The PAC module also report the errors occurring at client bus level when an access to reserved area is detected. AHB Client Bus Interrupt Flag register (INTFLAGAHB) informs the user on the status of the violation in the corresponding client. Refer to the AHB Client Bus Errors for details.

15.5.2.3 Peripheral Access Errors

The following events will 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 normally. If a peripheral is write protected and if a write access is attempted, data will not be written and peripheral returns an access error. The corresponding interrupt flag bit in the INTFLAGn register will be set.
- Illegal access: Access to an unimplemented register within the module.
- Synchronized write error: For write-synchronized registers an error will be reported if the register is written while a Synchronization is ongoing.

When any of the INTFLAGn registers bit are set, an interrupt will be requested if the PAC interrupt enable bit is set.

15.5.2.4 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 and WRCTRL.KEY. The WRCTRL.PERID is an unique identifier corresponding to a peripheral. The WRCTRL.KEY is a key value that defines the operation to be done on the control access bit. These operations can be "clear protection", "set protection" and "set and lock protection bit".

The "clear protection" operation will remove the write access protection for the peripheral selected by WRCTRL.PERID. Write accesses are allowed for the registers in this peripheral.

The "set protection" operation will set the write access protection for the peripheral selected by WRCTRL.PERID. Write accesses are not allowed for the registers with write protection property in this peripheral.

The "set and lock protection" operation will set 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.

15.5.2.5 Write Access Protection Management Errors

Only word-wise writes to the WRCTRL register will effectively change the access protection. Other type of accesses will have no effect and will cause a PAC write access error. This error is reported in the INTFLAGn.PAC bit corresponding to the PAC module.

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 then the PAC returns an error, and 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 and conversely. However in applications where a write protected peripheral is used in several contexts, for example, interrupt, care should be taken so that either the interrupt can not happen while the main application or other interrupt levels manipulates 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) will set the INTFLAGn.PAC flag.

15.5.2.6 PIC32CM LS00/LS60 Security Attribution Management

The peripheral security attribution status can be read from the corresponding NONSECn register.

15.5.2.7 PIC32CM LS00/LS60 Security Attribution Management Errors

The errors generated while accessing the PAC module registers (for example, key error, double protect error) will set the INTFLAGn.PAC flag.

15.5.2.8 AHB Client Bus Errors

The PAC module reports errors occurring at the AHB Client bus level. These errors are generated when an access is performed at an address where no client (bridge or peripheral) is mapped or where Non-Secure accesses are prohibited. These errors are reported in the corresponding bits of the INTFLAGAHB register.

15.5.2.9 Generating Events

The PAC module can also generate an event when any of the Interrupt Flag registers bit are set. To enable the PAC event generation, the control bit EVCTRL.ERREO must be set a '1'.

15.5.3 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. 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 on 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 that interrupts must be globally enabled for interrupt requests to be generated.

For further information, refer to 22.5.3.3. Sleep Mode Controller and the 11.2. Nested Vector Interrupt Controller.

15.5.4 Events

The PAC can generate 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.

Refer to the 32. Event System (EVSYS) for further information.

15.5.5 Sleep Mode Operation

In Sleep mode, the PAC is kept enabled if an available bus host (CPU, DMA) is running. The PAC will continue to catch access errors from the module and generate interrupts or events.

15.5.6 Debug Operation

When the CPU is halted in Debug mode, write protection of all peripherals is disabled and the PAC continues normal operation.

15.5.7 PIC32CM LS00/LS60 Secure and Non-Secure Read/Write Accesses

Non-Secure write to EVCTRL, INTENCLR, INTENSET, INTFLAGAHB, INTFLAGx, and NONSECx registers is prohibited.

Non-Secure read to EVCTRL, INTENCLR, INTENSET, INTFLAGAHB, and INTFLAGx registers will return zero with no error resulting.

Non-secure write to a bit of STATUSx registers (by writing to the WRCTRL register) is prohibited if the corresponding bit in NONSECx is zero.

STATUSx bits relating to secure peripherals (i.e., the corresponding bits in NONSECx are zero), read as zero in Non-Secure mode, with no error resulting.

15.6 Register Summary

Refer to the Registers Description section for more details on register properties and access permissions.

Important: (PIC32CM LS00 only)
The peripheral register map is automatically duplicated in a Secure and Non-Secure alias:

- The Non-Secure alias is at the peripheral base address
- The Secure alias is located at the peripheral base address + 0x200

Refer to Mix-Secure Peripherals for more information on register access rights.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	WRCTRL	31:24								
		23:16	KEY[7:0]							
		15:8	PERID[15:8]							
		7:0	PERID[7:0]							
0x04	EVCTRL	7:0								ERREO
0x05 ... 0x07	Reserved									
0x08	INTENCLR	7:0								ERR
0x09	INTENSET	7:0								ERR
0x0A ... 0x0F	Reserved									
0x10	INTFLAGAHB	31:24								
		23:16								
		15:8								
		7:0	BROM	HSRAMDSU	HSRAMDMAC	HSRAMCPU	APBC	APBB	APBA	FLASH
0x14	INTFLAGA	31:24								
		23:16								
		15:8	Reserved	Reserved	AC	PORT	FREQM	EIC	RTC	WDT
		7:0	GCLK	SUPC	OSC32KCTRL	OSCCTRL	RSTC	MCLK	PM	PAC
0x18	INTFLAGB	31:24								
		23:16								
		15:8								
		7:0			USB	HMATRIXHS	DMAC	NVMCTRL	DSU	IDAU
0x1C	INTFLAGC	31:24								
		23:16			TRAM	OPAMP	I2S	CCL	TRNG	PTC
		15:8	DAC	ADC	TCC3	TCC2	TCC1	TCC0	TC2	TC1
		7:0	TC0	SERCOM5	SERCOM4	SERCOM3	SERCOM2	SERCOM1	SERCOM0	EVSYS
0x20 ... 0x33	Reserved									
0x34	STATUSA	31:24								
		23:16								
		15:8	Reserved	Reserved	AC	PORT	FREQM	EIC	RTC	WDT
		7:0	GCLK	SUPC	OSC32KCTRL	OSCCTRL	RSTC	MCLK	PM	PAC
0x38	STATUSB	31:24								
		23:16								
		15:8								
		7:0			USB	HMATRIXHS	DMAC	NVMCTRL	DSU	IDAU

.....continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x3C	STATUSC	31:24								
		23:16			TRAM	OPAMP	I2S	CCL	TRNG	PTC
		15:8	DAC	ADC	TCC3	TCC2	TCC1	TCC0	TC2	TC1
		7:0	TC0	SERCOM5	SERCOM4	SERCOM3	SERCOM2	SERCOM1	SERCOM0	EVSYS
0x40 ... 0x53	Reserved									
0x54	NONSECA	31:24								
		23:16								
		15:8	Reserved	Reserved	AC	PORT	FREQM	EIC	RTC	WDT
		7:0	GCLK	SUPC	OSC32KCTRL	OSCCTRL	RSTC	MCLK	PM	PAC
0x58	NONSECB	31:24								
		23:16								
		15:8								
		7:0			USB	HMATRIXHS	DMAC	NVMCTRL	DSU	IDAU
0x5C	NONSECC	31:24								
		23:16			TRAM	OPAMP	I2S	CCL	TRNG	PTC
		15:8	DAC	ADC	TCC3	TCC2	TCC1	TCC0	TC2	TC1
		7:0	TC0	SERCOM5	SERCOM4	SERCOM3	SERCOM2	SERCOM1	SERCOM0	EVSYS
0x60 ... 0x73	Reserved									
0x74	SECLOCKA	31:24								
		23:16								
		15:8	Reserved	Reserved	AC	PORT	FREQM	EIC	RTC	WDT
		7:0	GCLK	SUPC	OSC32KCTRL	OSCCTRL	RSTC	MCLK	PM	PAC
0x78	SECLOCKB	31:24								
		23:16								
		15:8								
		7:0			USB	HMATRIXHS	DMAC	NVMCTRL	DSU	IDAU
0x7C	SECLOCKC	31:24								
		23:16			TRAM	OPAMP	I2S	CCL	TRNG	PTC
		15:8	DAC	ADC	TCC3	TCC2	TCC1	TCC0	TC2	TC1
		7:0	TC0	SERCOM5	SERCOM4	SERCOM3	SERCOM2	SERCOM1	SERCOM0	EVSYS

15.6.1 Write Control

Name: WRCTRL
Offset: 0x00
Reset: 0x00020021
Property: Mix-Secure

Important: if BOCOR.SECCFGLOCK == 0 after exiting the Boot ROM, the secure software code of the Flash BOOT region, before passing control on to the secure software code of the Flash APPLICATION region, **MUST** lock the current security attribution of each peripheral using SECLOCK command.

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/RW	RW/RW/RW	RW/RW/RW	RW/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/RW	RW/RW/RW	RW/RW/RW	RW/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/RW	RW/RW/RW	RW/RW/RW	RW/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	CLR	Clear the peripheral write control
0x2	SET	Set the peripheral write control
0x3	SETLCK	Set and lock the peripheral write control until the next hardware reset
0x4	SETSEC	PIC32CM LS00/LS60 only: Set the peripheral as Secure (the NONSEC bit is set to 0) This command is not supported if BOCOR.SECCFGLOCK == 1 after exiting the Boot ROM
0x5	SETNONSEC	PIC32CM LS00/LS60 only: Set the peripheral as Non-Secure (the NONSEC bit is set to 1) This command is not supported if BOCOR.SECCFGLOCK == 1 after exiting the Boot ROM
0x6	SECLOCK	PIC32CM LS00/LS60 only: Lock the current peripheral security attribution until the next reset This command is not supported if BOCOR.SECCFGLOCK == 1 after exiting the Boot ROM

Bits 15:0 – PERID[15:0] Peripheral Identifier

The PERID represents the peripheral whose control is changed using the WRCTRL.KEY:

Table 15-2. Peripheral Identifier

Peripheral	Identifier
APBA Peripherals	
PAC	0
PM	1
MCLK	2
RSTC	3
OSCCTRL	4
OSC32KCTRL	5
SUPC	6
GCLK	7
WDT	8
RTC	9
EIC	10
FREQM	11
PORT	12
AC	13
APBB Peripherals	
IDAU	32
DSU	33
NVMCTRL	34
DMAC	35
HMATRIXHS	36
USB	37
APBC Peripherals	
EVSYS	64
SERCOM0	65
SERCOM1	66
SERCOM2	67
SERCOM3	68
SERCOM4	69
SERCOM5	70
TC0	71
TC1	72
TC2	73
TCC0	74
TCC1	75
TCC2	76
TCC3	77
ADC	78
DAC	79
PTC	80
TRNG	81
CCL	82
I2S	83
OPAMP	84
TRAM	85

15.6.2 Event Control

Name: EVCTRL
Offset: 0x04
Reset: 0x00
Property: PAC Write-Protection, Secure

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 will be 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.

15.6.3 Interrupt Enable Clear

Name: INTENCLR
Offset: 0x08
Reset: 0x00
Property: PAC Write-Protection, Secure

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
								ERR
Access								RW/-/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 will be 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 will clear 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.

15.6.4 Interrupt Enable Set

Name: INTENSET
Offset: 0x09
Reset: 0x00
Property: PAC Write-Protection, Secure

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
								ERR
Access								RW/-/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 will be 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 will set 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.

15.6.5 AHB Client Bus Interrupt Flag Status and Clear

Name: INTFLAGAHB
Offset: 0x10
Reset: 0x000000
Property: Secure

This flag is cleared by writing a '1' to the flag.

This flag is set when an access error is detected by the CLIENT n, and will generate an interrupt request if INTENSET.ERR is '1'.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the corresponding INTFLAGAHB 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
	BROM	HSRAMDSU	HSRAMDMAC	HSRAMCPU	APBC	APBB	APBA	FLASH
Access	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW
Reset	0	0	0	0	0	0	0	0

Bit 7 – BROM Interrupt Flag for Boot ROM

Bit 6 – HSRAMDSU Interrupt Flag for CLIENT SRAM Port 2 - DSU Access

Bit 5 – HSRAMDMAC Interrupt Flag for CLIENT SRAM Port 1 - DMAC Access

Bit 4 – HSRAMCPU Interrupt Flag for CLIENT SRAM Port 0 - CPU Access

Bit 3 – APBC Interrupt Flag for CLIENT AHB-APB Bridge C

Bit 2 – APBB Interrupt Flag for CLIENT AHB-APB Bridge B

Bit 1 – APBA Interrupt Flag for CLIENT AHB-APB Bridge A

Bit 0 – FLASH Interrupt Flag for CLIENT FLASH

15.6.6 Peripheral Interrupt Flag Status and Clear A

Name: INTFLAGA
Offset: 0x14
Reset: 0x000000
Property: Secure

This flag is cleared by writing a one to the flag.

This flag is set when a Peripheral Access Error occurs while accessing the peripheral associated with the respective INTFLAGA bit, and will generate an interrupt request if INTENSET.ERR is one.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the corresponding INTFLAGA 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
	Reserved	Reserved	AC	PORT	FREQM	EIC	RTC	WDT
Access	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW	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
	GCLK	SUPC	OSC32KCTRL	OSCCTRL	RSTC	MCLK	PM	PAC
Access	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW
Reset	0	0	0	0	0	0	0	0

Bit 15 – Reserved

Bit 14 – Reserved

Bit 13 – AC Interrupt Flag for AC

Bit 12 – PORT Interrupt Flag for PORT

Bit 11 – FREQM Interrupt Flag for FREQM

Bit 10 – EIC Interrupt Flag for EIC

Bit 9 – RTC Interrupt Flag for RTC

Bit 8 – WDT Interrupt Flag for WDT

Bit 7 – GCLK Interrupt Flag for GCLK

Bit 6 – SUPC Interrupt Flag for SUPC

Bit 5 – OSC32KCTRL Interrupt Flag for OSC32KCTRL

Bit 4 – OSCCTRL Interrupt Flag for OSCCTRL

Bit 3 – RSTC Interrupt Flag for RSTC

Bit 2 – MCLK Interrupt Flag for MCLK

Bit 1 – PM Interrupt Flag for PM

Bit 0 – PAC Interrupt Flag for PAC

15.6.7 Peripheral Interrupt Flag Status and Clear B

Name: INTFLAGB
Offset: 0x18
Reset: 0x000000
Property: Secure

This flag is cleared by writing a '1' to the flag.

This flag is set when a Peripheral Access Error occurs while accessing the peripheral associated with the respective INTFLAGB bit, and will generate an interrupt request if INTENSET.ERR is '1'.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the corresponding INTFLAGB 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
			USB	HMATRIXHS	DMAC	NVMCTRL	DSU	IDAU
Access			RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW
Reset			0	0	0	0	0	0

Bit 5 – USB Interrupt Flag for USB

Bit 4 – HMATRIXHS Interrupt Flag for HMATRIXHS

Bit 3 – DMAC Interrupt Flag for DMAC

Bit 2 – NVMCTRL Interrupt Flag for NVMCTRL

Bit 1 – DSU Interrupt Flag for DSU

Bit 0 – IDAU Interrupt Flag for IDAU

Note: This bit field is only available for **PIC32CM LS00/LS60** and has no effect for **PIC32CM LE00**.

15.6.8 Peripheral Interrupt Flag Status and Clear C

Name: INTFLAGC
Offset: 0x1C
Reset: 0x000000
Property: Secure

This flag is cleared by writing a one to the flag.

This flag is set when a Peripheral Access Error occurs while accessing the peripheral associated with the respective INTFLAGC bit, and will generate an interrupt request if INTENSET.ERR is one.

Writing a zero to this bit has no effect.

Writing a one to this bit will clear the corresponding INTFLAGC interrupt flag.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
			TRAM	OPAMP	I2S	CCL	TRNG	PTC
Access			RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW
Reset			0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	DAC	ADC	TCC3	TCC2	TCC1	TCC0	TC2	TC1
Access	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW	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
	TC0	SERCOM5	SERCOM4	SERCOM3	SERCOM2	SERCOM1	SERCOM0	EVSYS
Access	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW
Reset	0	0	0	0	0	0	0	0

Bit 21 – TRAM Interrupt Flag for TRAM

Bit 20 – OPAMP Interrupt Flag for OPAMP

Bit 19 – I2S Interrupt Flag for I2S

Bit 18 – CCL Interrupt Flag for CCL

Bit 17 – TRNG Interrupt Flag for TRNG

Bit 16 – PTC Interrupt Flag for PTC

Bit 15 – DAC Interrupt Flag for DAC

Bit 14 – ADC Interrupt Flag for ADC

Bits 10, 11, 12, 13 – TCC Interrupt Flag for TCCn [n = 3..0]

Bits 7, 8, 9 – TC Interrupt Flag for TCn [n = 2..0]

Bits 1, 2, 3, 4, 5, 6 – SERCOM Interrupt Flag for SERCOMn [n = 5..0]

Bit 0 – EVSYS Interrupt Flag for EVSYS

15.6.9 Peripheral Write Protection Status A

Name: STATUSA
Offset: 0x34
Reset: 0x0000C000
Property: Mix-Secure

Reading STATUSA register returns peripheral write protection status:

Value	Description
0	Peripheral is not write protected.
1	Peripheral is write protected.

Important: For **PIC32CM LS00/LS60 Non-Secure** accesses, read accesses (R*) are allowed only if the peripheral security attribution for the corresponding peripheral is set as Non-Secured in the NONSECx 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
	Reserved	Reserved	AC	PORT	FREQM	EIC	RTC	WDT
Access	R/R*/R	R/R*/R	R/R*/R	R/R*/R	R/R*/R	R/R*/R	R/R*/R	R/R*/R
Reset	1	1	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	GCLK	SUPC	OSC32KCTRL	OSCCTRL	RSTC	MCLK	PM	PAC
Access	R/R*/R	R/R*/R	R/R*/R	R/R*/R	R/R*/R	R/R*/R	R/R*/R	R/R*/R
Reset	0	0	0	0	0	0	0	0

Bit 15 – Reserved

Bit 14 – Reserved

Bit 13 – AC Peripheral AC Write Protection Status

Bit 12 – PORT Peripheral PORT Write Protection Status

Bit 11 – FREQM Peripheral FREQM Write Protection Status

Bit 10 – EIC Peripheral EIC Write Protection Status

Bit 9 – RTC Peripheral RTC Write Protection Status

Bit 8 – WDT Peripheral WDT Write Protection Status

Bit 7 – GCLK Peripheral GCLK Write Protection Status

Bit 6 – SUPC Peripheral SUPC Write Protection Status

Bit 5 – OSC32KCTRL Peripheral OSC32KCTRL Write Protection Status

Bit 4 – OSCCTRL Peripheral OSCCTRL Write Protection Status

Bit 3 – RSTC Peripheral RSTC Write Protection Status

Bit 2 – MCLK Peripheral MCLK Write Protection Status

Bit 1 – PM Peripheral PM Write Protection Status

Bit 0 – PAC Peripheral PAC Write Protection Status

15.6.10 Peripheral Write Protection Status B

Name: STATUSB
Offset: 0x38
Reset: 0x00000002
Property: Mix-Secure

Reading the STATUSB register returns the peripheral write protection status:

Value	Description
0	Peripheral is not write protected.
1	Peripheral is write protected.

Important: For **PIC32CM LS00/LS60 Non-Secure** accesses, read accesses (R*) are allowed only if the peripheral security attribution for the corresponding peripheral is set as Non-Secured in the NONSECx 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
			USB	HMATRIXHS	DMAC	NVMCTRL	DSU	IDAU
Access			R/R*/R	R/R*/R	R/R*/R	R/R*/R	R/R*/R	R/R*/R
Reset			0	0	0	0	1	0

Bit 5 – USB Peripheral USB Write Protection Status

Bit 4 – HMATRIXHS Peripheral HMATRIXHS Write Protection Status

Bit 3 – DMAC Peripheral DMAC Write Protection Status

Bit 2 – NVMCTRL Peripheral NVMCTRL Write Protection Status

Bit 1 – DSU Peripheral DSU Write Protection Status

Bit 0 – IDAU Peripheral IDAU Write Protection Status
Note: This bit field is only available for **PIC32CM LS00/LS60** and has no effect for **PIC32CM LE00**.

15.6.11 Peripheral Write Protection Status C

Name: STATUSC
Offset: 0x3C
Reset: 0x000000
Property: Mix-Secure

Reading the STATUSC register returns the peripheral write protection status:

Value	Description
0	Peripheral is not write protected.
1	Peripheral is write protected.

Important: For **PIC32CM LS00/LS60 Non-Secure** accesses, read accesses (R*) are allowed only if the peripheral security attribution for the corresponding peripheral is set as Non-Secured in the NONSECx register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
			TRAM	OPAMP	I2S	CCL	TRNG	PTC
Access			R/R*/R	R/R*/R	R/R*/R	R/R*/R	R/R*/R	R/R*/R
Reset			0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	DAC	ADC	TCC3	TCC2	TCC1	TCC0	TC2	TC1
Access	R/R*/R	R/R*/R	R/R*/R	R/R*/R	R/R*/R	R/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
	TC0	SERCOM5	SERCOM4	SERCOM3	SERCOM2	SERCOM1	SERCOM0	EVSYS
Access	R/R*/R	R/R*/R	R/R*/R	R/R*/R	R/R*/R	R/R*/R	R/R*/R	R/R*/R
Reset	0	0	0	0	0	0	0	0

Bit 21 – TRAM Peripheral TRAM Write Protection Status

Bit 20 – OPAMP Peripheral OPAMP Write Protection Status

Bit 19 – I2S Peripheral I2S Write Protection Status

Bit 18 – CCL Peripheral CCL Write Protection Status

Bit 17 – TRNG Peripheral TRNG Write Protection Status

Bit 16 – PTC Peripheral PTC Write Protection Status

Bit 15 – DAC Peripheral DAC Write Protection Status

Bit 14 – ADC Peripheral ADC Write Protection Status

Bits 10, 11, 12, 13 – TCC Peripheral TCn Write Protection Status [n = 3..0]

Bits 7, 8, 9 – TC Peripheral TCn Write Protection Status [n = 2..0]

Bits 1, 2, 3, 4, 5, 6 – SERCOM Peripheral SERCOMn Write Protection Status [n = 5..0]

Bit 0 – EVSYS Peripheral EVSYS Write Protection Status

15.6.12 Peripheral Non-Secure Status - Bridge A

Name: NONSECA
Offset: 0x54
Reset: x initially determined from NVM User Row after reset
Property: Write-Secure

Important: This register is only available for **PIC32CM LS00/LS60** and has no effect for **PIC32CM LE00**.

This register is loaded from UROW at boot.

Reading NONSECA register returns peripheral security attribution status:

Value	Description
0	Peripheral is secured.
1	Peripheral is non-secured.

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
	Reserved	Reserved	AC	PORT	FREQM	EIC	RTC	WDT
Access	R/R/R	R/R/R	R/R/R	R/R/R	R/R/R	R/R/R	R/R/R	R/R/R
Reset	x	x	x	x	x	x	x	x

Bit	7	6	5	4	3	2	1	0
	GCLK	SUPC	OSC32KCTRL	OSCCTRL	RSTC	MCLK	PM	PAC
Access	R/R/R	R/R/R	R/R/R	R/R/R	R/R/R	R/R/R	R/R/R	R/R/R
Reset	x	x	x	x	x	x	x	x

Bit 15 – Reserved

Bit 14 – Reserved

Bit 13 – AC Peripheral AC Non-Secure

Bit 12 – PORT Peripheral PORT Non-Secure

Bit 11 – FREQM Peripheral FREQM Non-Secure

Bit 10 – EIC Peripheral EIC Non-Secure

Bit 9 – RTC Peripheral RTC Non-Secure

Bit 8 – WDT Peripheral WDT Non-Secure

Bit 7 – GCLK Peripheral GCLK Non-Secure

Bit 6 – SUPC Peripheral SUPC Non-Secure

Bit 5 – OSC32KCTRL Peripheral OSC32KCTRL Non-Secure

Bit 4 – OSCCTRL Peripheral OSCCTRL Non-Secure

Bit 3 – RSTC Peripheral RSTC Non-Secure

Bit 2 – MCLK Peripheral MCLK Non-Secure

Bit 1 – PM Peripheral PM Non-Secure

Bit 0 – PAC Peripheral PAC Non-Secure
The PAC Peripheral is always secured if BOCOR.SECCFGLOCK == 1 after exiting the Boot ROM (NONSECA.PAC=0).

15.6.13 Peripheral Non-Secure Status - Bridge B

Name: NONSECB
Offset: 0x58
Reset: x initially determined from NVM User Row after reset
Property: Write-Secure

Important: This register is only available for **PIC32CM LS00/LS60** and has no effect for **PIC32CM LE00**.

This register is loaded from UROW at boot.

Reading NONSECB register returns peripheral security attribution status:

Value	Description
0	Peripheral is secured.
1	Peripheral is non-secured.

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
			USB	HMATRIXHS	DMAC	NVMCTRL	DSU	IDAU
Access			R/R/R	R/R/R	R/R/R	R/R/R	R/R/R	R/R/R
Reset			x	x	x	x	1	0

Bit 5 – USB Peripheral USB Non-Secure

Bit 4 – HMATRIXHS Peripheral HMATRIXHS Non-Secure

Bit 3 – DMAC Peripheral DMAC Non-Secure

Bit 2 – NVMCTRL Peripheral NVMCTRL Non-Secure
The NVMCTRL Peripheral is always secured if BOCOR.SECCFGLOCK == 1 after exiting the Boot ROM (NONSECB.NVMCTRL=0).

Bit 1 – DSU Peripheral DSU Non-Secure
The DSU Peripheral is always non-secured (NONSECB.DSU == 1).

Bit 0 – IDAU Peripheral IDAU Non-Secure
The IDAU Peripheral is always secured (NONSECB.IDAU == 0).

15.6.14 Peripheral Non-Secure Status - Bridge C

Name: NONSECC
Offset: 0x5C
Reset: x initially determined from NVM User Row after reset
Property: Write-Secure

Important: This register is only available for **PIC32CM LS00/LS60** and has no effect for **PIC32CM LE00**.

This register is loaded from UROW at boot.

Reading NONSECC register returns peripheral Security Attribution status:

Value	Description
0	Peripheral is secured.
1	Peripheral is non-secured.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
			TRAM	OPAMP	I2S	CCL	TRNG	PTC
Access			R/R/R	R/R/R	R/R/R	R/R/R	R/R/R	R/R/R
Reset			x	x	x	x	x	x

Bit	15	14	13	12	11	10	9	8
	DAC	ADC	TCC3	TCC2	TCC1	TCC0	TC2	TC1
Access	R/R/R	R/R/R	R/R/R	R/R/R	R/R/R	R/R/R	R/R/R	R/R/R
Reset	x	x	x	x	x	x	x	x

Bit	7	6	5	4	3	2	1	0
	TC0	SERCOM5	SERCOM4	SERCOM3	SERCOM2	SERCOM1	SERCOM0	EVSYS
Access	R/R/R	R/R/R	R/R/R	R/R/R	R/R/R	R/R/R	R/R/R	R/R/R
Reset	x	x	x	x	x	x	x	x

Bit 21 – TRAM Peripheral TRAM Non-Secure

Bit 20 – OPAMP Peripheral OPAMP Non-Secure

Bit 19 – I2S Peripheral I2S Non-Secure

Bit 18 – CCL Peripheral CCL Non-Secure

Bit 17 – TRNG Peripheral TRNG Non-Secure

Bit 16 – PTC Peripheral PTC Non-Secure

Bit 15 – DAC Peripheral DAC Non-Secure

Bit 14 – ADC Peripheral ADC Non-Secure

Bits 10, 11, 12, 13 – TCC Peripheral TCCn Non-Secure [n = 3..0]

Bits 7, 8, 9 – TC Peripheral TCn Non-Secure [n = 2..0]

Bits 1, 2, 3, 4, 5, 6 – SERCOM Peripheral SERCOMn Non-Secure [n = 5..0]

Bit 0 – EVSYS Peripheral EVSYS Non-Secure

15.6.15 Peripheral Security Lock Status - Bridge A

Name: SECLOCKA
Offset: 0x74
Reset: 0x00000000 if BOCOR.SECCFGLOCK == 0 / 0x00000001 if BOCOR.SECCFGLOCK == 1
Property: Write-Secure

Important:
- This register is only available for **PIC32CM LS00/LS60** and has no effect for **PIC32CM LE00**.
- if BOCOR.SECCFGLOCK == 0 after exiting the Boot ROM, the secure software code of the Flash BOOT region, before passing control on to the secure software code of the Flash APPLICATION region, **MUST** lock the current security attribution of each peripheral using WRCTRL.SECLOCK command.

Reading SECLOCK register returns peripheral security lock status:

Value	Description
0	Peripheral security attribution is not locked.
1	Peripheral security attribution is locked.

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
	Reserved	Reserved	AC	PORT	FREQM	EIC	RTC	WDT
Access	R/R/R	R/R/R	R/R/R	R/R/R	R/R/R	R/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
	GCLK	SUPC	OSC32KCTRL	OSCCTRL	RSTC	MCLK	PM	PAC
Access	R/R/R	R/R/R	R/R/R	R/R/R	R/R/R	R/R/R	R/R/R	R/R/R
Reset	0	0	0	0	0	0	0	x

Bit 15 – Reserved

Bit 14 – Reserved

Bit 13 – AC Peripheral AC Security Lock

Bit 12 – PORT Peripheral PORT Security Lock

Bit 11 – FREQM Peripheral FREQM Security Lock

Bit 10 – EIC Peripheral EIC Security Lock

Bit 9 – RTC Peripheral RTC Security Lock

Bit 8 – WDT Peripheral WDT Security Lock

Bit 7 – GCLK Peripheral GCLK Security Lock

Bit 6 – SUPC Peripheral SUPC Security Lock

Bit 5 – OSC32KCTRL Peripheral OSC32KCTRL Security Lock

Bit 4 – OSCCTRL Peripheral OSCCTRL Security Lock

Bit 3 – RSTC Peripheral RSTC Security Lock

Bit 2 – MCLK Peripheral MCLK Security Lock

Bit 1 – PM Peripheral PM Security Lock

Bit 0 – PAC Peripheral PAC Security Lock

15.6.16 Peripheral Security Lock Status - Bridge B

Name: SECLOCKB
Offset: 0x78
Reset: 0x00000003 if BOCOR.SECCFGLOCK == 0 / 0x00000007 if BOCOR.SECCFGLOCK == 1
Property: Write-Secure

Important:
- This register is only available for **PIC32CM LS00/LS60** and has no effect for **PIC32CM LE00**.
- if BOCOR.SECCFGLOCK == 0 after exiting the Boot ROM, the secure software code of the Flash BOOT region, before passing control on to the secure software code of the Flash APPLICATION region, **MUST** lock the current security attribution of each peripheral using WRCTRL.SECLOCK command.

Reading SECLOCK register returns peripheral security lock status:

Value	Description
0	Peripheral security attribution is not locked.
1	Peripheral security attribution is locked.

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
			USB	HMATRIXHS	DMAC	NVMCTRL	DSU	IDAU
Access			R/R/R	R/R/R	R/R/R	R/R/R	R/R/R	R/R/R
Reset			0	0	0	x	1	1

Bit 5 – USB Peripheral USB Security Lock

Bit 4 – HMATRIXHS Peripheral HMATRIXHS Security Lock

Bit 3 – DMAC Peripheral DMAC Security Lock

Bit 2 – NVMCTRL Peripheral NVMCTRL Security Lock

Bit 1 – DSU Peripheral DSU Security Lock

Bit 0 – IDAU Peripheral IDAU Security Lock

15.6.17 Peripheral Security Lock Status - Bridge C

Name: SECLOCKC
Offset: 0x7C
Reset: 0x00000000
Property: Write-Secure

Important:
- This register is only available for **PIC32CM LS00/LS60** and has no effect for **PIC32CM LE00**.
- if BOCOR.SECCFGLOCK == 0 after exiting the Boot ROM, the secure software code of the Flash BOOT region, before passing control on to the secure software code of the Flash APPLICATION region, **MUST** lock the current security attribution of each peripheral using WRCTRL.SECLOCK command.

Reading SECLOCK register returns peripheral security lock status:

Value	Description
0	Peripheral security attribution is not locked.
1	Peripheral security attribution is locked.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
			TRAM	OPAMP	I2S	CCL	TRNG	PTC
Access			R/R/R	R/R/R	R/R/R	R/R/R	R/R/R	R/R/R
Reset			0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	DAC	ADC	TCC3	TCC2	TCC1	TCC0	TC2	TC1
Access	R/R/R	R/R/R	R/R/R	R/R/R	R/R/R	R/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
	TC0	SERCOM5	SERCOM4	SERCOM3	SERCOM2	SERCOM1	SERCOM0	EVSYS
Access	R/R/R	R/R/R	R/R/R	R/R/R	R/R/R	R/R/R	R/R/R	R/R/R
Reset	0	0	0	0	0	0	0	0

Bit 21 – TRAM Peripheral TRAM Security Lock

Bit 20 – OPAMP Peripheral OPAMP Security Lock

Bit 19 – I2S Peripheral I2S Security Lock

Bit 18 – CCL Peripheral CCL Security Lock

Bit 17 – TRNG Peripheral TRNG Security Lock

Bit 16 – PTC Peripheral PTC Security Lock

Bit 15 – DAC Peripheral DAC Security Lock

Bit 14 – ADC Peripheral ADC Security Lock

Bits 10, 11, 12, 13 – TCC Peripheral TCn Security Lock [n = 3..0]

Bits 7, 8, 9 – TC Peripheral TCn Security Lock [n = 2..0]

Bits 1, 2, 3, 4, 5, 6 – SERCOM Peripheral SERCOMn Security Lock [n = 5..0]

Bit 0 – EVSYS Peripheral EVSYS Security Lock

16. Device Service Unit (DSU)

16.1 Overview

The Device Service Unit (DSU) provides a means to detect debugger probes. This 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 and 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, for example, as required for IEC60730 Class B compliance The DSU can be accessed simultaneously by a debugger and the CPU, as it is connected on the High-Speed Bus Matrix. It implements communication channels between the device and external tools which can be used at boot time to make use of Boot ROM services. For security reasons, some of the DSU features will be limited or unavailable when the Debug Access Level (DAL) is less than 0x2.

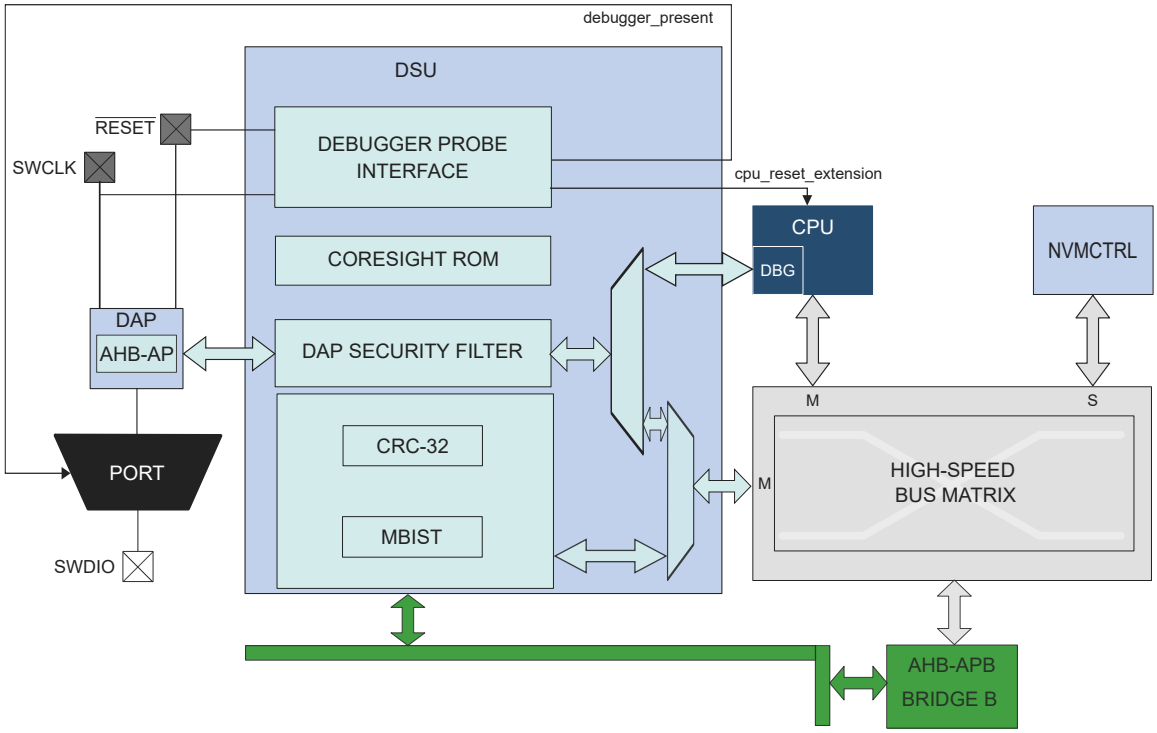
Refer to 29. Non-volatile Memory Controller (NVMCTRL) for further information.

16.2 Features

- CPU reset extension
- Debugger probe detection (Cold- and Hot-Plugging)
- 32-bit cyclic redundancy check (CRC32) of any memory accessible through the bus matrix
- Arm® CoreSight™ compliant device identification
- Two debug communications channels
- Two Boot communications channels
- Debug access port security filter
- Onboard memory built-in self-test (MBIST)

16.3 Block Diagram

Figure 16-1. DSU Block Diagram



16.4 Signal Description

The DSU uses the following three signals to function.

Signal Name	Type	Description
$\overline{\text{RESET}}$	Digital Input	External reset
SWCLK	Digital Input	SW clock
SWDIO	Digital I/O	SW bidirectional data pin

Note: The SWCLK pin is by default assigned to a SWCLK Peripheral Function from pinout multiplexing tables to allow debugger probe detection. A debugger probe detection (cold-plugging or hot-plugging) will automatically switch the SWDIO I/O pin to the SWDIO function, as long as the SWLCK peripheral function is selected. If the SWCLK pin function is changed in the PORT, the Hot-Plugging feature is disabled until a power-reset or an external reset is performed.

16.5 Peripheral Dependencies

Table 16-1. DSU Configuration Summary

Peripheral name	Base address	MCLK AHB/APB Clocks	PAC Peripheral Identifier Index (PAC.WRCTRL)	DMA Trigger Source Index (DMAC.CHCTRLB)	Power Domain (PM.STDBYCFG)
DSU	0x41002000	CLK_DSU_AHB CLK_DSU_APB	33	2-3: DCC0-1	PDSW

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
- Boot Communication Channels

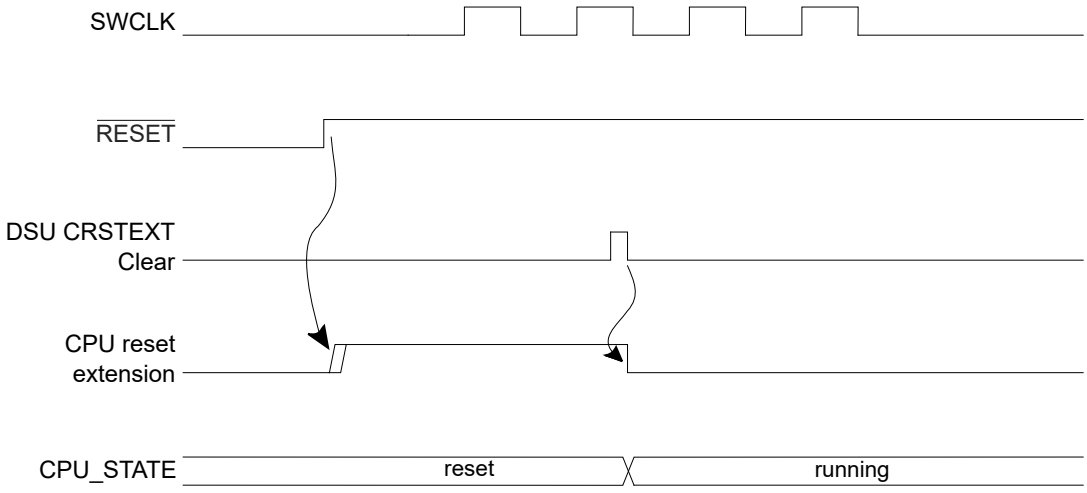
For more details on the Arm debug components, refer to the Arm Debug Interface v5 Architecture Specification.

16.6.2 CPU Reset Extension

CPU reset extension refers to the extension of the reset phase of the CPU core after the external reset is released. This ensures that the CPU is not executing code at startup while a debugger connects to the system. It is detected on a $\overline{\text{RESET}}$ release event when SWCLK is low. At startup, SWCLK is internally pulled up to avoid false detection of a debugger if SWCLK 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 zero. Writing a '0' to STATUSA.CRSTEXT has no effect. Releasing the "CPU reset extension" is possible for all DAL levels. The CPU then executes the Boot ROM that offers basic failure analysis services and security checks. It is not possible to access the bus system until the Boot ROM has performed these security checks.

Note: Refer to 13. Boot ROM for more information.

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 $\overline{\text{RESET}}$ are asserted. Hot-Plugging is active when a SWCLK falling edge is detected. The SWCLK pad is multiplexed with other functions and the user must ensure that its default function is assigned to the debug system. If the SWCLK function is changed, the Hot-Plugging feature is disabled until a power-reset or external reset occurs. 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. Once detected, the Debugger Present bit of the Status B register (STATUSB.DBGPRES) is set. For security reasons, Hot-Plugging is not available when DAL equals to 0x0.

This detection requires that pads are correctly powered. Thus, at cold startup, this detection cannot be done until POR is released. If DAL equals 0x0, Cold-Plugging is the only way to detect a debugger probe, and so the external reset timing must be longer than the POR timing. If external reset is de-asserted before POR release, the user must retry the procedure above until it gets connected to the device.

16.6.4 Boot Communication Channels

Boot Communication Channels allow communication between a debug adapter and the CPU executing the Boot ROM at startup. The Boot ROM implements system level commands. Refer to 13. Boot ROM for more information.

16.7 Programming

Programming the Flash or RAM memories is only possible when the debugger access level is sufficient to access the desired resource:

If DAL is equal to:

- 0x2: debugger can access secured and non-secure areas
- 0x1 (**PIC32CM LS00/LS60 only**): debugger can access only non-secure areas, refer to Table 16-5.
- 0x0: debugger can only access the DSU external address space making it possible to communicate with the Boot ROM after reset.

A typical programming procedure when DAL=0x2 is as follows:

1. At power up, $\overline{\text{RESET}}$ is driven low by a debugger. The on-chip regulator holds the system in a POR state until the input supply is above the POR threshold. The system continues to be held in this static state until the internally regulated supplies have reached a safe operating state.
2. The Power Manager (PM) starts, clocks are switched to the slow clock (Core Clock, System Clock, Flash Clock and any Bus Clocks that do not have clock gate control). Internal resets are maintained due to the external reset.

3. The debugger maintains a low level on SWCLK. $\overline{\text{RESET}}$ is released, resulting in a debugger Cold-Plugging procedure.
4. The debugger generates a clock signal on the SWCLK pin, the Debug Access Port (DAP) receives a clock.
5. The CPU executes the Boot ROM.
6. It is recommended to issue a Chip-Erase (supported by the Boot ROM) to ensure that the Flash is fully erased prior to programming.
7. If the operation issued above was accepted and has completed successfully then DAL equals 0x2 thus programming is available through the AHB-AP.
8. After the operation is completed, the chip can be restarted either by asserting $\overline{\text{RESET}}$, toggling power, or sending a command to the Boot ROM to jump to the NVM code. Make sure that the SWCLK pin is high when releasing $\overline{\text{RESET}}$ to prevent entering again the cold-plugging procedure with the Boot ROM stalling the CPU.

16.8 Security Enforcement

Security enforcement aims at protecting intellectual property, which includes:

- Restricts access to internal memories from external tools depending on the debugger access level.
- Restricts access to a portion of the DSU address space from non-secure AHB hosts depending on the debugger access level.

The DAL setting can be locked or reverted using Boot ROM commands depending on the Boot ROM user configuration. When DAL is equal to 0x0, read/write accesses using the AHB-AP are limited to the DSU external address range and DSU commands are restricted. When issuing a Boot ROM Chip-Erase, sensitive information is erased from volatile memory and Flash. Refer to 13. Boot ROM more information about the Boot ROM features.

The DSU implements a security filter that monitors the AHB transactions generated by the ARM AHB-AP inside the DAP. If DAL=0x0, 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 (refer to the "ARM Debug Interface v5 Architecture Specification", which is available for download at www.arm.com).

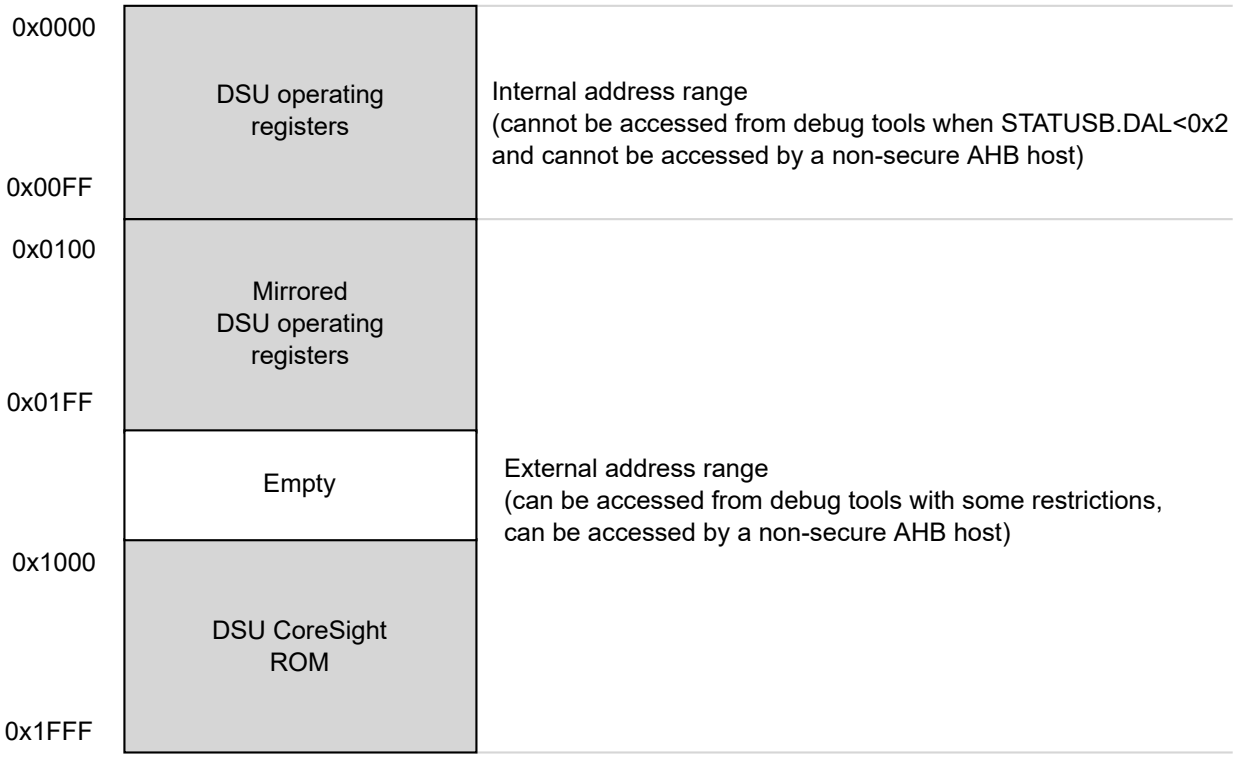
For security reasons, DSU features have limitations when used from a debug adapter. To differentiate external accesses from internal ones, the first 0x100 bytes of the DSU register map have been replicated at offset 0x100:

- The first 0x100 bytes form the internal address range
- The next 0x1F00 bytes form the external address range

When the device is protected, the DAP can only issue MEM-AP accesses in the DSU address range limited to the 0x100- 0x2000 offset range.

The DSU operating registers are located in the 0x00-0xFF area and mirrored to 0x100-0x1FF to differentiate accesses coming from a debugger and the CPU. If the device is protected and an access is issued in the region 0x100-0x1FF, it is subject to security restrictions. For more information, refer to the *DAP Access Rights Depending on DAL* table.

Figure 16-4. APB Memory Mapping



The DSU filters-out DAP transactions depending on the DAL setting and routes DAP transactions:

- In the PPB or IOBUS space to the CPU debug port
- Outside the PPB space and outside the IOBUS space to the DSU host port

Table 16-2. DAP Access Rights Depending on DAL

DAP access to	PIC32CM LS00/LS60			PIC32CM LE00	
	DAL=0	DAL=1	DAL=2	DAL=0	DAL=2
PPB or IOBUS	No	Yes (see **Note 1**)	Yes	No	Yes
DSU internal address space	No	No (see **Note 2**)	Yes	No	Yes
DSU external address space	Yes	Yes	Yes	Yes	Yes
Other secure areas	No	No	Yes	No	Yes
Other non-secure areas	No	Yes	Yes	No	Yes

Notes:

1. Refer to ARMv8-M debug documentation for detailed information on PPB and IOBUS access restrictions.
2. When DAL=1 DAP transfers are always non-secure. The DSU internal address space can only be accessed by secure hosts.

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

Table 16-3. Feature Availability Under Protection

Features	Availability when DAL equals to		
	0x0	0x1 (PIC32CM LS00/LS60 only)	0x2
CPU Reset Extension	Yes	Yes	Yes
Clear CPU Reset extension	Yes	Yes	Yes
Debugger Cold-Plugging	Yes	Yes	Yes
Debugger Hot-Plugging	No	Yes	Yes

16.9 Device Identification

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

16.9.1 CoreSight Identification

A system-level Arm® CoreSight™ ROM table is present in the device to identify the vendor and the chip identification method. Its address is provided in the MEM-AP BASE register inside the Arm Debug Access Port. The CoreSight ROM implements a 64-bit conceptual ID composed as follows from the PID0 to PID7 CoreSight ROM Table registers:

Figure 16-5. Conceptual 64-bit Peripheral ID

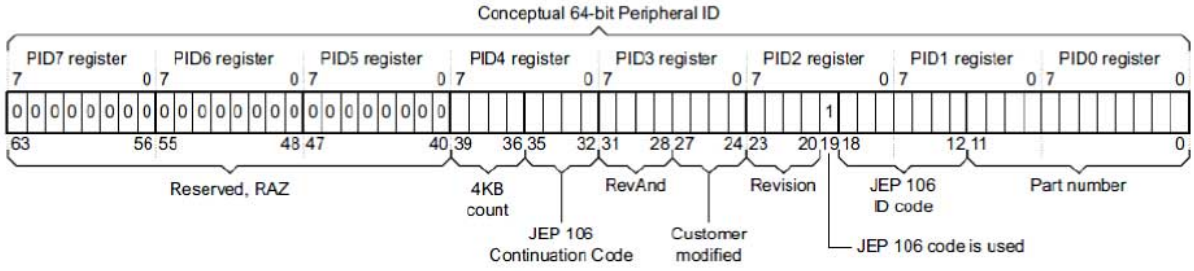


Table 16-4. Conceptual 64-Bit Peripheral ID Bit Descriptions

Field	Size	Description	Location
JEP-106 CC code	4	Continuation code: 0x0	PID4
JEP-106 ID code	7	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 information, refer to the Arm Debug Interface Version 5 Architecture Specification.

16.9.2 Chip Identification Method

The DSU DID register identifies the device by implementing the following information:

- Processor identification
- Product family identification
- Product series identification
- Device select

16.10 Functional Description

16.10.1 Principle of Operation

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

16.10.2 Basic Operation

16.10.2.1 Initialization

The module is enabled by enabling its clocks.

The DSU registers are PAC write-protected by default after reset.

16.10.2.2 Operation From a Debug Adapter

Debug adapters must access the DSU registers in the external address range [0x100 – 0x1FFF].

If STATUSB.DAL is equal to 0x0, accessing the first 0x100 bytes causes the DSU security filter to return an error to the DAP.

(**PIC32CM LS00/LS60 only**): If STATUSB.DAL is equal to 0x1, debug accesses will go through the DSU security filter but will be forced as non-secure, therefore the DSU internal address space will not be accessible and any access in this case is discarded (writes are ignored, reads return 0) and raise STATUSA.PERR.

Table 16-5. DAP transaction authorizations and error response types

DAL	Debugger Access Type	DAP transaction allowed?			
		DSU internal address space	DSU external address space	Other than PPB	PPB
0	Secure	No (Bus Error)	Yes	No (Bus Error)	No (Bus Error)
0	Non-Secure	No (Bus Error)	Yes	No (Bus Error)	No (Bus Error)
1 (**PIC32CM LS00/LS60 only**)	Secure	No (PERR)	Yes	Yes (NS,PERR)	Yes (ARMv8M)
1 (**PIC32CM LS00/LS60 only**)	Non-Secure	No (PERR)	Yes	Yes (NS,PERR)	Yes (ARMv8M)
2	Secure	Yes	Yes	Yes	Yes
2	Non-Secure	No (PERR)	Yes	Yes	Yes

Bus Error: A Bus Error is sent back to the DAP setting its sticky bit error.

PERR: No bus error, STATUSA.PERR rises, writes are discarded, reads always return 0

NS, PERR: Access forced to non-secure, secure violations are reported in STATUSA.PERR.

Note: Refer to the 29. Non-volatile Memory Controller (NVMCTRL) for details.

16.10.2.3 Operation From the CPU

Only secure hosts can access the DSU internal address space. Attempting to access the internal address space from a non-secure AHB host will report a PAC error, such accesses are discarded. The external address space can be

accessed by either secure or non-secure AHB hosts. The user should access DSU registers in the internal address range (0x0 – 0xFF) to avoid external security restrictions. Refer to 16.8. Security Enforcement.

16.10.3 Sleep Mode Operation

The DSU will continue to operate in IDLE mode.

16.10.4 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 from a secure AHB host, the CRC32 can be operated at any memory location
- The external range, the CRC32 operation is not available

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

16.10.4.1 Starting CRC32 Calculation

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

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

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

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

For further information, refer to the 29. Non-volatile Memory Controller (NVMCTRL).

16.10.4.2 Interpreting the Results

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

16.10.5 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 with no security restriction. 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 always accessible from the external address space. When the device starts with the cold-plugging procedure, a specific Boot ROM command is needed to exit the Boot ROM main routine.

Important: This command is allowed only when DAL= 0x2, otherwise the device must be reset to leave the cold plugging state to let the CPU exit the Boot ROM routine and execute the user code.

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

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

Note: The DCC0 and DCC1 registers are shared with the BCC0 and BCC1 registers therefore mixing DCC and BCC communication is not recommended.

For additional information, refer to 29. Non-volatile Memory Controller (NVMCTRL).

16.10.6 Boot Communication Channels

The Boot Communication Channels (BCC0 and BCC1) consist of a pair of registers with associated handshake logic, accessible by both CPU and debugger with no security restriction. The registers are intended to communicate with the CPU while executing the Boot ROM which implements security and failure analysis commands and therefore must not be used for another purpose.

Note: The BCC0 and BCC1 register values are not reset except in case of POR or BOD resets.

Two Boot Communication Channel status bits in the Status B registers (STATUS.BCCDx) indicate whether a new value has been written in BCC0 or BCC1. These bits, BCCD0 and BCCD1, are located in the STATUSB registers. They are automatically set on write and cleared on read.

Note: The DCC0 and DCC1 registers are shared with the BCC0 and BCC1 registers therefore using DCC is not recommended while the Boot ROM is being executed.

16.10.7 Testing of On-Board Memories MBIST

The DSU implements a feature for automatic testing of memory also known as memory built-in self test (MBIST). This is primarily intended for production test of on-board memories. MBIST cannot be operated from the external address range when DAL<0x2. If an MBIST command is issued when the device is protected, it is filtered-out, a protection error is reported in the Protection Error bit in the Status A register (STATUSA.PERR) and STATUSA.DONE don't rise.

1. Algorithm
   The algorithm used for testing is a type of March algorithm called "March LR". This algorithm is able to detect a wide range of memory defects, while still keeping a linear run time. The algorithm is:
   a. Write entire memory to '0', in any order.
   b. Bit for bit read '0', write '1', in descending order.
   c. Bit for bit read '1', write '0', read '0', write '1', in ascending order.
   d. Bit for bit read '1', write '0', in ascending order.
   e. Bit for bit read '0', write '1', read '1', write '0', in ascending order.
   f. Read '0' from entire memory, in ascending order.

   The specific implementation used has a run time which depends on the CPU clock frequency and the number of bytes tested in the RAM. The detected faults are:
   - Address decoder faults
   - Stuck-at faults
   - Transition faults
   - Coupling faults
   - Linked Coupling faults
2. Starting MBIST
   To test a memory, you need to write the start address of the memory to the ADDR.ADDR bit field, and the size of the memory into the Length register.

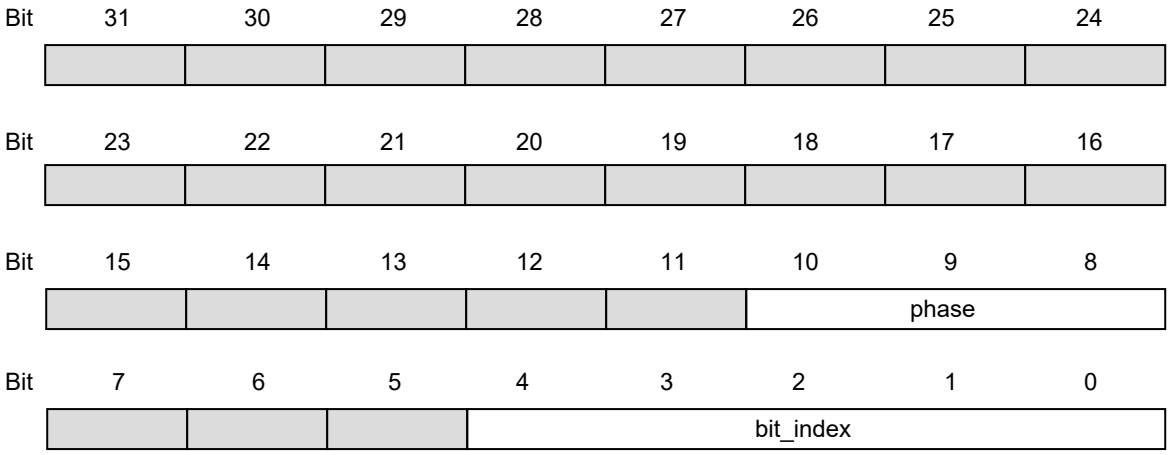
   For best test coverage, an entire physical memory block should be tested at once. It is possible to test only a subset of a memory, but the test coverage will then be somewhat lower.

   The actual test is started by writing a '1' to CTRL.MBIST. A running MBIST operation can be canceled by writing a '1' to CTRL.SWRST.
3. Interpreting the Results
   The tester should monitor the STATUSA register. When the operation is completed, STATUSA.DONE is set. There are two different modes:
   - ADDR.AMOD = 0: exit-on-error (default)
     In this mode, the algorithm terminates either when a fault is detected or on successful completion. In both cases, STATUSA.DONE is set. If an error was detected, STATUSA.FAIL will be set. User then can read the DATA and ADDR registers to locate the fault.
   - ADDR.AMOD = 1: pause-on-error
     In this mode, the MBIST algorithm is paused when an error is detected. In such a situation, only STATUSA.FAIL is asserted. The state machine waits for user to clear STATUSA.FAIL by writing a '1' in

STATUSA.FAIL to resume. Prior to resuming, user can read the DATA and ADDR registers to locate the fault.

4. Locating Faults
If the test stops with STATUSA.FAIL set, one or more bits failed the test. The test stops at the first detected error. The position of the failing bit can be found by reading the following registers:
   – ADDR: Address of the word containing the failing bit
   – DATA: contains data to identify which bit failed, and during which phase of the test it failed. The DATA register will in this case contains the following bit groups:

Figure 16-6. DATA bits Description When MBIST Operation Returns an Error



- bit_index: contains the bit number of the failing bit
- phase: indicates which phase of the test failed and the cause of the error, as listed in the following table.

Table 16-6. MBIST Operation Phases

Phase	Test actions
0	Write all bits to zero. This phase cannot fail.
1	Read '0', write '1', increment address
2	Read '1', write '0'
3	Read '0', write '1', decrement address
4	Read '1', write '0', decrement address
5	Read '0', write '1'
6	Read '1', write '0', decrement address
7	Read all zeros. bit_index is not used

Table 16-7. AMOD Bit Descriptions for MBIST

AMOD[1:0]	Description
0x0	Exit on Error
0x1	Pause on Error
0x2, 0x3	Reserved

For additional information, refer to the 29. Non-volatile Memory Controller (NVMCTRL).

16.10.8 System Services Availability when Accessed Externally

External access: Access performed in the DSU address offset 0x100-0x1FFF range.

Internal access: Access performed in the DSU address offset 0x0-0xFF range.

Table 16-8. Available Features when Operated From The External Address Range and Device is Protected

Features	Availability From The External Address Range when DAL<2
CRC32	No
CoreSight Compliant Device identification	Yes
Debug communication channels	Yes
Boot communication channels	Yes
Testing of onboard memories (MBIST)	No

16.11 Register Summary

Refer to the Registers Description section for more details on register properties and access permissions.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRL	7:0					MBIST	CRC		SWRST
0x01	STATUSA	7:0			BREXT	PERR	FAIL	BERR	CRSTEXT	DONE
0x02	STATUSB	7:0	BCCD1	BCCD0	DCCD1	DCCD0	HPE	DBGPRES	DAL[1:0]	
0x03	Reserved									
0x04	ADDR	31:24	ADDR[29:22]							
		23:16	ADDR[21:14]							
		15:8	ADDR[13:6]							
		7:0	ADDR[5:0]						AMOD[1:0]	
0x08	LENGTH	31:24	LENGTH[29:22]							
		23:16	LENGTH[21:14]							
		15:8	LENGTH[13:6]							
		7:0	LENGTH[5:0]							
0x0C	DATA	31:24	DATA[31:24]							
		23:16	DATA[23:16]							
		15:8	DATA[15:8]							
		7:0	DATA[7:0]							
0x10	DCC0	31:24	DATA[31:24]							
		23:16	DATA[23:16]							
		15:8	DATA[15:8]							
		7:0	DATA[7:0]							
0x14	DCC1	31:24	DATA[31:24]							
		23:16	DATA[23:16]							
		15:8	DATA[15:8]							
		7:0	DATA[7:0]							
0x18	DID	31:24	PROCESSOR[3:0]				FAMILY[4:1]			
		23:16	FAMILY[0]		SERIES[5:0]					
		15:8	DIE[3:0]				REVISION[3:0]			
		7:0	DEVSEL[7:0]							
0x1C	CFG	31:24								
		23:16								
		15:8								
		7:0					DCCDMALEVEL1	DCCDMALEVEL0	LQOS[1:0]	
0x20	BCC0	31:24	DATA[31:24]							
		23:16	DATA[23:16]							
		15:8	DATA[15:8]							
		7:0	DATA[7:0]							
0x24	BCC1	31:24	DATA[31:24]							
		23:16	DATA[23:16]							
		15:8	DATA[15:8]							
		7:0	DATA[7:0]							
0x28 ... 0x0FFF	Reserved									
0x1000	ENTRY0	31:24	ADDOFF[19:12]							
		23:16	ADDOFF[11:4]							
		15:8	ADDOFF[3:0]							
		7:0							FMT	EPRES
0x1004	ENTRY1	31:24	Reserved[19:12]							
		23:16	Reserved[11:4]							
		15:8	Reserved[3:0]							
		7:0							Reserved	Reserved

.....continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x1008	END	31:24	END[31:24]							
		23:16	END[23:16]							
		15:8	END[15:8]							
		7:0	END[7:0]							
0x100C ... 0x1FCB	Reserved									
0x1FCC	MEMTYPE	31:24								
		23:16								
		15:8								
		7:0								SMEMP
0x1FD0	PID4	31:24								
		23:16								
		15:8								
		7:0	FKBC[3:0]				JEPCC[3:0]			
0x1FD4 ... 0x1FDF	Reserved									
0x1FE0	PID0	31:24								
		23:16								
		15:8								
		7:0	PARTNBL[7:0]							
0x1FE4	PID1	31:24								
		23:16								
		15:8								
		7:0	JEPIDCL[3:0]				PARTNBH[3:0]			
0x1FE8	PID2	31:24								
		23:16								
		15:8								
		7:0	REVISION[3:0]				JEPU	JEPIDCH[2:0]		
0x1FEC	PID3	31:24								
		23:16								
		15:8								
		7:0	REVAND[3:0]				CUSMOD[3:0]			
0x1FF0	CID0	31:24								
		23:16								
		15:8								
		7:0	PREAMBLEB0[7:0]							
0x1FF4	CID1	31:24								
		23:16								
		15:8								
		7:0	CCLASS[3:0]				PREAMBLE[3:0]			
0x1FF8	CID2	31:24								
		23:16								
		15:8								
		7:0	PREAMBLEB2[7:0]							
0x1FFC	CID3	31:24								
		23:16								
		15:8								
		7:0	PREAMBLEB3[7:0]							

16.11.1 Control

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

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

Bit 3 – MBIST Memory Built-In Self-Test
Writing a '0' to this bit has no effect.
Writing a '1' to this bit starts the memory BIST algorithm.

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.

16.11.2 Status A

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

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

Bit 5 – BREXT Boot ROM Phase Extension
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Boot ROM Phase Extension bit.
This bit is set when a debug adapter Cold-Plugging is detected, which extends the Boot ROM phase.

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 upon access to:
- A reserved address
- CTRL, ADDR, LENGTH, DATA, CFG from the external address space when DAL<2
- The internal address space with a Non-Secure access (security violation) (PIC32CM LS00/LS60 only)

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.

16.11.3 Status B

Name: STATUSB
Offset: 0x0002
Reset: x determined from latest Set DAL or Chip Erase command
Property: -

Bit	7	6	5	4	3	2	1	0
	BCCD1	BCCD0	DCCD1	DCCD0	HPE	DBGPRES	DAL[1:0]	
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	x	0	x	x

Bits 6, 7 – BCCDx BOOT Communication Channel x Dirty [x=1..0]
Writing a '0' to this bit has no effect.
Writing a '1' to this bit has no effect.
This bit is set when BCCx is written.
This bit is cleared when BCCx is read.

Bits 4, 5 – DCCDx Debug Communication Channel x Dirty [x=1..0]
Writing a '0' to this bit has no effect.
Writing a '1' to this bit has no effect.
This bit is set when DCCx is written.
This bit is cleared when DCCx is read.

Bit 3 – 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 and only a power-reset or a external reset can set it again.
Note: For security reasons, Hot-Plugging is not available when the Debug Access Level (DAL) equals to 0x0.

Bit 2 – 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.

Bits 1:0 – DAL[1:0] Debugger Access Level
Indicates the current debugger access level:

Value	Name	Description
0x0	SECURED	Debugger can only access the DSU external address space.
0x1	NON-SECURE DEBUG	Debugger can access only Non-Secure regions (**PIC32CM LS00/LS60 only**).
0x2	FULL DEBUG	Debugger can access any regions.
0x3	Reserved	-

16.11.4 Address

Name: ADDR
Offset: 0x0004
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] Address Mode
The functionality of these bits is dependent on the operation mode.
Bit description when testing on-16.10.7. Testing of On-Board Memories MBIST board memories (MBIST).

16.11.5 Length

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

Bit	31	30	29	28	27	26	25	24
	LENGTH[29:22]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	LENGTH[21:14]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	LENGTH[13:6]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

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

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

16.11.6 Data

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

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

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

Bit	15	14	13	12	11	10	9	8
	DATA[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

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

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

16.11.7 Debug Communication Channel 0

Name: DCC0
Offset: 0x0010
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.

16.11.8 Debug Communication Channel 1

Name: DCC1
Offset: 0x0014
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.

16.11.9 Device Identification

Name: DID
Offset: 0x0018
Reset: Device Dependent
Property: -

Some information in this register are related to the Ordering Information.

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	0	0	1	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	FAMILY[0]		SERIES[5:0]					
Access	R		R	R	R	R	R	R
Reset	1		z	z	z	z	z	z

Bit	15	14	13	12	11	10	9	8
	DIE[3:0]				REVISION[3:0]			
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	y	y	y	y

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.
Note: For this device, the value of this field is 0x2, corresponding to the Arm Cortex-M23 processor.

Bits 27:23 – FAMILY[4:0] Product Family
The value of this field corresponds to the Product Family part of the ordering code.
Note: For this device, the value of this field is 0x1, corresponding to the PIC32CM Ultra-Low Power Entry Level Families.

Bits 21:16 – SERIES[5:0] Product Series
The value of this field corresponds to the Key Feature Set and Family Variant part of the ordering code.

Value	Description
0x00-0x04	Reserved
0x05	PIC32CM LE00
0x06	PIC32CM LS00
0x07	PIC32CM LS60
0x08-0x3F	Reserved

Bits 15:12 – DIE[3:0] Die Number
Identifies the family die number.
Note: For this device, the value of this field is 0x0.

Bits 11:8 – REVISION[3:0] Revision Number
Identifies the family revision number.

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

This bit field identifies a device within a product family and product series. Refer to Ordering Information for device configurations and corresponding values for Flash memory density and pin count.

Value	Description
0x00	512 KB Flash / 64 KB SRAM / 100-pin
0x01	512 KB Flash / 64 KB SRAM / 64-pin
0x02	512 KB Flash / 64 KB SRAM / 48-pin
0x03	Reserved
0x04	256 KB Flash / 32 KB SRAM / 100-pin
0x05	256 KB Flash / 32 KB SRAM / 64-pin
0x06	256 KB Flash / 32 KB SRAM / 48-pin
0x07 – 0xFF	Reserved

16.11.10 Configuration

Name: CFG
Offset: 0x001C
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
					DCCDMALEVEL1	DCCDMALEVEL0	LQOS[1:0]	
Access					R/W	R/W	R/W	R/W
Reset					0	0	1	0

Bit 3 – DCCDMALEVEL1 DMA TriggerLevel 1

Value	Name	Description
0x0	READ	DCC1 trigger is the image of STATUSB.DCC1D, this signals to the DMA that a data is available for read, this is the correct configuration for a channel that reads DCC1.
0x1	WRITE	DCC1 trigger is the image of STATUSB.DCC1D inverted, this signals to the DMA that DCC1 is ready for write, this is the correct configuration for a channel that writes DCC1.

Bit 2 – DCCDMALEVEL0 DMA TriggerLevel 0

Value	Name	Description
0x0	READ	DCC0 trigger is the image of STATUSB.DCC0D, this signals to the DMA that a data is available for read, this is the correct configuration for a channel that reads DCC0.
0x1	WRITE	DCC0 trigger is the image of STATUSB.DCC0D inverted, this signals to the DMA that DCC0 is ready for write, this is the correct configuration for a channel that writes DCC0.

Bits 1:0 – LQOS[1:0] Latency Quality Of Service
Defines the latency quality of service required when accessing the RAM:
0: Background Transfers
1: Bandwidth Sensitive
2: Latency sensitive
3: Latency critical

16.11.11 Boot Communication Channel 0

Name: BCC0
Offset: 0x0020
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.

16.11.12 Boot Communication Channel 1

Name: BCC1
Offset: 0x0024
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.

16.11.13 CoreSight ROM Table Entry 0

Name: ENTRY0
Offset: 0x1000
Reset: 0x9F0FC002 - x determined by Debug Access Level (DAL)
Property: -

Bit	31	30	29	28	27	26	25	24
	ADDOFF[19:12]							
Access	R	R	R	R	R	R	R	R
Reset	1	0	0	1	1	1	1	1

Bit	23	22	21	20	19	18	17	16
	ADDOFF[11:4]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	1	1	1	1

Bit	15	14	13	12	11	10	9	8
	ADDOFF[3:0]							
Access	R	R	R	R				
Reset	1	1	0	0				

Bit	7	6	5	4	3	2	1	0
							FMT	EPRES
Access							R	R
Reset							1	x

Bits 31:12 – ADDOFF[19:0] Address Offset
The base address of the component, relative to the base address of this ROM table.

Bit 1 – FMT Format
Always reads as '1', indicating a 32-bit ROM table.

Bit 0 – EPRES Entry Present
This bit indicates whether an entry is present at this location in the ROM table.
This bit is set at power-up if the Debug Access Level is different from DAL0, indicating that the entry is present.
This bit is cleared at power-up if the Debug Access Level is DAL0, indicating that the entry is not present.

16.11.14 CoreSight ROM Table Entry 1

Name: ENTRY1
Offset: 0x1004
Reset: -
Property: -

Bit	31	30	29	28	27	26	25	24
	Reserved[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
	Reserved[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
	Reserved[3:0]							
Access	R	R	R	R				
Reset	x	x	x	x				

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

Bits 31:12 – Reserved[19:0]

Bit 1 – Reserved

Bit 0 – Reserved

16.11.15 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.

16.11.16 CoreSight ROM Table Memory Type

Name: MEMTYPE
Offset: 0x1FCC
Reset: x determined by Debug Access Level (DAL)
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 Debug Access Level is different from DAL0, indicating that the system memory is accessible from a debug adapter.
This bit is cleared at power-up if the Debug Access Level is DAL0, indicating that the system memory is not accessible from a debug adapter.

16.11.17 Peripheral Identification 4

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

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	FKBC[3:0]				JEPCC[3:0]			
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

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

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

16.11.18 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								

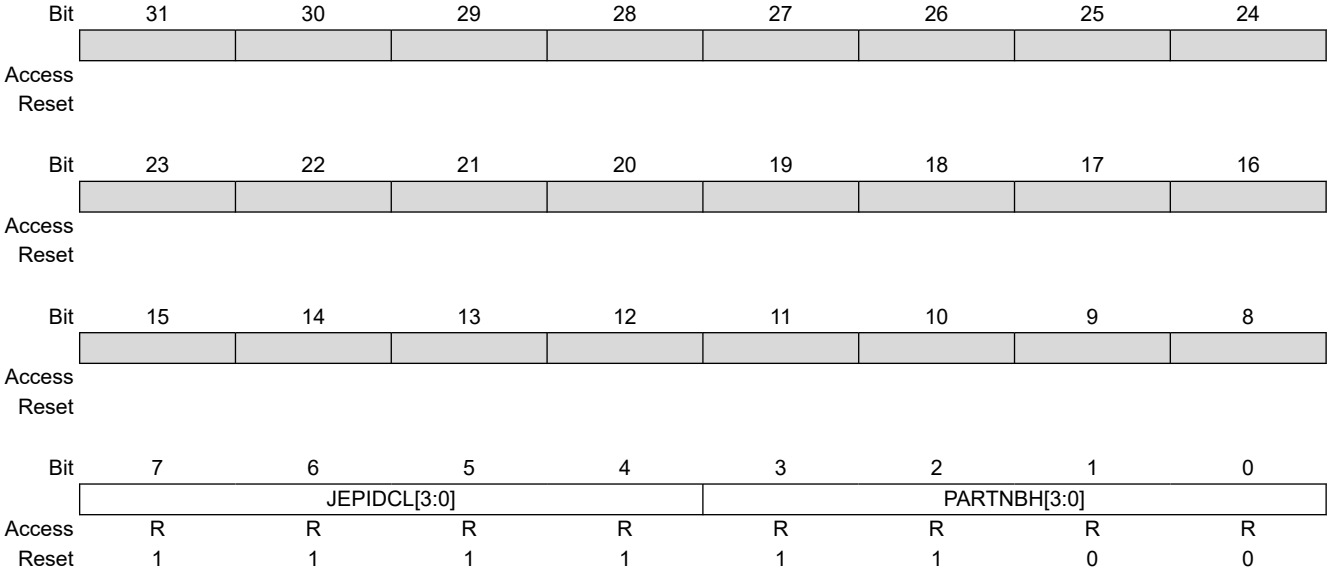
Bit	15	14	13	12	11	10	9	8
Access								
Reset								

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.

16.11.19 Peripheral Identification 1

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



Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	JEPIDCL[3:0]				PARTNBH[3:0]			
Access	R	R	R	R	R	R	R	R
Reset	1	1	1	1	1	1	0	0

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

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

16.11.20 Peripheral Identification 2

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

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	REVISION[3:0]				JEPU	JEPIDCH[2:0]		
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	1	0	0	1

Bits 7:4 – REVISION[3:0] Revision Number
Revision of the peripheral. Starts at 0x0 and increments by one at both major and minor revisions.

Bit 3 – JEPU JEP-106 Identity Code is used
This bit will always return one when read, indicating that JEP-106 code is used.

Bits 2:0 – JEPIDCH[2:0] JEP-106 Identity Code High
These bits will always return 0x1 when read, indicating an Microchip device (Microchip JEP-106 identity code is 0x1F).

16.11.21 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.

16.11.22 Component Identification 0

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

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	PREAMBLEB0[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	1	1	0	1

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

16.11.23 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 (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 0x00 when read.

16.11.24 Component Identification 2

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

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	PREAMBLEB2[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	1	0	1

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

16.11.25 Component Identification 3

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

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	PREAMBLEB3[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	1	0	1	1	0	0	0	1

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

17. Generic Clock Controller (GCLK)

17.1 Overview

Depending on the application, peripherals may require specific clock frequencies to operate correctly. The Generic Clock controller (GCLK) features 8 Generic Clock Generators 0..7 that can provide a wide range of clock frequencies.

Generators can be set to use different external and internal oscillators as source. The clock of each Generator can be divided. The outputs from the Generators are used as sources for the Peripheral Channels, which provide the Generic Clock (GCLK_PERIPH) to the peripheral modules, as shown in 17.3. Block Diagram. The number of Peripheral Clocks depends on how many peripherals the device has.

Note: The Generator 0 is always the direct source of the GCLK_MAIN signal.

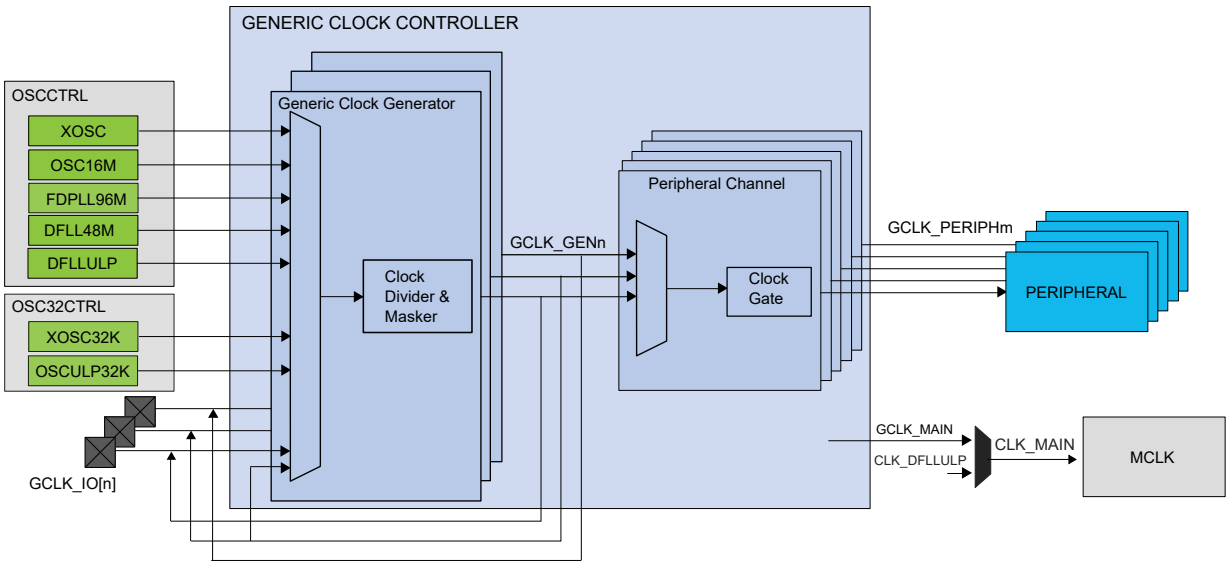
17.2 Features

- Provides a device-defined, configurable number of Peripheral Channel clocks
- Wide frequency range:
  - Various clock sources
  - Embedded dividers

17.3 Block Diagram

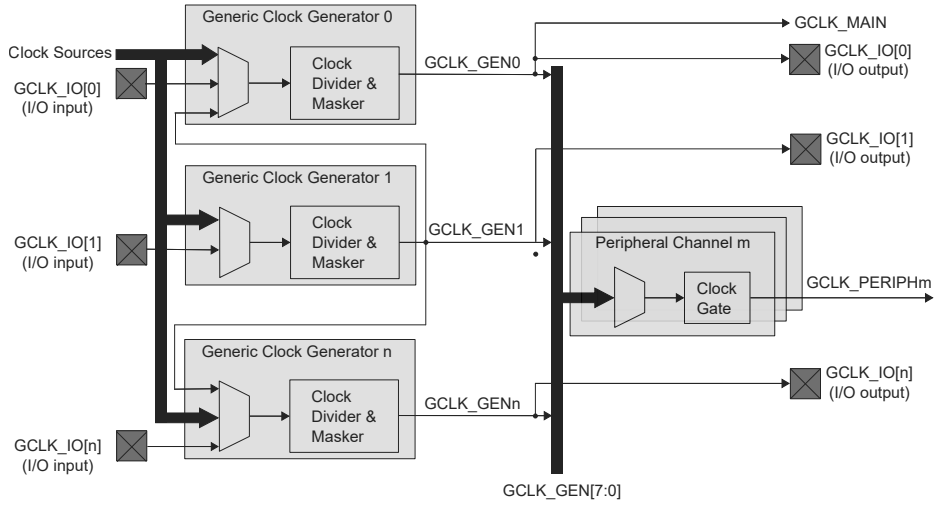
The generation of Peripheral Clock signals (GCLK_PERIPHm) and the Main Clock (GCLK_MAIN) can be seen in the following figure.

Figure 17-1. Device Clocking Diagram



The GCLK block diagram is shown in the following figure:

Figure 17-2. Generic Clock Controller Block Diagram



17.4 Signal Description

Table 17-1. GCLK Signal Description

Signal Name	Type	Description
GCLK_IO[7:0]	Digital I/O	Clock source for Generators when input. Generic Clock signal when output.

Note: One signal can be mapped on several pins.

For further information, refer to the Pinout.

17.5 Peripheral Dependencies

Table 17-2. GCLK Configuration Summary

Peripheral name	Base address	MCLK AHB/APB Clocks	PAC Peripheral Identifier Index (PAC.WRCTRL)	Power Domain (PM.STDBYCFG)
GCLK	0x40001C00	CLK_GCLK_APB	7	PDSW

17.6 Functional Description

17.6.1 Principle of Operation

The GCLK module is comprised of 8 Generic Clock Generators sourcing up to 35 Peripheral Channels and the Main Clock signal CLK_MAIN.

A clock source selected as input to a Generator can either be used directly, or it can be prescaled in the Generator. A generator output is used by one or more Peripheral Channels to provide a peripheral generic clock signal (GCLK_PERIPH) to the peripherals.

17.6.2 Basic Operation

17.6.2.1 Initialization

Before a Generator is enabled, the corresponding clock source should be enabled. The Peripheral clock must be configured as outlined by the following steps:

1. The Generator must be enabled (GENCTRLn.GENEN=1) and the division factor must be set (GENTRLn.DIVSEL and GENCTRLn.DIV) by performing a single 32-bit write to the Generator Control register (GENCTRLn).
2. The Generic Clock for a peripheral must be configured by writing to the respective Peripheral Channel Control register (PCHCTRLm). The Generator used as the source for the Peripheral Clock must be written to the GEN bit field in the Peripheral Channel Control register (PCHCTRLm.GEN).

Note: Each Generator n is configured by one dedicated register GENCTRLn.

Note: Each Peripheral Channel m is configured by one dedicated register PCHCTRLm.

17.6.2.2 Enabling, Disabling, and Resetting

The GCLK module has no enable/disable bit to enable or disable the whole module.

The GCLK is reset by setting the Software Reset bit in the Control A register (CTRLA.SWRST) to 1. All registers in the GCLK will be reset to their initial state, except for Peripheral Channels and associated Generators that have their Write Lock bit set to 1 (PCHCTRLm.WRTLOCK). For further details, refer to 17.6.3.4. Configuration Lock.

17.6.2.3 Generic Clock Generator

Each Generator (GCLK_GEN) can be set to run from one of 9 different clock sources except GCLK_GEN1, which can be set to run from one of 8 sources, as GCLK_GEN1 is the only Generator that can be selected as source to others Generators.

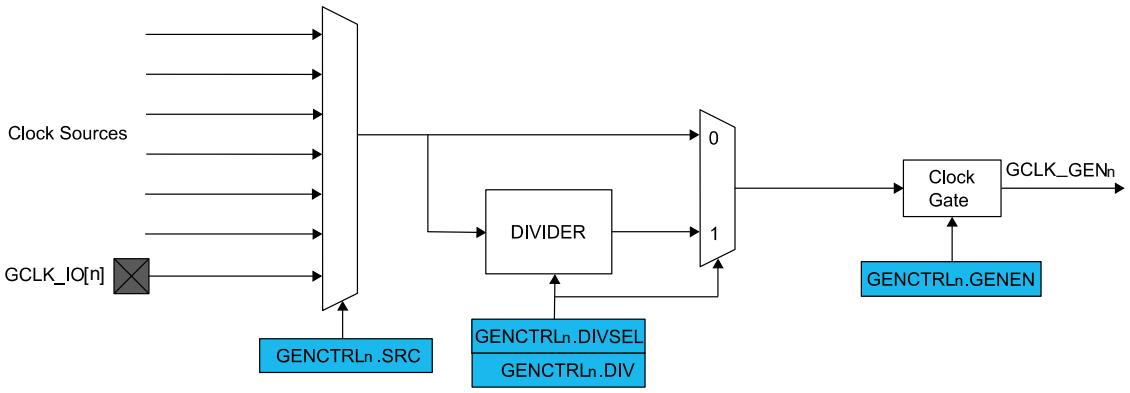
Each generator GCLK_GENn can be connected to one specific pin GCLK_IO[n]. This GCLK_IO[n] pin can be set either to act as source to GCLK_GENn or to output the clock signal generated by GCLK_GENn.

The selected source can be divided. Each Generator can be enabled or disabled independently.

Each GCLK_GENn clock signal can then be used as clock source for Peripheral Channels. Each Generator output is allocated to one or several Peripherals.

GCLK_GEN0 is used as GCLK_MAIN for the synchronous clock controller inside the 18. Main Clock (MCLK) . Refer to the 18. Main Clock (MCLK) description for details on the synchronous clock generation.

Figure 17-3. Generic Clock Generator



17.6.2.4 Enabling a Generator

A Generator is enabled by writing a '1' to the Generator Enable bit in the Generator Control register (GENCTRLn.GENEN=1).

17.6.2.5 Disabling a Generator

A Generator is disabled by writing a '0' to GENCTRLn.GENEN. When GENCTRLn.GENEN=0, the GCLK_GENn clock is disabled and gated.

17.6.2.6 Selecting a Clock Source for the Generator

Each Generator can individually select a clock source by setting the Source Select bit group in the Generator Control register (GENCTRLn.SRC).

Changing from one clock source, for example A, to another clock source, B, can be done on the fly: If clock source B is not ready, the Generator will continue using clock source A. As soon as source B is ready, the Generator will switch to it. During the switching operation, the Generator maintains clock requests to both clock sources A and B, and will release source A as soon as the switch is done. The according bit in SYNCBUSY register (SYNCBUSY.GENCTRLn) will remain '1' until the switch operation is completed.

The available clock sources are device dependent (usually the oscillators, RC oscillators, DPLL, and DFLLULP). Only Generator 1 can be used as a common source for all other generators.

17.6.2.7 Changing the Clock Frequency

The selected source for a Generator can be divided by writing a division value in the Division Factor bit field of the Generator Control register (GENCTRLn.DIV). How the actual division factor is calculated is depending on the Divide Selection bit (GENCTRLn.DIVSEL).

If GENCTRLn.DIVSEL=0 and GENCTRLn.DIV is either 0 or 1, the output clock will be undivided.

Note: The number of available DIV bits may vary from Generator to Generator.

17.6.2.8 Duty Cycle

When dividing a clock with an odd division factor, the duty-cycle will not be 50/50. Setting the Improve Duty Cycle bit of the Generator Control register (GENCTRLn.IDC) will result in a 50/50 duty cycle.

17.6.2.9 External Clock

The output clock (GCLK_GENn) of each Generator can be sent to I/O pins (GCLK_IO[n]).

If the Output Enable bit in the Generator Control register is set (GENCTRLn.OE = 1) and the generator is enabled (GENCTRLn.GENEN=1), the Generator requests its clock source and the GCLK_GENn clock is output to an I/O pin.
Note: The I/O pin (GCLK_IO[n]) must first be configured as output by writing the corresponding PORT registers.

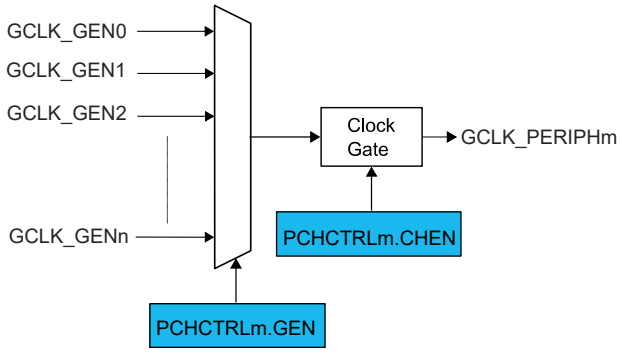
If GENCTRLn.OE is 0, the according I/O pin is set to an Output Off Value, which is selected by GENCTRLn.OOV: If GENCTRLn.OOV is '0', the output clock will be low. If this bit is '1', the output clock will be high.

In Standby mode, if the clock is output (GENCTRLn.OE=1), the clock on the I/O pin is frozen to the OOV value if the Run In Standby bit of the Generic Control register (GENCTRLn.RUNSTDBY) is zero.
Note: With GENCTRLn.OE=1 and RUNSTDBY=0, entering the Standby mode can take longer due to a clock source dependent delay between turning off Power Domain PDSW. The maximum delay can be equal to the clock source period multiplied by the division factor.

If GENCTRLn.RUNSTDBY is '1', the GCLK_GENn clock is kept running and output to the I/O pin.

17.6.3 Peripheral Clock

Figure 17-4. Peripheral Clock



17.6.3.1 Enabling a Peripheral Clock

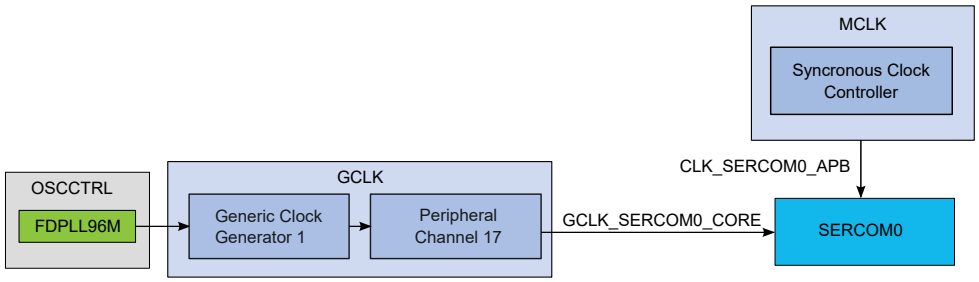
Before a Peripheral Clock is enabled, one of the Generators must be enabled (GENCTRLn.GENEN) and selected as source for the Peripheral Channel by setting the Generator Selection bits in the Peripheral Channel Control register (PCHCTRLm.GEN). Any available Generator can be selected as clock source for each Peripheral Channel.

When a Generator has been selected, the peripheral clock is enabled by setting the Channel Enable bit in the Peripheral Channel Control register, PCHCTRLm.CHEN = 1. The PCHCTRLm.CHEN bit must be synchronized to the generic clock domain. PCHCTRLm.CHEN will continue to read as its previous state until the synchronization is complete.

The following figure illustrates SERCOM0 clock distribution:

- The FDPLL96M is used as clock source
- The Generic Clock Generator 1 selects the FDPLL96M as clock source
- The Peripheral Channel (17 for SERCOM0) selects the Generic Clock Generator 1 to generate the Generic Clock: GCLK_SERCOM0_CORE
- The SERCOM0 user interface is clocked by CLK_SERCOM0_APB

Figure 17-5. SERCOM0 Clock Distribution



This results in the following register configuration:

- The source oscillator for a generic clock generator 'n' is selected by writing to the Source bit field in the Generator Control n register (GCLK.GENCTRLn.SRC)
- A Peripheral Channel m can be configured to use a specific Generic Clock Generator by writing to the Generic Clock Generator bit field in the respective Peripheral Channel m register (GCLK.PCHCTRLm.GEN)
- The Peripheral Channel number, *m*, is fixed for a given peripheral. See the Mapping table in the description of GCLK.PCHCTRLm.

17.6.3.2 Disabling a Peripheral Clock

A Peripheral Clock is disabled by writing PCHCTRLm.CHEN=0. The PCHCTRLm.CHEN bit must be synchronized to the Generic Clock domain. PCHCTRLm.CHEN will stay in its previous state until the synchronization is complete. The Peripheral Clock is gated when disabled.

17.6.3.3 Selecting the Clock Source for a Peripheral

When changing a peripheral clock source by writing to PCHCTRLm.GEN, the peripheral clock must be disabled before re-enabling it with the new clock source setting. This prevents glitches during the transition:

1. Disable the Peripheral Channel by writing PCHCTRLm.CHEN=0.
2. Assert that PCHCTRLm.CHEN reads '0'.
3. Change the source of the Peripheral Channel by writing PCHCTRLm.GEN.
4. Re-enable the Peripheral Channel by writing PCHCTRLm.CHEN=1.
5. Assert that PCHCTRLm.CHEN reads '1'.

17.6.3.4 Configuration Lock

The peripheral clock configuration can be locked for further write accesses by setting the Write Lock bit in the Peripheral Channel Control register (PCHCTRLm.WRTLOCK=1). All writing to the PCHCTRLm register will be ignored. It can only be unlocked by a Power Reset.

The Generator source of a locked Peripheral Channel will be locked, too: The corresponding GENCTRLn register is locked, and can be unlocked only by a Power Reset.

There is one exception concerning the Generator 0. As it is used as GCLK_MAIN, it cannot be locked. It is reset by any Reset and will start up in a known configuration. The software reset (CTRLA.SWRST) can not unlock the registers.

In case of an external Reset, the Generator source will be disabled. Even if the WRTLOCK bit is written to '1' the peripheral channels are disabled (PCHCTRLm.CHEN set to '0') until the Generator source is enabled again. Then, the PCHCTRLm.CHEN are set to '1' again.

17.6.4 Additional Features

17.6.4.1 Peripheral Clock Enable after Reset

The Generic Clock Controller must be able to provide a generic clock to some specific peripherals after a Reset. That means that the configuration of the Generators and Peripheral Channels after Reset is device-dependent.

Refer to GENCTRLn.SRC for details on GENCTRLn reset.

Refer to 17.7.5. PCHCTRLm.SRC for details on PCHCTRLm reset.

17.6.5 Sleep Mode Operation

17.6.5.1 SleepWalking

The GCLK module supports the SleepWalking feature.

If the system is in a sleep mode where the Generic Clocks are stopped, a peripheral that needs its clock in order to execute a process must request it from the Generic Clock Controller.

The Generic Clock Controller receives this request, determines which Generic Clock Generator is involved and which clock source needs to be awakened. It then wakes up the respective clock source, enables the Generator and Peripheral Channel stages successively, and delivers the clock to the peripheral.

The RUNSTDBY bit in the Generator Control register controls clock output to pin during standby sleep mode. If the bit is cleared, the Generator output is not available on pin. When set, the GCLK can continuously output the generator output to GCLK_IO[n]. Refer to 17.6.2.9. External Clock for details.

17.6.5.2 Minimize Power Consumption in Standby

The following table identifies when a Clock Generator is off in Standby Mode, minimizing the power consumption:

Table 17-3. Clock Generator n Activity in Standby Mode

Request for Clock n present	GENCTRLn.RUNSTDBY	GENCTRLn.OE	Clock Generator n
Yes	-	-	ACTIVE
No	1	1	ACTIVE
No	1	0	OFF
No	0	1	OFF
No	0	0	OFF

17.6.5.3 Entering Standby Mode

There may occur a delay when the device is put into Standby, until the power is turned off. This delay is caused by running Clock Generators: if the Run in Standby bit in the Generator Control register (GENCTRLn.RUNSTDBY) is '0', GCLK must verify that the clock is turned of properly. The duration of this verification is frequency-dependent.

For further information, refer to the 22. Power Manager (PM).

17.6.6 Debug Operation

When the CPU is halted in debug mode the GCLK continues normal operation. If the GCLK 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.

17.6.7 Synchronization

Some registers (or bit fields within a register) require synchronization when read and/or written.

Synchronization is denoted by the "Read-Synchronized" (or "Read-Synchronized Bits”) and/or "Write-Synchronized" (or "Write-Synchronized Bits”) property in each individual register description.

For more details, refer to Register Synchronization.

17.7 Register Summary

Refer to the Registers Description section for more details on register properties and access permissions.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRLA	7:0								SWRST
0x01 ... 0x03	Reserved									
0x04	SYNCBUSY	31:24								
		23:16								
		15:8							GENCTRL7	GENCTRL6
		7:0	GENCTRL5	GENCTRL4	GENCTRL3	GENCTRL2	GENCTRL1	GENCTRL0		SWRST
0x08 ... 0x1F	Reserved									
0x20	GENCTRL0	31:24								
		23:16	DIV[7:0]							
		15:8			RUNSTDBY	DIVSEL	OE	OOV	IDC	GENEN
		7:0					SRC[3:0]			
0x24	GENCTRL1	31:24	DIV[15:8]							
		23:16	DIV[7:0]							
		15:8			RUNSTDBY	DIVSEL	OE	OOV	IDC	GENEN
		7:0					SRC[3:0]			
0x28	GENCTRL2	31:24								
		23:16	DIV[7:0]							
		15:8			RUNSTDBY	DIVSEL	OE	OOV	IDC	GENEN
		7:0					SRC[3:0]			
0x2C	GENCTRL3	31:24								
		23:16	DIV[7:0]							
		15:8			RUNSTDBY	DIVSEL	OE	OOV	IDC	GENEN
		7:0					SRC[3:0]			
0x30	GENCTRL4	31:24								
		23:16	DIV[7:0]							
		15:8			RUNSTDBY	DIVSEL	OE	OOV	IDC	GENEN
		7:0					SRC[3:0]			
0x34	GENCTRL5	31:24								
		23:16	DIV[7:0]							
		15:8			RUNSTDBY	DIVSEL	OE	OOV	IDC	GENEN
		7:0					SRC[3:0]			
0x38	GENCTRL6	31:24								
		23:16	DIV[7:0]							
		15:8			RUNSTDBY	DIVSEL	OE	OOV	IDC	GENEN
		7:0					SRC[3:0]			
0x3C	GENCTRL7	31:24								
		23:16	DIV[7:0]							
		15:8			RUNSTDBY	DIVSEL	OE	OOV	IDC	GENEN
		7:0					SRC[3:0]			
0x40 ... 0x7F	Reserved									
0x80	PCHCTRL0	31:24								
		23:16								
		15:8								
		7:0	WRTLOCK	CHEN				GEN[2:0]		
...										
0x0108	PCHCTRL34	31:24								
		23:16								
		15:8								
		7:0	WRTLOCK	CHEN				GEN[2:0]		

17.7.1 Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized Bits

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

Bit 0 – SWRST Software Reset
Writing a zero to this bit has no effect.
Setting this bit to 1 will reset all registers in the GCLK to their initial state after a Power Reset, except for generic clocks and associated Generators that have their WRTLOCK bit in PCHCTRLm set to 1.
Refer to GENCTRLn Reset Value for details on GENCTRLn register reset.
Refer to PCHCTRLm Reset Value for details on PCHCTRLm register reset.
Notes:

1. When the CTRLA.SWRST is written, the user must poll the SYNCBUSY.SWRST bit to know when the reset operation is complete.
2. During a SWRST, access to registers/bits without SWRST are disallowed until the SYNCBUSY.SWRST is cleared by hardware.

Value	Description
0	There is no Reset operation ongoing.
1	A Reset operation is ongoing.

17.7.2 Synchronization Busy

Name: SYNCBUSY
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
							GENCTRL7	GENCTRL6
Access							R	R
Reset							0	0

Bit	7	6	5	4	3	2	1	0
	GENCTRL5	GENCTRL4	GENCTRL3	GENCTRL2	GENCTRL1	GENCTRL0		SWRST
Access	R	R	R	R	R	R		R
Reset	0	0	0	0	0	0		0

Bits 2, 3, 4, 5, 6, 7, 8, 9 – GENCTRLn Generator Control n Synchronization Busy [n=0..4]
This bit is cleared when the synchronization of the Generator Control n register (GENCTRLn) between clock domains is complete, or when clock switching operation is complete.
This bit is set when the synchronization of the Generator Control n register (GENCTRLn) between clock domains is started.

Bit 0 – SWRST Software Reset Synchronization Busy
This bit is cleared when the synchronization of the CTRLA.SWRST register bit between clock domains is complete.
This bit is set when the synchronization of the CTRLA.SWRST register bit between clock domains is started.

17.7.3 Generator Control (GENCTRL0,2,3,4,5,6,7)

Name: GENCTRLn
Offset: 0x20,0x28,0x2C,0x30,0x34,0x38,0x3C
Reset: 0x00000105 (GENCTRL0), 0x00000000 (others)
Property: PAC Write-Protection, Write-Synchronized

GENCTRLn controls the settings of Generic Generator n (n=0,2,3,4,5,6,7).

Important: GENCTRL1 has its own register described in GENCTRL1 register.

A User Reset will reset the associated GENCTRLn registers unless the Generator is the source of a locked Peripheral Channel m (PCHCTRLm.WRTLOCK=1).

Note: GENCTRLn is a write-synchronized register: SYNCBUSY.GENCTRLn must be checked to ensure the GENCTRLn synchronization is complete.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	DIV[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
			RUNSTDBY	DIVSEL	OE	OOV	IDC	GENEN
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	x

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

Bits 23:16 – DIV[7:0] Division Factor
These bits represent a division value for the corresponding Generator. The actual division factor is dependent on the state of DIVSEL.

Bit 13 – RUNSTDBY Run in Standby
This bit is used to keep the Generator running in Standby as long as it is configured to output to a dedicated GCLK_IO[n] pin. If GENCTRLn.OE is zero, this bit has no effect and the generator will only be running if a peripheral requires the clock.

Value	Description
0	The Generator is stopped in Standby and the GCLK_IO[n] pin state (one or zero) will be dependent on the setting in GENCTRL.OOV.
1	The Generator is kept running and output to its dedicated GCLK_IO[n] pin during Standby mode.

Bit 12 – DIVSEL Divide Selection
This bit determines how the division factor of the clock source of the Generator will be calculated from DIV. If the clock source should not be divided, DIVSEL must be 0 and the GENCTRLn.DIV value must be either 0 or 1.

Value	Description
0	The Generator clock frequency equals the clock source frequency divided by GENCTRLn.DIV.
1	The Generator clock frequency equals the clock source frequency divided by 2^(GENCTRLn.DIV + 1).

Bit 11 – OE Output Enable

This bit is used to output the Generator clock output to the corresponding pin (GCLK_IO[n]), as long as GCLK_IO[n] is not defined as the Generator source in the GENCTRLn.SRC bit field.

Value	Description
0	No Generator clock signal on pin GCLK_IO[n].
1	The Generator clock signal is output on the corresponding GCLK_IO[n], unless GCLK_IO[n] is selected as a generator source in the GENCTRLn.SRC bit field.

Bit 10 – OOV Output Off Value

This bit is used to control the clock output value on pin (GCLK_IO[n]) when the Generator is turned off or the OE bit is zero, as long as GCLK_IO[n] is not defined as the Generator source in the GENCTRLn.SRC bit field.

Value	Description
0	The GCLK_IO[n] will be LOW when generator is turned off or when the OE bit is zero.
1	The GCLK_IO[n] will be HIGH when generator is turned off or when the OE bit is zero.

Bit 9 – IDC Improve Duty Cycle

This bit is used to improve the duty cycle of the Generator output to 50/50 for odd division factors.

Value	Description
0	Generator output clock duty cycle is not balanced to 50/50 for odd division factors.
1	Generator output clock duty cycle is 50/50.

Bit 8 – GENEN Generator Enable

This bit is used to enable and disable the Generator.

Value	Description
0	Generator is disabled.
1	Generator is enabled.

Bits 3:0 – SRC[3:0] Generator Clock Source Selection

These bits select the Generator clock source, as shown in this table.

Table 17-4. Generator Clock Source Selection

Value	Name	Description
0x00	XOSC	XOSC oscillator output
0x01	GCLK_IN	Generator input pad (GCLK_IO[n])
0x02	GCLK_GEN1	Generic clock generator 1 output
0x03	OSCULP32K	OSCULP32K oscillator output
0x04	XOSC32K	XOSC32K oscillator output
0x05	OSC16M	OSC16M oscillator output
0x06	DFLLULP	DFLLULP ultra low power output
0x07	DFLL48M	DFLL48M output
0x08	FDPLL96M	FDPLL96M output
0x09-0x0F	Reserved	Reserved for future use

17.7.4 Generator Control (GENCTRL1)

Name: GENCTRL1
Offset: 0x24
Reset: 0x00000000
Property: PAC Write-Protection, Write-Synchronized

GENCTRLn controls the settings of Generic Generator n (n=1).

A User Reset will reset the associated GENCTRLn registers unless the Generator is the source of a locked Peripheral Channel m (PCHCTRLm.WRTLOCK=1).

Note: GENCTRL is a write-synchronized register: SYNCBUSY.GENCTRLn must be checked to ensure the GENCTRLn synchronization is complete.

Bit	31	30	29	28	27	26	25	24
	DIV[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
	DIV[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
			RUNSTDBY	DIVSEL	OE	OOV	IDC	GENEN
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	x

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

Bits 31:16 – DIV[15:0] Division Factor
These bits represent a division value for the corresponding Generator. The actual division factor is dependent on the state of DIVSEL.

Bit 13 – RUNSTDBY Run in Standby
This bit is used to keep the Generator running in Standby as long as it is configured to output to a dedicated GCLK_IO[n] pin. If GENCTRLn.OE is zero, this bit has no effect and the generator will only be running if a peripheral requires the clock.

Value	Description
0	The Generator is stopped in Standby and the GCLK_IO[n] pin state (one or zero) will be dependent on the setting in GENCTRL.OOV.
1	The Generator is kept running and output to its dedicated GCLK_IO[n] pin during Standby mode.

Bit 12 – DIVSEL Divide Selection
This bit determines how the division factor of the clock source of the Generator will be calculated from DIV. If the clock source should not be divided, DIVSEL must be 0 and the GENCTRLn.DIV value must be either 0 or 1.

Value	Description
0	The Generator clock frequency equals the clock source frequency divided by GENCTRLn.DIV.
1	The Generator clock frequency equals the clock source frequency divided by 2^(GENCTRLn.DIV + 1).

Bit 11 – OE Output Enable
This bit is used to output the Generator clock output to the corresponding pin (GCLK_IO[n]), as long as GCLK_IO[n] is not defined as the Generator source in the GENCTRLn.SRC bit field.

Value	Description
0	No Generator clock signal on pin GCLK_IO[n].
1	The Generator clock signal is output on the corresponding GCLK_IO[n], unless GCLK_IO[n] is selected as a generator source in the GENCTRLn.SRC bit field.

Bit 10 – OOV Output Off Value

This bit is used to control the clock output value on pin (GCLK_IO[n]) when the Generator is turned off or the OE bit is zero, as long as GCLK_IO[n] is not defined as the Generator source in the GENCTRLn.SRC bit field.

Value	Description
0	The GCLK_IO[n] will be LOW when generator is turned off or when the OE bit is zero.
1	The GCLK_IO[n] will be HIGH when generator is turned off or when the OE bit is zero.

Bit 9 – IDC Improve Duty Cycle

This bit is used to improve the duty cycle of the Generator output to 50/50 for odd division factors.

Value	Description
0	Generator output clock duty cycle is not balanced to 50/50 for odd division factors.
1	Generator output clock duty cycle is 50/50.

Bit 8 – GENEN Generator Enable

This bit is used to enable and disable the Generator.

Value	Description
0	Generator is disabled.
1	Generator is enabled.

Bits 3:0 – SRC[3:0] Generator Clock Source Selection

These bits select the Generator clock source, as shown in this table.

Table 17-5. Generator Clock Source Selection

Value	Name	Description
0x00	XOSC	XOSC oscillator output
0x01	GCLK_IN	Generator input pad (GCLK_IO[n])
0x02	GCLK_GEN1	Generic clock generator 1 output
0x03	OSCULP32K	OSCULP32K oscillator output
0x04	XOSC32K	XOSC32K oscillator output
0x05	OSC16M	OSC16M oscillator output
0x06	DFLLULP	DFLLULP ultra low power output
0x07	DFLL48M	DFLL48M output
0x08	FDPLL96M	FDPLL96M output
0x09-0x0F	Reserved	Reserved for future use

17.7.5 Peripheral Channel Control

Name: PCHCTRLm
Offset: 0x80 + m*0x04 [m=0..34]
Reset: 0x00000000
Property: PAC Write-Protection

PCHTRLm controls the settings of Peripheral Channel number m (m=0..34).

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
	WRTLOCK	CHEN				GEN[2:0]		
Access	R/W	R/W				R/W	R/W	R/W
Reset	0	0				0	0	0

Bit 7 – WRTLOCK Write Lock
After this bit is set to '1', further writes to the PCHCTRLm register will be discarded. The control register of the corresponding Generator n (GENCTRLn), as assigned in PCHCTRLm.GEN, will also be locked. It can only be unlocked by a Power Reset.
Note that Generator 0 cannot be locked.

Value	Description
0	The Peripheral Channel register and the associated Generator register are not locked
1	The Peripheral Channel register and the associated Generator register are locked

Bit 6 – CHEN Channel Enable
This bit is used to enable and disable a Peripheral Channel.

⚠ CAUTION This bit requires synchronization. When changing it, the bit value must be read-back (polling method) to ensure the synchronization is complete. Changing the bit value under ongoing synchronization will *not* generate an error.

Value	Description
0	The Peripheral Channel is disabled
1	The Peripheral Channel is enabled

Bits 2:0 – GEN[2:0] Generator Selection
This bit field selects the Generator to be used as the source of a peripheral clock, as shown in the table below:

Table 17-6. Generator Selection

Value	Description
0x0	Generic Clock Generator 0

.....continued	
Value	**Description**
0x1	Generic Clock Generator 1
0x2	Generic Clock Generator 2
0x3	Generic Clock Generator 3
0x4	Generic Clock Generator 4
0x5	Generic Clock Generator 5
0x6	Generic Clock Generator 6
0x7	Generic Clock Generator 7

A Power Reset will reset all the PCHCTRLm registers.
A User Reset will reset a PCHCTRL if WRTLOCK=0, or else, the content of that PCHCTRL remains unchanged.

Table 17-7. Reset Value after a User Reset or a Power Reset

Reset	PCHCTRLm.GEN	PCHCTRLm.CHEN	PCHCTRLm.WRTLOCK
Power Reset	0x0	0x0	0x0
User Reset	If WRTLOCK = 0 : 0x0 If WRTLOCK = 1: no change	If WRTLOCK = 0 : 0x0 If WRTLOCK = 1: no change	No change

The PCHCTRLm registers index values are shown in the table below:

Table 17-8. PCHCTRLm Mapping

index(m)	Name	Description
0	GCLK_FDPLL96M	FDPLL96M input clock source for reference
1	GCLK_FDPLL96M_32K	FDPLL96M slow clock for FDPLL96M internal clock timer
2	GCLK_DFLLULP	DFLLULP clock for DFLLULP
3	GCLK_DFLL48M	DFLL48M clock for DFLL48M
4	GCLK_EIC	EIC
5	GCLK_FREQM_MSR	FREQM Measure
6	GCLK_FREQM_REF	FREQM Reference
7	GCLK_USB	USB
8	GCLK_EVSYS_CHANNEL_0	EVSYS_CHANNEL_0
9	GCLK_EVSYS_CHANNEL_1	EVSYS_CHANNEL_1
10	GCLK_EVSYS_CHANNEL_2	EVSYS_CHANNEL_2
11	GCLK_EVSYS_CHANNEL_3	EVSYS_CHANNEL_3
12	GCLK_EVSYS_CHANNEL_4	EVSYS_CHANNEL_4
13	GCLK_EVSYS_CHANNEL_5	EVSYS_CHANNEL_5
14	GCLK_EVSYS_CHANNEL_6	EVSYS_CHANNEL_6
15	GCLK_EVSYS_CHANNEL_7	EVSYS_CHANNEL_7
16	GCLK_SERCOM0_SLOW GCLK_SERCOM1_SLOW GCLK_SERCOM2_SLOW GCLK_SERCOM3_SLOW GCLK_SERCOM4_SLOW GCLK_SERCOM5_SLOW	SERCOM0_SLOW SERCOM1_SLOW SERCOM2_SLOW SERCOM3_SLOW SERCOM4_SLOW SERCOM5_SLOW
17	GCLK_SERCOM0_CORE	SERCOM0_CORE
18	GCLK_SERCOM1_CORE	SERCOM1_CORE
19	GCLK_SERCOM2_CORE	SERCOM2_CORE
20	GCLK_SERCOM3_CORE	SERCOM3_CORE
21	GCLK_SERCOM4_CORE	SERCOM4_CORE
22	GCLK_SERCOM5_CORE	SERCOM5_CORE
23	GCLK_TC0_TC1	TC0,TC1

.....continued		
index(m)	**Name**	**Description**
24	GCLK_TC2	TC2
25	GCLK_TCC0_TCC1	TCC0,TCC1
26	GCLK_TCC2	TCC2
27	GCLK_TCC3	TCC3
28	GCLK_ADC	ADC
29	GCLK_AC	AC
30	GCLK_DAC	DAC
31	GCLK_PTC	PTC
32	GCLK_CCL	CCL
33	GCLK_I2S_0	I2S Clock Unit 0
34	GCLK_I2S_1	I2S Clock Unit 1

18. Main Clock (MCLK)

18.1 Overview

The Main Clock (MCLK) controls the synchronous clock generation of the device.

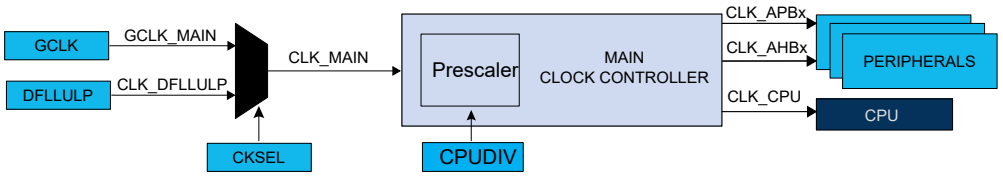
Using a clock provided by the Generic Clock Controller (GCLK_MAIN) or the DFLLULP (CLK_DFLLULP), the Main Clock Controller provides synchronous system clocks to the CPU and the modules connected to the AHBx and the APBx bus. The synchronous system clocks are divided into a number of clock domains.

18.2 Features

- Generates CPU, AHB, and APB system clocks
  - Clock source and division factor from GCLK
  - Clock prescaler with 1x to 128x division
- Safe run-time clock switching from GCLK
- Module-level clock gating through maskable peripheral clocks

18.3 Block Diagram

Figure 18-1. MCLK Block Diagram



18.4 Peripheral Dependencies

Table 18-1. MCLK Configuration Summary

Peripheral name	Base address	NVIC IRQ Index	MCLK AHB/APB Clocks	PAC Peripheral Identifier Index (PAC.WRCTRL)	Power Domain (PM.STDBYCFG)
MCLK	0x40000800	0: CKRDY	CLK_MCLK_APB	2	PDSW

18.5 Functional Description

18.5.1 Principle of Operation

The GCLK Main (GCLK_MAIN) or the DFLLULP (CLK_DFLLULP) clocks are the sources for the main clock (CLK_MAIN), which in turn is the common root for the synchronous clocks for the CPU, APBx, and AHBx modules. The CLK_MAIN is divided by an 8-bit prescaler. The clock domain (CPU) can be changed on the fly to respond to variable load in the application. Depending on the sleep mode, some clock domains can be turned off.

The APBx clocks (CLK_APBx) and the AHBx clocks (CLK_AHBx) are the root clock source used by modules requiring a clock on the APBx and the AHBx bus. These clocks are always synchronous to the CPU clock (CLK_CPU) and can run even when the CPU clock is turned off in sleep mode. A clock gater is inserted after the common APB clock to gate any APBx clock of a module on APBx bus, as well as the AHBx clock.

The device has the following synchronous clock domain:

- CPU synchronous clock domain (CPU Clock Domain). Frequency is f_{CPU}.

18.5.2 Basic Operation

18.5.2.1 Initialization

After a Reset, the default clock source of the CLK_MAIN clock (GCLK_MAIN) is started and calibrated before the CPU starts running. The GCLK_MAIN clock is selected as the main clock without any prescaler division.

By default, only the necessary clocks are enabled. Refer to 18.5.2.6. Peripheral Clock Masking for more information.

18.5.2.2 Enabling, Disabling, and Resetting

The MCLK module is always enabled and cannot be reset.

18.5.2.3 Selecting the Main Clock Source

Refer to the Generic Clock Controller description for details on how to configure the clock source of the GCLK_MAIN clock.

Refer to the Oscillators Controller (OSCCTRL) description for details on how to configure the clock source of the CLK_DFLLULP clock.

18.5.2.4 Selecting the Synchronous Clock Division Ratio

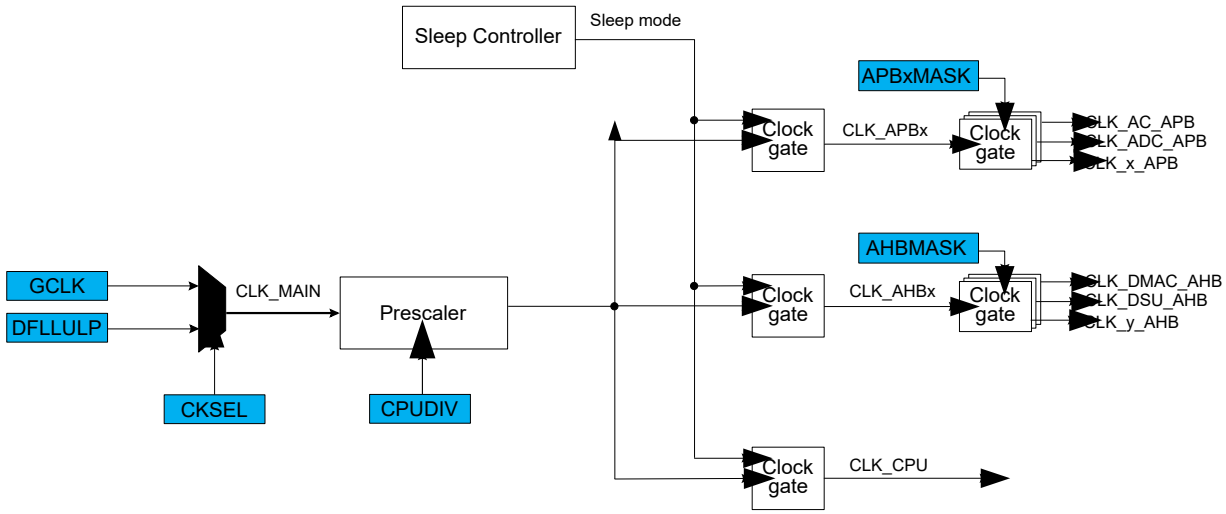
The main clock CLK_MAIN feeds an 8-bit prescaler, which can be used to generate the synchronous clocks. By default, the synchronous clocks run on the undivided main clock. The user can select a prescaler division for the CPU clock domain by writing the CPU Division (CPUDIV) bits in the CPU Clock Division register CPUDIV, resulting in a CPU clock domain frequency determined by this equation:

$$f_{CPU} = \frac{f_{main}}{CPUDIV}$$

If the application attempts to write forbidden values in CPUDIV register, registers are written but these bad values are not used and a violation is reported to the PAC module.

CPU Division bits (CPUDIV) can be written without halting or disabling peripheral modules. Writing CPUDIV bits allows a new clock setting to be written to all synchronous clocks belonging to the corresponding clock domain at the same time.

Figure 18-2. Synchronous Clock Selection and Prescaler



18.5.2.5 Clock Ready Flag

There is a slight delay between writing to CPUDIV until the new clock settings become effective.

During this interval, the Clock Ready flag in the Interrupt Flag Status and Clear register (INTFLAG.CKRDY) will return zero when read. If CKRDY in the INTENSET register is set to '1', the Clock Ready interrupt will be triggered when the new clock setting is effective. The new CPUDIV value must not be re-written while INTFLAG.CKRDY reads '0'. The system may become unstable or hang, and a violation is reported to the PAC module.

18.5.2.6 Peripheral Clock Masking

All peripheral clocks are by default enabled after reset.

⚠ CAUTION Disabling the different peripherals clocks is not required as each peripheral clock is automatically switched off when the peripheral is not accessed. Disabling specific system peripheral clocks, such as NVMCTRL, HMATRIXHS, APB Bridges will even prevent correct device behavior.

18.5.3 Interrupts

The peripheral has the following interrupt sources: Clock Ready (CKRDY), indicates that CPU clocks are ready. 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 (INTFLAG) register is set when the interrupt condition occurs. Each interrupt can be enabled individually by writing a '1' to the corresponding enabling bit in the Interrupt Enable Set (INTENSET) register, and disabled by writing a '1' to the corresponding clearing 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 peripheral is reset. An interrupt flag is cleared by writing a '1' to the corresponding bit in the INTFLAG register. Each peripheral can have one interrupt request line per interrupt source or one common interrupt request line for all the interrupt sources.If the peripheral 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.

18.5.4 Sleep Mode Operation

In Idle sleep mode, the MCLK is still running on the selected main clock.

In Standby sleep mode, the MCLK is frozen if no synchronous clock is required.

18.5.5 Debug Operation

When the CPU is halted in Debug mode, the MCLK continues normal operation. In Sleep mode, the clocks generated from the MCLK are kept running to allow the debugger accessing any module. As a consequence, power measurements are incorrect in Debug mode.

18.6 Register Summary

Refer to the Registers Description section for more details on register properties and access permissions.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRLA	7:0						CKSEL		
0x01	INTENCLR	7:0								CKRDY
0x02	INTENSET	7:0								CKRDY
0x03	INTFLAG	7:0								CKRDY
0x04	CPUDIV	7:0	CPUDIV[7:0]							
0x05 ... 0x0F	Reserved									
0x10	AHBMASK	31:24								
		23:16								
		15:8			USB	TRAM	Reserved	Reserved	Reserved	Reserved
		7:0	NVMCTRL	PAC	HMATRIXHS	DSU	DMAC	APBC	APBB	APBA
0x14	APBAMASK	31:24								
		23:16								
		15:8		Reserved	AC	PORT	FREQM	EIC	RTC	WDT
		7:0	GCLK	SUPC	OSC32KCTRL	OSCCTRL	RSTC	MCLK	PM	PAC
0x18	APBBMASK	31:24								
		23:16								
		15:8								
		7:0			USB	HMATRIXHS		NVMCTRL	DSU	IDAU
0x1C	APBCMASK	31:24								
		23:16				OPAMP	I2S	CCL	TRNG	PTC
		15:8	DAC	ADC	TCC3	TCC2	TCC1	TCC0	TC2	TC1
		7:0	TC0	SERCOM5	SERCOM4	SERCOM3	SERCOM2	SERCOM1	SERCOM0	EVSYS

18.6.1 Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00
Property: PAC Write-Protection

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

Bit 2 – CKSEL Main Clock Select

Value	Description
0	The GCLK_MAIN clock is selected for the main clock.
1	The DFLLULP clock is selected for the main clock.

18.6.2 Interrupt Enable Clear

Name: INTENCLR
Offset: 0x01
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
								CKRDY
Access								R/W
Reset								0

Bit 0 – CKRDY Clock Ready Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Clock Ready Interrupt Enable bit and the corresponding interrupt request.

Value	Description
0	The Clock Ready interrupt is disabled.
1	The Clock Ready interrupt is enabled.

18.6.3 Interrupt Enable Set

Name: INTENSET
Offset: 0x02
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 (INTENCLR) register.

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

Bit 0 – CKRDY Clock Ready Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Clock Ready Interrupt Enable bit and enable the Clock Ready interrupt.

Value	Description
0	The Clock Ready interrupt is disabled.
1	The Clock Ready interrupt is enabled.

18.6.4 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x03
Reset: 0x01
Property: –

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

Bit 0 – CKRDY Clock Ready
This flag is cleared by writing a '1' to the flag.
This flag is set when the synchronous CPU, APBx, and AHBx clocks are stable and will generate an interrupt if INTENSET.CKRDY is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Clock Ready interrupt flag.

18.6.5 CPU Clock Division

Name: CPUDIV
Offset: 0x04
Reset: 0x01
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
	CPUDIV[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 – CPUDIV[7:0] CPU Clock Division Factor
These bits define the division ratio of the main clock prescaler related to the CPU clock domain.
Frequencies must never exceed the specified maximum frequency for each clock domain.

Value	Name	Description
0x01	DIV1	Divide by 1
0x02	DIV2	Divide by 2
0x04	DIV4	Divide by 4
0x08	DIV8	Divide by 8
0x10	DIV16	Divide by 16
0x20	DIV32	Divide by 32
0x40	DIV64	Divide by 64
0x80	DIV128	Divide by 128
others	-	Reserved

18.6.6 AHB Mask

Name: AHBMASK
Offset: 0x10
Reset: 0x000003FFF
Property: PAC Write-Protection

> **⚠ CAUTION** Disabling the different peripherals clocks is not required as each peripheral clock is automatically switched off when the peripheral is not accessed. Disabling specific system peripheral clocks (NVMCTRL, HMATRIXHS, APB Bridges...) will even prevent correct device behavior.

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
			USB	TRAM	Reserved	Reserved	Reserved	Reserved
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			1	1	1	1	1	1

Bit	7	6	5	4	3	2	1	0
	NVMCTRL	PAC	HMATRIXHS	DSU	DMAC	APBC	APBB	APBA
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 13 – USB USB AHB Clock Enable

Value	Description
0	The AHB clock for the USB is stopped
1	The AHB clock for the USB is enabled

Bit 12 – TRAM TRAM AHB Clock Enable

Value	Description
0	The AHB clock for the TRAM is stopped
1	The AHB clock for the TRAM is enabled

Bit 11 – Reserved Must Be Set to 1

Bit 10 – Reserved Must Be Set to 1

Bit 9 – Reserved Must Be Set to 1

Bit 8 – Reserved Must Be Set to 1

Bit 7 – NVMCTRL NVMCTRL AHB Clock Enable

Value	Description
0	The AHB clock for the NVMCTRL is stopped
1	The AHB clock for the NVMCTRL is enabled

Bit 6 – PAC PAC AHB Clock Enable

Value	Description
0	The AHB clock for the PAC is stopped.
1	The AHB clock for the PAC is enabled.

Bit 5 – HMATRIXHS HMATRIXHS AHB Clock Enable

Value	Description
0	The AHB clock for the HMATRIXHS is stopped.
1	The AHB clock for the HMATRIXHS is enabled.

Bit 4 – DSU DSU AHB Clock Enable

Value	Description
0	The AHB clock for the DSU is stopped.
1	The AHB clock for the DSU is enabled.

Bit 3 – DMAC DMAC AHB Clock Enable

Value	Description
0	The AHB clock for the DMAC is stopped.
1	The AHB clock for the DMAC is enabled.

Bit 2 – APBC AHB-APB Bridge C AHB Clock Enable

Value	Description
0	The AHB clock for the APBC is stopped.
1	The AHB clock for the APBC is enabled

Bit 1 – APBB AHB-APB Bridge B AHB Clock Enable

Value	Description
0	The AHB clock for the APBB is stopped.
1	The AHB clock for the APBB is enabled.

Bit 0 – APBA AHB-APB Bridge A AHB Clock Enable

Value	Description
0	The AHB clock for the APBA is stopped.
1	The AHB clock for the APBA is enabled.

18.6.7 APBA Mask

Name: APBAMASK
Offset: 0x14
Reset: 0x000007FFF
Property: PAC Write-Protection

> **⚠ CAUTION** Disabling the different peripherals clocks is not required as each peripheral clock is automatically switched off when the peripheral is not accessed. Disabling specific system peripheral clocks (NVMCTRL, HMATRIXHS, APB Bridges...) will even prevent correct device behavior.

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
		Reserved	AC	PORT	FREQM	EIC	RTC	WDT
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		1	1	1	1	1	1	1

Bit	7	6	5	4	3	2	1	0
	GCLK	SUPC	OSC32KCTRL	OSCCTRL	RSTC	MCLK	PM	PAC
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 14 – Reserved For future use

Reserved bits are unused and reserved for future use. For compatibility with future devices, always write reserved bits to their reset value. If no reset value is given, write 0.

Bit 13 – AC AC APBA Clock Enable

Value	Description
0	The APBA clock for the AC is stopped.
1	The APBA clock for the AC is enabled.

Bit 12 – PORT PORT APBA Clock Enable

Value	Description
0	The APBA clock for the PORT is stopped.
1	The APBA clock for the PORT is enabled.

Bit 11 – FREQM FREQM APBA Clock Enable

Value	Description
0	The APBA clock for the FREQM is stopped.
1	The APBA clock for the FREQM is enabled.

Bit 10 – EIC EIC APBA Clock Enable

Value	Description
0	The APBA clock for the EIC is stopped.

Value	Description
1	The APBA clock for the EIC is enabled.

Bit 9 – RTC RTC APBA Clock Enable

Value	Description
0	The APBA clock for the RTC is stopped.
1	The APBA clock for the RTC is enabled.

Bit 8 – WDT WDT APBA Clock Enable

Value	Description
0	The APBA clock for the WDT is stopped.
1	The APBA clock for the WDT is enabled.

Bit 7 – GCLK GCLK APBA Clock Enable

Value	Description
0	The APBA clock for the GCLK is stopped.
1	The APBA clock for the GCLK is enabled.

Bit 6 – SUPC SUPC APBA Clock Enable

Value	Description
0	The APBA clock for the SUPC is stopped.
1	The APBA clock for the SUPC is enabled.

Bit 5 – OSC32KCTRL OSC32KCTRL APBA Clock Enable

Value	Description
0	The APBA clock for the OSC32KCTRL is stopped.
1	The APBA clock for the OSC32KCTRL is enabled.

Bit 4 – OSCCTRL OSCCTRL APBA Clock Enable

Value	Description
0	The APBA clock for the OSCCTRL is stopped.
1	The APBA clock for the OSCCTRL is enabled.

Bit 3 – RSTC RSTC APBA Clock Enable

Value	Description
0	The APBA clock for the RSTC is stopped.
1	The APBA clock for the RSTC is enabled.

Bit 2 – MCLK MCLK APBA Clock Enable

Value	Description
0	The APBA clock for the MCLK is stopped.
1	The APBA clock for the MCLK is enabled.

Bit 1 – PM PM APBA Clock Enable

Value	Description
0	The APBA clock for the PM is stopped.
1	The APBA clock for the PM is enabled.

Bit 0 – PAC PAC APBA Clock Enable

Value	Description
0	The APBA clock for the PAC is stopped.
1	The APBA clock for the PAC is enabled.

18.6.8 APBB Mask

Name: APBBMASK
Offset: 0x18
Reset: 0x00000037
Property: PAC Write-Protection

> ⚠ **CAUTION** Disabling the different peripherals clocks is not required as each peripheral clock is automatically switched off when the peripheral is not accessed. Disabling specific system peripheral clocks (NVMCTRL, HMATRIXHS, APB Bridges...) will even prevent correct device behavior.

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
			USB	HMATRIXHS		NVMCTRL	DSU	IDAU
Access			R/W	R/W		R/W	R/W	R/W
Reset			1	1		1	1	1

Bit 5 – USB USB APBB Clock Enable

Value	Description
0	The APBB clock for the USB is stopped
1	The APBB clock for the USB is enabled

Bit 4 – HMATRIXHS HMATRIXHS APBB Clock Enable

Value	Description
0	The APBB clock for the HMATRIXHS is stopped
1	The APBB clock for the HMATRIXHS is enabled

Bit 2 – NVMCTRL NVMCTRL APBB Clock Enable

Value	Description
0	The APBB clock for the NVMCTRL is stopped
1	The APBB clock for the NVMCTRL is enabled

Bit 1 – DSU DSU APBB Clock Enable

Value	Description
0	The APBB clock for the DSU is stopped
1	The APBB clock for the DSU is enabled

Bit 0 – IDAU IDAU APBB Clock Enable

Note: This bit field is only available for **PIC32CM LS00/LS60** and has no effect for **PIC32CM LE00**.

Value	Description
0	The APBB clock for the IDAU is stopped
1	The APBB clock for the IDAU is enabled

18.6.9 APBC Mask

Name: APBCMASK
Offset: 0x1C
Reset: 0x001FFFFF
Property: PAC Write-Protection

> **⚠ CAUTION** Disabling the different peripherals clocks is not required as each peripheral clock is automatically switched off when the peripheral is not accessed. Disabling specific system peripheral clocks (NVMCTRL, HMATRIXHS, APB Bridges...) will even prevent correct device behavior.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
				OPAMP	I2S	CCL	TRNG	PTC
Access				R/W	R	R	R	R
Reset				1	1	1	1	1

Bit	15	14	13	12	11	10	9	8
	DAC	ADC	TCC3	TCC2	TCC1	TCC0	TC2	TC1
Access	R	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
	TC0	SERCOM5	SERCOM4	SERCOM3	SERCOM2	SERCOM1	SERCOM0	EVSYS
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 20 – OPAMP OPAMP APBC Clock Enable

Value	Description
0	The APBC clock for the OPAMP is stopped.
1	The APBC clock for the OPAMP is enabled.

Bit 19 – I2S I2S APBC Mask Clock Enable

Value	Description
0	The APBC clock for the I2S is stopped.
1	The APBC clock for the I2S is enabled.

Bit 18 – CCL CCL APBC Mask Clock Enable

Value	Description
0	The APBC clock for the CCL is stopped.
1	The APBC clock for the CCL is enabled.

Bit 17 – TRNG TRNG APBC Mask Clock Enable

Value	Description
0	The APBC clock for the TRNG is stopped.
1	The APBC clock for the TRNG is enabled.

Bit 16 – PTC PTC APBC Mask Clock Enable

Value	Description
0	The APBC clock for the PTC is stopped.
1	The APBC clock for the PTC is enabled.

Bit 15 – DAC DAC APBC Mask Clock Enable

Value	Description
0	The APBC clock for the DAC is stopped.
1	The APBC clock for the DAC is enabled.

Bit 14 – ADC ADC APBC Mask Clock Enable

Value	Description
0	The APBC clock for the ADC is stopped.
1	The APBC clock for the ADC is enabled.

Bit 13 – TCC3 TCC3 APBC Mask Clock Enable

Value	Description
0	The APBC clock for the TCC3 is stopped.
1	The APBC clock for the TCC3 is enabled.

Bit 12 – TCC2 TCC2 APBC Mask Clock Enable

Value	Description
0	The APBC clock for the TCC2 is stopped.
1	The APBC clock for the TCC2 is enabled.

Bit 11 – TCC1 TCC1 APBC Mask Clock Enable

Value	Description
0	The APBC clock for the TCC1 is stopped.
1	The APBC clock for the TCC1 is enabled.

Bit 10 – TCC0 TCC0 APBC Mask Clock Enable

Value	Description
0	The APBC clock for the TCC0 is stopped.
1	The APBC clock for the TCC0 is enabled.

Bit 9 – TC2 TC2 APBC Mask Clock Enable

Value	Description
0	The APBC clock for the TC2 is stopped.
1	The APBC clock for the TC2 is enabled.

Bit 8 – TC1 TC1 APBC Mask Clock Enable

Value	Description
0	The APBC clock for the TC1 is stopped.
1	The APBC clock for the TC1 is enabled.

Bit 7 – TC0 TC0 APBC Mask Clock Enable

Value	Description
0	The APBC clock for the TC0 is stopped.
1	The APBC clock for the TC0 is enabled.

Bit 6 – SERCOM5 SERCOM5 APBC Mask Clock Enable

Value	Description
0	The APBC clock for the SERCOM5 is stopped.
1	The APBC clock for the SERCOM5 is enabled.

Bit 5 – SERCOM4 SERCOM4 APBC Mask Clock Enable

Value	Description
0	The APBC clock for the SERCOM4 is stopped.
1	The APBC clock for the SERCOM4 is enabled.

Bit 4 – SERCOM3 SERCOM3 APBC Mask Clock Enable

Value	Description
0	The APBC clock for the SERCOM3 is stopped.
1	The APBC clock for the SERCOM3 is enabled.

Bit 3 – SERCOM2 SERCOM2 APBC Mask Clock Enable

Value	Description
0	The APBC clock for the SERCOM2 is stopped.
1	The APBC clock for the SERCOM2 is enabled.

Bit 2 – SERCOM1 SERCOM1 APBC Mask Clock Enable

Value	Description
0	The APBC clock for the SERCOM1 is stopped.
1	The APBC clock for the SERCOM1 is enabled.

Bit 1 – SERCOM0 SERCOM0 APBC Mask Clock Enable

Value	Description
0	The APBC clock for the SERCOM0 is stopped.
1	The APBC clock for the SERCOM0 is enabled.

Bit 0 – EVSYS EVSYS APBC Clock Enable

Value	Description
0	The APBC clock for the EVSYS is stopped.
1	The APBC clock for the EVSYS is enabled.

19. 32.768 kHz Oscillators Controller (OSC32KCTRL)

19.1 Overview

The 32.768 kHz Oscillators Controller (OSC32KCTRL) provides a user interface to the 32.768 kHz oscillators:

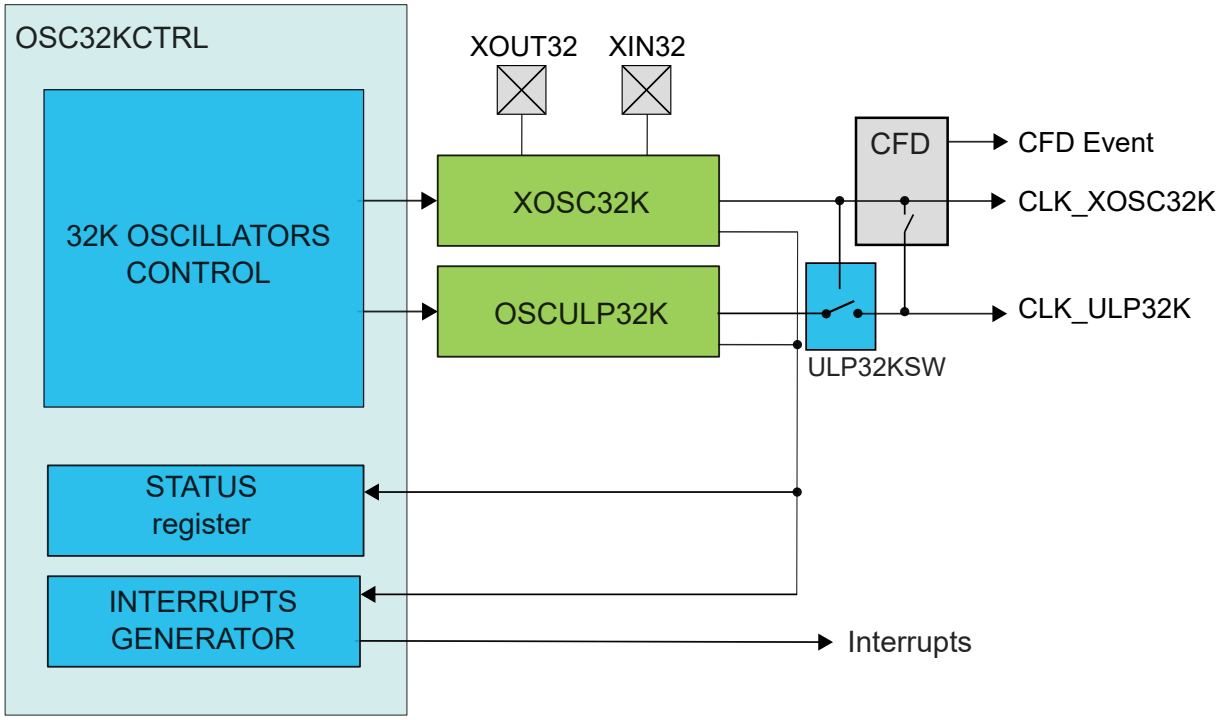
- A 32.768 kHz crystal oscillator (XOSC32K)
- A 32.768 kHz ultra low-power internal RC oscillator (OSCULP32K)

19.2 Features

- 32.768 kHz Crystal Oscillator (XOSC32K)
  - Programmable start-up time
  - Crystal or external input clock on XIN32 I/O
  - Clock failure detection with safe clock switch
  - Clock failure event output
- 32.768 kHz Ultra Low-Power Internal Oscillator (OSCULP32K)
  - Ultra low-power, always-on oscillator
  - Frequency fine tuning
- 1024 Hz clock outputs available

19.3 Block Diagram

Figure 19-1. OSC32KCTRL Block Diagram



19.4 Signal Description

Signal	Description	Type
XIN32	Analog Input	32.768 kHz Crystal Oscillator or external clock input
XOUT32	Analog Output	32.768 kHz Crystal Oscillator output

The I/O lines are automatically selected when XOSC32K is enabled.

⚠ CAUTION The transition time of the following pins must be greater than 50µs in order to not affect the XOSC32 cycle to cycle jitter.

Table 19-1. XOSC32K Jitter Minimization

Package	Pin Name
TQFP48 / VQFN48	PA02, PB03
TQFP64 / VQFN64	PA02, PB03
TQFP100	PC00, PB03

19.5 Peripheral Dependencies

Table 19-2. OSC32KCTRL Configuration Summary

Peripheral name	Base address	NVIC IRQ Index	MCLK AHB/APB Clocks	PAC Peripheral Identifier Index (PAC.WRCTRL)	Events (EVSYS)		Power Domain (PM.STDBYCFG)
					Users (EVSYS.USER)	Generators (EVSYS.CHANNEL)	
OSC32KCTRL	0x40001400	0: XOSC32KRDY, CLKFAIL	CLK_OSC32KCTRL_APB	5	—	2: CFD	PDAO

19.6 Functional Description

19.6.1 Principle of Operation

XOSC32K and OSCULP32K are configured through OSC32KCTRL control registers. Through this interface, the sub-peripherals are enabled, disabled.

The STATUS register gathers different status signals coming from the sub-peripherals of OSC32KCTRL. The status signals can be used to generate system interrupts, and in some cases wake up the system from standby mode, provided the corresponding interrupt is enabled.

19.6.2 32.768 kHz External Crystal Oscillator (XOSC32K) Operation

The XOSC32K can operate in the following two modes:

- External clock, with an external clock signal connected to XIN32
- Crystal oscillator, with an external 32.768 kHz crystal connected between XIN32 and XOUT32

At reset, the XOSC32K is disabled, and the XIN32/XOUT32 pins can either be used as General Purpose I/O (GPIO) pins or by other peripherals in the system.

When XOSC32K is enabled, the operating mode determines the GPIO usage. When in crystal oscillator mode, the XIN32 and XOUT32 pins are controlled by the OSC32KCTRL, and GPIO functions are overridden on both pins. When in external clock mode, the only XIN32 pin will be overridden and controlled by the OSC32KCTRL, while the XOUT32 pin can still be used as a GPIO pin.

The XOSC32K is enabled by writing a '1' to the Enable bit in the 32.768 kHz External Crystal Oscillator Control register (XOSC32K.ENABLE=1). The XOSC32K is disabled by writing a '0' to the Enable bit in the 32.768 kHz External Crystal Oscillator Control register (XOSC32K.ENABLE=0).

To enable the XOSC32K as a crystal oscillator, the XTALEN bit in the 32.768 kHz External Crystal Oscillator Control register must be set (XOSC32K.XTALEN=1). If XOSC32K.XTALEN is '0', the external clock input will be enabled.

Notes: Disabling the XOSC32K when the crystal oscillator is enabled (XOSC32K.ENABLE = 1) must be done as follows:

1. Disable the XOSC32K (XOSC32K.ENABLE = 0).
2. Disable the Crystal Oscillator (XOSC32K.XTALEN = 0).

The XOSC32K 32.768 kHz output is enabled by setting the 32.768 kHz Output Enable bit in the 32.768 kHz External Crystal Oscillator Control register (XOSC32K.EN32K=1). The XOSC32K also has a 1024 Hz clock output, which can only be used by the RTC. This clock output is enabled by setting the 1024 Hz Output Enable bit in the 32.768 kHz External Crystal Oscillator Control register (XOSC32K.EN1K=1).

It is also possible to lock the XOSC32K configuration by setting the Write Lock bit in the 32.768 kHz External Crystal Oscillator Control register (XOSC32K.WRTLOCK=1). If set, the XOSC32K configuration is locked until a Power-On Reset (POR) is detected.

The XOSC32K will behave differently in different sleep modes based on the settings of XOSC32K.RUNSTDBY, XOSC32K.ONDEMAND, and XOSC32K.ENABLE. If XOSC32K.ENABLE = 0, the XOSC32K will be always stopped. For XOSC32K.ENABLE = 1, this table is valid:

Table 19-3. XOSC32K Sleep Behavior

CPU Mode	XOSC32K. RUNSTDBY	XOSC32K. ONDEMAND	Sleep Behavior
Active or Idle	-	0	Always run
Active or Idle	-	1	Run if requested by peripheral
Standby	1	0	Always run
Standby	1	1	Run if requested by peripheral
Standby	0	-	Run if requested by peripheral

As a crystal oscillator usually requires a long start-up time, the 32.768 kHz External Crystal Oscillator will keep running across resets when XOSC32K.ONDEMAND = 0, except for Power-on Reset (POR).

After a reset or when waking up from a sleep mode where the XOSC32K was disabled, the XOSC32K will need a certain amount of time to stabilize on the correct frequency. This stabilization time can be configured by changing the Oscillator Start-Up Time bit group (XOSC32K.STARTUP) in the 32.768 kHz External Crystal Oscillator Control register. During this time, the oscillator output is masked to ensure that no unstable clock propagates to the digital logic.

Once the external clock or crystal oscillator is stable and ready to be used as a clock source, the XOSC32K Ready bit in the Status register is set (STATUS.XOSC32KRDY=1). The transition of STATUS.XOSC32KRDY from '0' to '1' generates an interrupt if the XOSC32K Ready bit in the Interrupt Enable Set register is set (INTENSET.XOSC32KRDY = 1).

Important: The right XOSC32K.STARTUP stabilization time must be selected by considering the Crystal Start-up Time parameter given in the XOSC32K Electrical Specifications section of the Electrical Characteristics chapter.

The XOSC32K can be used as a source for Generic Clock Generators (GCLK) or for the Real-Time Counter (RTC). Before enabling the GCLK or the RTC module, the corresponding oscillator output must be enabled (XOSC32K.EN32K or XOSC32K.EN1K) in order to ensure proper operation. In the same way, the GCLK or RTC

modules must be disabled before the clock selection is changed. For details on RTC clock configuration, refer also to 19.6.6. Real-Time Counter Clock Selection.

19.6.3 Clock Failure Detection Operation

The Clock Failure Detector (CFD) allows the user to monitor the external clock or crystal oscillator signal provided by the external oscillator (XOSC32K). The CFD detects failing operation of the XOSC32K clock with reduced latency, and allows to switch to a safe clock source in case of clock failure. The user can also switch from the safe clock back to XOSC32K in case of recovery. The safe clock is derived from the OSCULP32K oscillator with a configurable prescaler. This allows to configure the safe clock in order to fulfill the operative conditions of the microcontroller.

In sleep modes, CFD operation is automatically disabled when the external oscillator is not requested to run by a peripheral, see “XOSC32K Sleep Behavior” for additional information.

The user interface registers allow to enable, disable, and configure the CFD. The Status register provides status flags on failure and clock switch conditions. The CFD can optionally trigger an interrupt or an event when a failure is detected.

Clock Failure Detection

The CFD is reset only at power-on (POR). The CFD does not monitor the XOSC32K clock when the oscillator is disabled (XOSC32K.ENABLE=0).

Before starting CFD operation, the user must start and enable the safe clock source (OSCULP32K oscillator).

CFD operation is started by writing a '1' to the CFD Enable bit in the External Oscillator Control register (CFDCTRL.CFDEN).

Notes: Disabling then re-enabling the Clock Failure Detector must be done as follows:

1. Disable the XOSC32K (XOSC32K.ENABLE = 0).
2. Disable the Clock Failure Detector (CFDCTRL.CFDEN = 0).
3. Re-enable the Clock Failure Detector (CFDCTRL.CFDEN = 1).
4. Re-enable the XOSC32K (XOSC32K.ENABLE = 1).

After starting or restarting the XOSC32K, the CFD does not detect failure until the crystal oscillator stabilization time has elapsed (XOSC32K.STARTUP). Once this stabilization time is elapsed, the XOSC32K clock is constantly monitored.

Important: The right XOSC32K.STARTUP stabilization time must be selected by considering the Crystal Start-up Time parameter given in the XOSC32K Electrical Specifications section of the Electrical Characteristics chapter.

During a period of 4 safe clocks (monitor period), the CFD watches for a clock activity from the XOSC32K. There must be at least one rising and one falling XOSC32K clock edge during 4 safe clock periods to meet non-failure conditions. If no or insufficient activity is detected, the failure status is asserted: The Clock Failure Detector status bit in the Status register (STATUS.CLKFAIL) and the Clock Failure Detector interrupt flag bit in the Interrupt Flag register (INTFLAG.CLKFAIL) are set. If the CLKFAIL bit in the Interrupt Enable Set register (INTENSET.CLKFAIL) is set, an interrupt is generated as well. If the Event Output enable bit in the Event Control register (EVCTRL.CFDEO) is set, an output event is generated, too.

After a clock failure was issued the monitoring of the XOSC32K clock is continued, and the Clock Failure Detector status bit in the Status register (STATUS.CLKFAIL) reflects the current XOSC32K activity.

Clock Switch

When a clock failure is detected, the XOSC32K clock (CLK_XOSC32K) is replaced by the safe clock in order to maintain an active clock during the XOSC32K clock failure. The safe clock source is the OSCULP32K oscillator clock (CLK_ULP32K).

Both 32.768 kHz and 1024 Hz outputs of the XOSC32K are replaced by the respective OSCULP32K 32.768 kHz and 1024 Hz outputs. The safe clock source can be scaled down by a configurable prescaler to ensure that the safe clock frequency does not exceed the operating conditions selected by the application. When the XOSC32K clock is switched to the safe clock, the Clock Switch bit in the Status register (STATUS.CLKSW) is set.

When the CFD has switched to the safe clock, the XOSC32K is not disabled. If desired, the application must take the necessary actions to disable the oscillator. The application must also take the necessary actions to configure the system clocks to continue normal operations. In the case the application can recover the XOSC32K, the application can switch back to the XOSC32K clock by writing a '1' to the Clock Switch Back Enable bit in the Clock Failure Control register (CFDCTRL.SWBACK). Once the XOSC32K clock is switched back, the Clock Switch Back Enable bit (CFDCTRL.SWBACK) is cleared by hardware.

Prescaler
The CFD has an internal configurable prescaler to generate the safe clock from the OSCULP32K oscillator. The prescaler size allows to scale down the OSCULP32K oscillator so the safe clock frequency is not higher than the XOSC32K clock frequency monitored by the CFD. The maximum division factor is 2.

The prescaler is applied on both outputs (32.768 kHz and 1024 Hz) of the safe clock.

> **Example 19-1.**
>
> For an external crystal oscillator at 32.768 kHz and the OSCULP32K frequency is 32.768 kHz, the XOSC32K.CFDPRESC should be set to 0 for a safe clock of equal frequency.

Event
If the Event Output Enable bit in the Event Control register (EVCTRL.CFDEO) is set, the CFD clock failure will be output on the Event Output. When the CFD is switched to the safe clock, the CFD clock failure will not be output on the Event Output.

Sleep Mode
The CFD is halted depending on configuration of the XOSC32K and the peripheral clock request. For further details, refer to the Sleep Behavior table above. The CFD interrupt can be used to wake up the device from sleep modes.

19.6.4 32.768 kHz Ultra Low-Power Internal Oscillator (OSCULP32K) Operation

The OSCULP32K provides a tunable, low-speed, and ultra low-power clock source.
Note: The OSCULP32K is factory-calibrated under typical voltage and temperature conditions.

The OSCULP32K is enabled by default after a Power-on Reset (POR), and will always run except during POR.

Users can lock the OSCULP32K configuration by setting the Write Lock bit in the 32.768 kHz Ultra Low-Power Internal Oscillator Control register (OSCULP32K.WRTLOCK = 1). If set, the OSCULP32K configuration is locked until POR is detected.

The OSCULP32K can be used as a source for Generic Clock Generators (GCLK) or for the Real-Time Counter (RTC). To ensure proper operation, the GCLK or RTC modules must be disabled before the clock selection is changed.

OSCULP32K Clock Switch
The Clock switch operation requires the XOSC32K to be enabled (XOSC32K.ENABLE=1 and STATUS.XOSC32KRDY = 1). When the OSCULP32K Clock Switch Enable bit (OSCULP32K.ULP32KSW) is set, the CLK_ULP32K clock is switched to the XOSC32K Clock Oscillator. When the clock switch process is complete, the OSCULP32K Clock Switch bit in Status register (STATUS.ULP32KSW) is set. The OSCULP32K oscillator is shut off, and the XOSC32K oscillator becomes always running. The CFD feature is also disabled by hardware. When set, the OSCULP32K.ULP32KSW can be reset only by POR operation.

19.6.5 Watchdog Timer Clock Selection

The Watchdog Timer (WDT) uses the internal 1024 kHz OSCULP32K output clock. This clock is running all the time and internally enabled when requested by the WDT module.

19.6.6 Real-Time Counter Clock Selection

Before enabling the RTC module, the RTC clock must be selected first. All oscillator outputs are valid as RTC clock. The selection is done in the RTC Control register (RTCCTRL). To ensure a proper operation, it is highly recommended to disable the RTC module first, before the RTC clock source selection is changed.

19.6.7 Interrupts

The OSC32KCTRL has the following interrupt sources:

- XOSC32KRDY - 32.768 kHz Crystal Oscillator Ready: A 0-to-1 transition on the STATUS.XOSC32KRDY bit is detected
- CLKFAIL - Clock Failure Detector: A 0-to-1 transition on the STATUS.CLKFAIL bit is detected

All these interrupts are synchronous wake-up source.

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 enabled individually by setting the corresponding bit in the Interrupt Enable Set register (INTENSET), and disabled by setting 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 the interrupt flag is cleared, the interrupt is disabled or the OSC32KCTRL is reset. See the INTFLAG register for details on how to clear interrupt flags.

The OSC32KCTRL 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. Refer to the INTFLAG register for details.

Note: Interrupts must be globally enabled for interrupt requests to be generated.

19.6.8 Events

The CFD can generate the following output event:

- Clock Failure Detector (CFD): Generated when the Clock Failure Detector status bit is set in the Status register (STATUS.CLKFAIL). The CFD event is not generated when the Clock Switch Back Enable bit (STATUS.SWBACK) in the Status register is set.

Writing a '1' to an Event Output bit in the Event Control register (EVCTRL.CFDEO) enables the CFD output event. Writing a '0' to this bit disables the CFD output event. Refer to the Event System chapter for details on configuring the event system.

19.6.9 Debug Operation

When the CPU is halted in debug mode, OSC32KCTRL will continue normal operation. If OSC32KCTRL 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.

19.7 Register Summary

Refer to the Registers Description section for more details on register properties and access permissions.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	INTENCLR	31:24								
		23:16								
		15:8								
		7:0						CLKFAIL		XOSC32KRDY
0x04	INTENSET	31:24								
		23:16								
		15:8								
		7:0						CLKFAIL		XOSC32KRDY
0x08	INTFLAG	31:24								
		23:16								
		15:8								
		7:0						CLKFAIL		XOSC32KRDY
0x0C	STATUS	31:24								
		23:16								
		15:8								
		7:0				ULP32KSW	CLKSW	CLKFAIL		XOSC32KRDY
0x10	RTCCTRL	7:0						RTCSEL[2:0]		
0x11 ... 0x13	Reserved									
0x14	XOSC32K	15:8				WRTLOCK		STARTUP[2:0]		
		7:0	ONDEMAND	RUNSTDBY		EN1K	EN32K	XTALEN	ENABLE	
0x16	CFDCTRL	7:0						CFDPRESC	SWBACK	CFDEN
0x17	EVCTRL	7:0								CFDEO
0x18 ... 0x1B	Reserved									
0x1C	OSCULP32K	31:24								
		23:16								
		15:8	WRTLOCK			Reserved[4:0]				
		7:0			ULP32KSW					

19.7.1 Interrupt Enable Clear

Name: INTENCLR
Offset: 0x00
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
						CLKFAIL		XOSC32KRDY
Access						R/W		R/W
Reset						0		0

Bit 2 – CLKFAIL XOSC32K Clock Failure Detection Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the XOSC32K Clock Failure Interrupt Enable bit, which disables the XOSC32K Clock Failure interrupt.

Value	Description
0	The XOSC32K Clock Failure Detection is disabled.
1	The XOSC32K Clock Failure Detection is enabled. An interrupt request will be generated when the XOSC32K Clock Failure Detection interrupt flag is set.

Bit 0 – XOSC32KRDY XOSC32K Ready Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the XOSC32K Ready Interrupt Enable bit, which disables the XOSC32K Ready interrupt.

Value	Description
0	The XOSC32K Ready interrupt is disabled.
1	The XOSC32K Ready interrupt is enabled.

19.7.2 Interrupt Enable Set

Name: INTENSET
Offset: 0x04
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 Clear register (INTENCLR).

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
						CLKFAIL		XOSC32KRDY
Access						R/W		R/W
Reset						0		0

Bit 2 – CLKFAIL XOSC32K Clock Failure Detection Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the XOSC32K Clock Failure Interrupt Enable bit, which enables the XOSC32K Clock Failure interrupt.

Value	Description
0	The XOSC32K Clock Failure Detection is disabled.
1	The XOSC32K Clock Failure Detection is enabled. An interrupt request will be generated when the XOSC32K Clock Failure Detection interrupt flag is set.

Bit 0 – XOSC32KRDY XOSC32K Ready Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the XOSC32K Ready Interrupt Enable bit, which enables the XOSC32K Ready interrupt.

Value	Description
0	The XOSC32K Ready interrupt is disabled.
1	The XOSC32K Ready interrupt is enabled.

19.7.3 Interrupt Flag Status and Clear

Name: INTFLAG
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
						CLKFAIL		XOSC32KRDY
Access						R/W		R/W
Reset						0		0

Bit 2 – CLKFAIL XOSC32K Clock Failure Detection
This flag is cleared by writing a '1' to it.
This flag is set on a zero-to-one transition of the XOSC32K Clock Failure Detection bit in the Status register (STATUS.CLKFAIL) and will generate an interrupt request if INTENSET.CLKFAIL is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the XOSC32K Clock Failure Detection flag.

Bit 0 – XOSC32KRDY XOSC32K Ready
This flag is cleared by writing a '1' to it.
This flag is set by a zero-to-one transition of the XOSC32K Ready bit in the Status register (STATUS.XOSC32KRDY), and will generate an interrupt request if INTENSET.XOSC32KRDY=1.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the XOSC32K Ready interrupt flag.

19.7.4 Status

Name: STATUS
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
				ULP32KSW	CLKSW	CLKFAIL		XOSC32KRDY
Access				R	R	R		R
Reset				0	0	0		0

Bit 4 – ULP32KSW OSCULP32K Clock Switch

Value	Description
0	CLK_ULP32K clock switch to CLK_XOSC32K is not complete.
1	CLK_ULP32K clock switch to CLK_XOSC32K is complete.

Bit 3 – CLKSW XOSC32K Clock Switch

Value	Description
0	CLK_XOSC32K clock switch to CLK_ULP32K clock is not complete.
1	CLK_XOSC32K clock switch to CLK_ULP32K clock is complete.

Bit 2 – CLKFAIL XOSC32K Clock Failure Detector

Value	Description
0	No XOSC32K failure is detected.
1	An XOSC32K failure is detected.

Bit 0 – XOSC32KRDY XOSC32K Ready

Value	Description
0	XOSC32K is not ready.
1	XOSC32K is stable and ready to be used as a clock source.

19.7.5 RTC Clock Selection Control

Name: RTCCTRL
Offset: 0x10
Reset: 0x00
Property: PAC Write-Protection

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

Bits 2:0 – RTCSEL[2:0] RTC Clock Selection
These bits select the source for the RTC.

Value	Name	Description
0x0	ULP1K	1024 Hz from the OSCULP32K low-power internal RC oscillator
0x1	ULP32K	32.768 kHz from the OSCULP32K low-power internal RC oscillator
0x2-0x3	Reserved	-
0x4	XOSC1K	1024 Hz from the XOSC32K crystal oscillator
0x5	XOSC32K	32.768 kHz from the XOSC32K crystal oscillator
0x6-0x7	Reserved	-

19.7.6 32.768 kHz External Crystal Oscillator (XOSC32K) Control

Name: XOSC32K
Offset: 0x14
Reset: 0x00000080
Property: PAC Write-Protection

Bit	15	14	13	12	11	10	9	8
				WRTLOCK		STARTUP[2:0]		
Access				R/W		R/W	R/W	R/W
Reset				0		0	0	0

Bit	7	6	5	4	3	2	1	0
	ONDEMAND	RUNSTDBY		EN1K	EN32K	XTALEN	ENABLE	
Access	R/W	R/W		R/W	R/W	R/W	R/W	
Reset	1	0		0	0	0	0	

Bit 12 – WRTLOCK Write Lock
This bit locks the XOSC32K register for future writes, effectively freezing the XOSC32K configuration.

Value	Description
0	The XOSC32K configuration is not locked.
1	The XOSC32K configuration is locked.

Bits 10:8 – STARTUP[2:0] Oscillator Start-Up Time
This bit field selects the XOSC32K crystal oscillator stabilization time.

Important: This stabilization time is for guidance only. A major component of crystal start-up time is based on the second party crystal MFG parasitics that are outside the scope of this specification. If this is a major concern, the customer would need to characterize this based on their design choices.

Table 19-4. Stabilization Time for 32.768 kHz External Crystal Oscillator (1)

Value	Number of OSCULP32K Clock Cycles	Approximate Equivalent Time (2) (s)
0x0	2048	0.06
0x1	4096	0.13
0x2	16384	0.5
0x3	32768	1
0x4	65536	2
0x5	131072	4
0x6	262144	8
0x7	Reserved	-

Notes:
1. The OSCULP32K oscillator is used to clock the start-up counter.
2. Actual Start-Up time is the number of selected OSCULP32K cycles + 3 XOSC32K cycles.

Bit 7 – ONDEMAND On Demand Control
This bit controls how the XOSC32K behaves when a peripheral clock request is detected. For details, refer to Table 19-3.

Bit 6 – RUNSTDBY Run in Standby
This bit controls how the XOSC32K behaves during standby sleep mode. For details, refer to Table 19-3.

Bit 4 – EN1K 1024 Hz Output Enable

Value	Description
0	The 1024 Hz output is disabled.
1	The 1024 Hz output is enabled.

Bit 3 – EN32K 32.768 kHz Output Enable

Value	Description
0	The 32.768 kHz output is disabled.
1	The 32.768 kHz output is enabled.

Bit 2 – XTALEN Crystal Oscillator Enable

This bit controls the connections between the I/O pads and the external clock or crystal oscillator.

Value	Description
0	External clock connected on XIN32. XOUT32 can be used as general-purpose I/O.
1	Crystal connected to XIN32/XOUT32.

Bit 1 – ENABLE Oscillator Enable

Value	Description
0	The oscillator is disabled.
1	The oscillator is enabled.

19.7.7 Clock Failure Detector Control

Name: CFDCTRL
Offset: 0x16
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
						CFDPRESC	SWBACK	CFDEN
Access						R/W	R/W	R/W
Reset						0	0	0

Bit 2 – CFDPRESC Clock Failure Detector Prescaler
This bit selects the prescaler for the Clock Failure Detector.

Value	Description
0	The CFD safe clock frequency is the OSCULP32K frequency
1	The CFD safe clock frequency is the OSCULP32K frequency divided by 2

Bit 1 – SWBACK Clock Switch Back Enable
This bit controls the XOSC32K output switch back to the external clock or crystal oscillator in case of clock recovery.

Value	Description
0	The clock switch back is disabled.
1	The clock switch back is enabled. This bit is reset when the XOSC32K output is switched back to the external clock or crystal oscillator.

Bit 0 – CFDEN Clock Failure Detector Enable
This bit selects the Clock Failure Detector state.

Value	Description
0	The CFD is disabled.
1	The CFD is enabled.

19.7.8 Event Control

Name: EVCTRL
Offset: 0x17
Reset: 0x00
Property: PAC Write-Protection

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

Bit 0 – CFDEO Clock Failure Detector Event Out Enable
This bit controls whether the Clock Failure Detector event output is enabled and an event will be generated when the CFD detects a clock failure.

Value	Description
0	Clock Failure Detector Event output is disabled, no event will be generated.
1	Clock Failure Detector Event output is enabled, an event will be generated.

19.7.9 32.768 kHz Ultra Low-Power Internal Oscillator (OSCULP32K) Control

Name: OSCULP32K
Offset: 0x1C
Reset: 0x0000XX00
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
	WRTLOCK			Reserved[4:0]				
Access	R/W			R	R	R	R	R
Reset	0			0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
			ULP32KSW					
Access			R/W					
Reset			0					

Bit 15 – WRTLOCK Write Lock

This bit locks the OSCULP32K register for future writes to fix the OSCULP32K configuration.

Value	Description
0	The OSCULP32K configuration is not locked.
1	The OSCULP32K configuration is locked.

Bits 12:8 – Reserved[4:0]

These bits must not be changed to ensure correct OSCULP32K behavior.

Bit 5 – ULP32KSW OSCULP32K Clock Switch Enable

Value	Description
0	CLK_ULP32K is provided by the OSCULP32K oscillator.
1	CLK_ULP32K is provided by the XOSC32K oscillator.

20. Oscillators Controller (OSCCTRL)

20.1 Overview

The Oscillators Controller (OSCCTRL) provides a user interface to the:

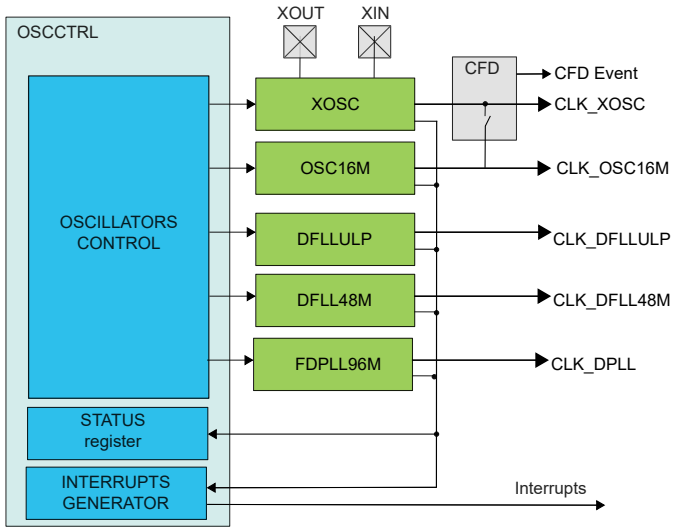
- 0.4 to 32 MHz crystal oscillator (XOSC)
- 16/12/8/4 MHz low-power internal RC oscillator (OSC16M)
- Ultra low-power digital Frequency-Locked Loop (DFLLULP)
- 48 MHz digital Frequency-Locked Loop (DFLL48M)
- 32-96 MHz fractional digital Phase-Locked Loop (FDPLL96M)

20.2 Features

- 0.4 MHz - 32 MHz Crystal Oscillator (XOSC)
  - Tunable gain control
  - Programmable start-up time
  - Crystal or external input clock on XIN I/O
  - Clock failure detection with safe clock switch
  - Clock failure event output
- 16 MHz Internal Oscillator (OSC16M)
  - Fast start-up
  - 4, 8, 12 or 16 MHz output frequencies available
- Ultra Low-Power Digital Frequency Locked Loop (DFLLULP)
  - Operates as a frequency multiplier against a known frequency in Closed-Loop mode
  - Optional frequency dithering
- Digital Frequency Locked Loop (DFLL48M)
  - Internal oscillator with no external components
  - 48 MHz output frequency
  - Operates stand-alone as a high-frequency programmable oscillator in Open-Loop mode
  - Operates as an accurate frequency multiplier against a known frequency in Closed-Loop mode
- Fractional Digital Phase Locked Loop (FDPLL96M)
  - 32 MHz to 96 MHz output frequency
  - 32 kHz to 2 MHz reference clock
  - A selection of sources for the reference clock
  - Adjustable proportional integral controller
  - Fractional part used to achieve 1/16th of reference clock step

20.3 Block Diagram

Figure 20-1. OSCCTRL Block Diagram



20.4 Signal Description

Signal	Description	Type
XIN	Multipurpose Crystal Oscillator or external clock generator input	Analog input
XOUT	Multipurpose Crystal Oscillator output	Analog output

The I/O lines are automatically selected when XOSC is enabled.

20.5 Peripheral Dependencies

Table 20-1. OSCCTRL Configuration Summary

Peripheral name	Base address	NVIC IRQ Index	MCLK AHB/APB Clocks	GCLK Peripheral Channel Index (GCLK.PCHCTRL)	PAC Peripheral Identifier Index (PAC.WRCTRL)	Events (EVSYS)		Power Domain (PM.STDBYCFG)
						Users (EVSYS.USER)	Generators (EVSYS.CHANNEL)	
OSCCTRL	0x40001000	0: XOSCRDY, XOSCFAIL, OSC16MRDY, DFLLULPRDY, DFLLULPLOCK, DFLLULPNOLOCK, DPLLLCKR, DPLLLCKF, DPLLLTO, DPLLLDRTO, DFLLRDY, DFLLOOB, DFLLLCKF, DFLLLCKC, DFLLRCS	CLK_OSCCTRL_APB	0: GCLK_FDPLL96M 1: GCLK_FDPLL96M_32K 2: GCLK_DFLLULP 3: GCLK_DFLL48M	4	0: TUNE	1: CFD	PDSW

20.6 Functional Description

20.6.1 0.4 MHz to 32 MHz Crystal Oscillator (XOSC) Operation

The XOSC can operate in the following two modes:

- External clock, with an external clock signal connected to the XIN pin
- Crystal oscillator, with an external 0.4 MHz - 32 MHz crystal

The XOSC can be used as a clock source for generic clock generators. This is configured by the Generic Clock Controller.

At reset, the XOSC is disabled, and the XIN/XOUT pins can be used as General Purpose I/O (GPIO) pins or by other peripherals in the system. When XOSC is enabled, the operating mode determines the GPIO usage. When in crystal oscillator mode, the XIN and XOUT pins are controlled by the OSCCTRL, and GPIO functions are overridden on both pins. When in external clock mode, only the XIN pin will be overridden and controlled by the OSCCTRL, while the XOUT pin can still be used as a GPIO pin.

The XOSC is enabled by writing a '1' to the Enable bit in the External Multipurpose Crystal Oscillator Control register (XOSCCTRL.ENABLE).

To enable XOSC as an external crystal oscillator, the XTAL Enable bit (XOSCCTRL.XTALEN) must be written to '1'. If XOSCCTRL.XTALEN is zero, the external clock input on XIN will be enabled.

Notes: Disabling the XOSC when the crystal oscillator is enabled (XOSCCTRL.ENABLE = 1) must be done as follows:

1. Disable the XOSC (XOSCCTRL.ENABLE = 0).
2. Disable the Crystal Oscillator (XOSCCTRL.XTALEN = 0).

When in crystal oscillator mode (XOSCCTRL.XTALEN = 1), the External Multipurpose Crystal Oscillator Gain (XOSCCTRL.GAIN) must be set to match the external crystal oscillator frequency. If the External Multipurpose Crystal Oscillator Automatic Amplitude Gain Control (XOSCCTRL.AMPGC) is '1', the oscillator amplitude will be automatically adjusted, and in most cases result in a lower power consumption.

The XOSC will behave differently in different sleep modes, based on the settings of XOSCCTRL.RUNSTDBY, XOSCCTRL.ONDEMAND, and XOSCCTRL.ENABLE. If XOSCCTRL.ENABLE = 0, the XOSC will be always stopped. For XOSCCTRL.ENABLE = 1, this table is valid:

Table 20-2. XOSC Sleep Behavior

CPU Mode	XOSCCTRL.RUNSTDBY	XOSCCTRL.ONDEMAND	Sleep Behavior
Active or Idle	-	0	Always run
Active or Idle	-	1	Run if requested by peripheral
Standby	1	0	Always run
Standby	1	1	Run if requested by peripheral
Standby	0	-	Run if requested by peripheral

After a hard reset, or when waking up from a sleep mode where the XOSC was disabled, the XOSC will need a certain amount of time to stabilize on the correct frequency. This stabilization time can be configured by changing the Oscillator Start-Up Time bit group (XOSCCTRL.STARTUP) in the External Multipurpose Crystal Oscillator Control register. During this time, the oscillator output is masked to ensure that no unstable clock propagates to the digital logic.

The External Multipurpose Crystal Oscillator Ready bit in the Status register (STATUS.XOSCRDY) is set once the external clock or crystal oscillator is stable and ready to be used as a clock source. An interrupt is generated on a zero-to-one transition on STATUS.XOSCRDY if the External Multipurpose Crystal Oscillator Ready bit in the Interrupt Enable Set register (INTENSET.XOSCRDY) is set.

Important: The right XOSCCTRL.STARTUP stabilization time must be selected by considering the Crystal Start-up Time parameter given in the XOSC Electrical Specifications section of the Electrical Characteristics chapter.

20.6.2 Clock Failure Detection Operation

The Clock Failure Detector (CFD) enables the user to monitor the external clock or crystal oscillator signal provided by the external oscillator (XOSC). The CFD detects failing operation of the XOSC clock with reduced latency, and allows to switch to a safe clock source in case of clock failure. The user can also switch from the safe clock back to XOSC in case of recovery. The safe clock is derived from the OSC16M oscillator with a configurable prescaler. This allows to configure the safe clock in order to fulfill the operative conditions of the microcontroller.

In sleep modes, CFD operation is automatically disabled when the external oscillator is not requested to run by a peripheral, for additional information, refer to the “XOSC Sleep Behavior” table.

The user interface registers allow to enable, disable, and configure the CFD. The Status register provides status flags on failure and clock switch conditions. The CFD can optionally trigger an interrupt or an event when a failure is detected.

Clock Failure Detection

The CFD is disabled at reset. The CFD does not monitor the XOSC clock when the oscillator is disabled (XOSCCTRL.ENABLE = 0).

Before starting the CFD operation, the user must start and enable the safe clock source (OSC16M oscillator).

The CFD operation is started by writing a '1' to the CFD Enable bit in the External Oscillator Control register (XOCCTRL.CFDEN).

Notes: Disabling then re-enabling the Clock Failure Detector must be done as follows:

1. Disable the XOSC (XOSCCTRL.ENABLE = 0).
2. Disable the Clock Failure Detector (XOSCCTRL.CFDEN = 0).
3. Re-enable the Clock Failure Detector (XOSCCTRL.CFDEN = 1).
4. Re-enable the XOSC (XOSCCTRL.ENABLE = 1).

After starting or restarting the XOSC, the CFD does not detect failure until the stabilization time has elapsed (XOSCCTRL.STARTUP). Once the XOSC Start-Up Time is elapsed, the XOSC clock is constantly monitored.

Important: The right XOSCCTRL.STARTUP stabilization time must be selected by considering the Crystal Start-up Time parameter given in the XOSC Electrical Specifications section of the Electrical Characteristics chapter.

During a period of 4 safe clocks (monitor period), the CFD watches for a clock activity from the XOSC. There must be at least one rising and one falling XOSC clock edge during 4 safe clock periods to meet non-failure conditions. If no or insufficient activity is detected, the failure status is asserted: The Clock Failure Detector status bit in the Status register (STATUS.XOSCFAIL) and the XOSC Clock Failure Detector interrupt flag bit in the Interrupt Flag register (INTFLAG.XOSCFAIL) are set. If the XOSCFAIL bit in the Interrupt Enable Set register (INTENSET.XOSCFAIL) is set, an interrupt is generated as well. If the Event Output enable bit in the Event Control register (EVCTRL.CFDEO) is set, an output event is generated, too.

After a clock failure was issued the monitoring of the XOSC clock is continued, and the Clock Failure Detector status bit in the Status register (STATUS.XOSCFAIL) reflects the current XOSC activity.

Note: The XOSC Ready bit (STATUS.XOSCRDY) must be ignored when a clock failure is detected: STATUS.XOSCFAIL must always be checked before STATUS.XOSCRDY, and STATUS.XOSCRDY must always be ignored when STATUS.XOSCFAIL=1.

Clock Switch

When a clock failure is detected, the XOSC clock is replaced by the safe clock in order to maintain an active clock during the XOSC clock failure. The safe clock source is the OSC16M oscillator clock. The safe clock source can

be scaled down by a configurable prescaler to ensure that the safe clock frequency does not exceed the operating conditions selected by the application. When the XOSC clock is switched to the safe clock, the Clock Switch bit in the Status register (STATUS.XOSCCKSW) is set.

When the CFD has switched to the safe clock, the XOSC is not disabled. If desired, the application must take the necessary actions to disable the oscillator. The application must also take the necessary actions to configure the system clocks to continue normal operations.

If the application can recover the XOSC, the application can switch back to the XOSC clock by writing a '1' to Switch Back Enable bit in the Clock Failure Control register (XOSCCTRL.SWBEN). Once the XOSC clock is switched back, the Switch Back Enable bit (XOSCCTRL.SWBEN) is cleared by hardware.

Prescaler
The CFD has an internal configurable prescaler to generate the safe clock from the OSC16M oscillator. The prescaler size allows to scale down the OSC16M oscillator so the safe clock frequency is not higher than the XOSC clock frequency monitored by the CFD. The division factor is 2^P, with P being the value of the CFD Prescaler bits in the CFD Prescaler Register (CFDPRESC.CFDPRESC).

Example 20-1.

For an external crystal oscillator at 0.4 MHz and the OSC16M frequency at 16 MHz, the CFDPRESC.CFDPRESC value should be set scale down by more than factor 16/0.4=80, for example 128, for a safe clock of adequate frequency.

Event
If the Event Output Enable bit in the Event Control register (EVCTRL.CFDEO) is set, the CFD clock failure will be output on the Event Output. When the CFD is switched to the safe clock, the CFD clock failure will not be output on the Event Output.

Sleep Mode
The CFD is halted depending on configuration of the XOSC and the peripheral clock request. For further details, refer to the Sleep Behavior table above. The CFD interrupt can be used to wake up the device from sleep modes.

20.6.3 16/12/8/4 MHz Low-Power Internal RC Oscillator (OSC16M) Operation

The OSC16M is an internal oscillator operating in open-loop mode and generating 4, 8, 12, or 16 MHz frequency. The OSC16M frequency is selected by writing to the Frequency Select field in the OSC16M register (OSC16MCTRL.FSEL). OSC16M is enabled by writing '1' to the Oscillator Enable bit in the OSC16M Control register (OSC16MCTRL.ENABLE), and disabled by writing a '0' to this bit. Frequency selection must be done when OSC16M is disabled.

After enabling OSC16M, the OSC16M clock is output as soon as the oscillator is ready (STATUS.OSC16MRDY=1). Users must ensure that the OSC16M is fully disabled before enabling it by reading STATUS.OSC16MRDY=0.

After reset, OSC16M is enabled and serves as the default clock source at 4 MHz.

OSC16M will behave differently in different sleep modes based on the settings of OSC16MCTRL.RUNSTDBY, OSC16MCTRL.ONDEMAND, and OSC16MCTRL.ENABLE. If OSC16MCTRL.ENABLE=0, the OSC16M will be always stopped. For OSC16MCTRL.ENABLE=1, this table is valid:

Table 20-3. OSC16M Sleep Behavior

CPU Mode	OSC16MCTRL.RUNSTDBY	OSC16MCTRL.ONDEMAND	Sleep Behavior
Active or Idle	-	0	Always run
Active or Idle	-	1	Run if requested by peripheral
Standby	1	0	Always run
Standby	1	1	Run if requested by peripheral
Standby	0	-	Run if requested by peripheral

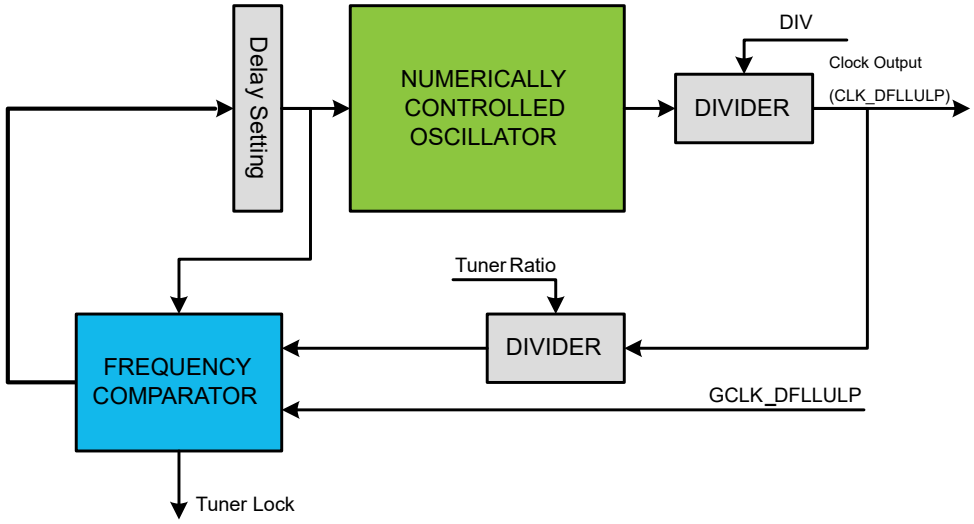
OSC16M is used as a clock source for the generic clock generators. This is configured by the Generic Clock Generator Controller.

20.6.4 Ultra Low-Power Digital Frequency Locked Loop (DFLLULP) Operation

The Ultra Low-Power Digital Frequency Locked Loop (DFLLULP) is an internal oscillator that can output a selectable frequency based on user inputs. The frequency is a multiplication ratio relative to a given reference clock using the tuning feature. The oscillator has to be enabled for the tuner to work.

A generic clock (GCLK_DFLLULP) is required to clock the DFLLULP tuner in closed-loop operation. This clock must be configured and enabled in the generic clock controller before using the DFLLULP tuner.

Figure 20-2. Block Diagram



20.6.4.1 Basic Operation

20.6.4.1.1 Initialization

The following bits are enable-protected, meaning that they can only be written when the DFLLULP is disabled (DFLLULPCTRL.ENABLE is zero):

- Binary Search Enable bit in Control register (DFLLULPCTRL.BINSE)
- Safe Mode bit in Control register (DFLLULPCTRL.SAFE)
- Dither Mode bit in Control register (DFLLULPCTRL.DITHER)
- Division Factor bits in Control register (DFLLULPCTRL.DIV)

The following registers are enable-protected:

- Dither Control register (DFLLULPDITHER)
- Target Ratio register (DFLLULPRATIO)

Enable-protected bits in the DFLLULPCTRL register can be written at the same time as DFLLULPCTRL.ENABLE is written to one, but not at the same time as DFLLULPCTRL.ENABLE is written to zero.

Enable-protection is denoted by the Enable-Protected property in the register description.

20.6.4.1.2 Enabling and Disabling

The DFLLULP is enabled by writing a one to the Enable bit in the Control register (DFLLULPCTRL.ENABLE). The DFLLULP is disabled by writing a zero to DFLLULPCTRL.ENABLE.

20.6.4.1.3 Closed Loop Mode

In closed loop mode the frequency is controlled by a tuner which measures the ratio between the DFLLULP output frequency and reference clock frequency. The reference clock is provided by a generic clock. The target ratio is written to the RATIO field in the Target Ratio Register (DFLLULPRATIO). When the oscillator is enabled, the output frequency will be tuned to the target frequency.

When the tuning is finished, the Lock bit in the OSCCTRL Status Register will be set (STATUS.DFLLULPLOCK). The No Lock bit in the Status register (STATUS.DFLLULPNOLOCK) will be set to indicate whether a frequency lock is achieved or not. The No Lock bit should be checked after the Lock bit has been set. Lock status is cleared when a new value is written to DFLLULPDLY. The DFLLULP will attempt to track any variation in the internal oscillator or reference clock, and will not release the lock. No Lock may be set after lock is achieved if the tuner ever reaches the minimum or maximum delay value.

Tuning starts from the delay value in DFLLULPDLY.DELAY. A write to DFLLULPDLY.DELAY while tuning is in progress will restart tuning from the newly written value. The tuned delay value can be read back from DFLLULPDLY.DELAY after requesting a synchronization via the Read Request bit (RREQ) in the DFLLULPRREQ register.

The accuracy of the tuner frequency comparison is limited by the inverse of the target ratio (1/RATIO). Larger ratios, i.e. much slower reference clocks will give better results.

20.6.4.1.4 Binary Search

By default, the tuner starts from the current value of the DFLLULPDLY register and increments or decrements every reference clock period. This linear search can take up to a maximum of 256 reference clock cycles before lock. To speed up the time to lock, binary search can be enabled by writing one to the Binary Search Enable bit in the Control register (DFLLULPCTRL.BINSE). Binary search takes a maximum of 8 reference clock cycles to lock. After 8 reference clock cycles the tuner will operate in normal linear mode to track any changes in the frequency. Note that neither search algorithm is guaranteed to lock if the target ratio is outside of the oscillator tunable range. Binary search will induce large swings in the oscillator frequency. If this is not desirable, an optional safe mode can be used to mask the output clock until the search is complete. Safe mode is enabled by writing a one to the Safe Mode bit in the Control register (DFLLULPCTRL.SAFE). The binary search is retriggered if there is a write to DFLLULPDLY. On demand or sleep modes will not retrigger the binary search.

20.6.4.1.5 Dithering

Dithering operation can improve the precision of the closed-loop tuner. Dithering works on two aspects: the delay step size and the comparator resolution. Normally the delay can only be changed in steps of one unit of the 8-bit delay field every reference clock period, for a total of 256 steps. Dithering allows for 8-bits of fractional delay value by automatically changing between DELAY and DELAY+1 values with a weight determined by the tuner. This also has the effect of smoothing the frequency over time. Dithering is therefore equivalent to 16-bits of delay value. If this full range is not needed, the step size can be increased by writing the Step Size field in the Dithering register (DFLLULPDITHER.STEP). By default the frequency comparator resolution is limited to 1/RATIO over a single reference clock period. In dithering operation, the comparator resolution can be made finer by comparing over multiple reference clock periods. This behavior is controlled by the Period field in the Dithering register (DFLLULPDITHER.PER). The fine control offered by dithering means that the tuner will take longer to adjust to coarse changes in the frequency. When dithering mode is active, the tuner will attempt to get close to the final locked value before starting the dithering engine. Dithering mode can be restarted if there is a write to DFLLULPDLY

20.6.4.1.6 Event Triggered Tuning

The EVCTRL.TUNEEI and EVCTRL.TUNEINV control bits allow to start a tuning sequence on an incoming event or inverted event. On an incoming rising or falling edge of the event input, the DFLLULP close loop tuner unlock and start over a frequency tuning, depending on the configuration of the DFLLULP registers, until the tuner achieves a new lock.

20.6.5 48 MHz Digital Frequency-Locked Loop (DFLL48M) Operation

The DFLL48M can operate in both open-loop mode and closed-loop mode. In closed-loop mode, a low-frequency clock with high accuracy should be used as the reference clock to get high accuracy on the output clock (CLK_DFLL48M).

The DFLL48M can be used as a source for the generic clock generators.

20.6.5.1 Basic Operation

20.6.5.1.1 Enabling

The DFLL48M is enabled as follows:

1. Disable the On Demand mode (DFLLCTRL.ONDEMAND = 0).
2. Configure the DFLLCTRL register except DFLLCTRL.ENABLE (DFLLCTRL.ENABLE = 0).

3. Wait for STATUS.DFLLRDY to ensure the DFLLCTRL register synchronization is complete.
4. Enable the DFLL48M (DFLLCTRL.ENABLE = 1).
5. Wait for STATUS.DFLLRDY to ensure the DFLLCTRL register synchronization is complete.
6. Enable the On Demand mode if required (DFLLCTRL.ONDEMAND = 1).
7. Wait for STATUS.DFLLRDY to ensure the DFLLCTRL register synchronization is complete.

20.6.5.1.2 Open-Loop Operation

After any reset, the open-loop mode is selected. When operating in open-loop mode, the output frequency of the DFLL48M clock, CLK_DFLL48M, will be determined by the values written to the DFLL Coarse Value bit group and the DFLL Fine Value bit group (DFLLVAL.COARSE and DFLLVAL.FINE) in the DFLL Value register. Using "DFLL48M COARSE" value from the Non Volatile Memory Software Calibration Area in DFLLVAL.COARSE helps to output a frequency close to 48MHz.

It is possible to change the values of DFLLVAL.COARSE and DFLLVAL.FINE while the DFLL48M is enabled and in use, and thereby to adjust the output frequency of CLK_DFLL48M.

20.6.5.1.3 Closed-Loop Operation

In closed-loop operation, the DFLL48M output frequency is continuously regulated against a precise reference clock of relatively low frequency. This will improve the accuracy and stability of the CLK_DFLL48M clock in comparison to the open-loop (free-running) configuration.

Before closed-loop operation can be enabled, the DFLL48M must be enabled and configured in the following way:

1. Enable and select a reference clock (GCLK_DFLL48M).
2. Select the maximum step size allowed for finding the Coarse and Fine values by writing the appropriate values to the DFLL Coarse Maximum Step and DFLL Fine Maximum Step bit groups (DFLLMUL.CSTEP and DFLLMUL.FSTEP) in the DFLL Multiplier register.
   A small step size will ensure low overshoot on the output frequency, but it will typically take longer until locking is achieved. A high value might give a large overshoot, but will typically provide faster locking.

   DFLLMUL.CSTEP and DFLLMUL.FSTEP should not be higher than 50% of the maximum value of DFLLVAL.COARSE and DFLLVAL.FINE, respectively.
3. Select the multiplication factor in the DFLL Multiply Factor bit group (DFLLMUL.MUL) in the DFLL Multiplier register.
   Note: When choosing DFLLMUL.MUL, the output frequency must not exceed the maximum frequency of the device.
   If the target frequency is below the minimum frequency of the DFLL48M, the output frequency will be equal to the DFLL minimum frequency.
4. Start the closed loop mode by writing '1' to the DFLL Mode Selection bit in the DFLL Control register (DFLLCTRL.MODE). See 20.6.5.1.4. Frequency Locking for details.

The frequency of CLK_DFLL48M ($F_{clkdfll48m}$) is given by:

$$F_{\mathrm{clkdfll48m}} = \mathrm{DFLLMUL} \cdot \mathrm{MUL} \times F_{\mathrm{clkdfll48m\_ref}}$$

where $F_{clkdfll48m\_ref}$ is the frequency of the generic clock generator: GCLK_DFLL48M.

20.6.5.1.4 Frequency Locking

After enabling closed-loop operation by writing DFLLCTRL.MODE=1, the Coarse Value and the Fine Value bit fields in the DFLL48M Value register (DFLLVAL.COARSE and DFLLVAL.FINE) are used as starting parameters for the locking procedure.

Note: DFLLVAL.COARSE and DFLLVAL.FINE are read-only in closed-loop mode, and are controlled by the frequency tuner to meet user specified frequency.

The frequency locking is divided into two stages: coarse and fine lock.

Coarse Lock. Starting from the original DFLLVAL.COARSE and DFLLVAL.FINE, the control logic quickly finds the correct value for DFLLVAL.COARSE and sets the output frequency to a value close to the correct frequency. On coarse lock, the DFLL Locked on Coarse Value bit (STATUS.DFLLLCKC) in the Status register will be set.

Fine Lock. In this stage, the control logic tunes the value in DFLLVAL.FINE so that the output frequency is very close to the desired frequency. On fine lock, the DFLL Locked on Fine Value bit (STATUS.DFLLLCKF) in the Status register will be set.

Interrupts are generated by STATUS.DFLLLCKC and STATUS.DFLLLCKF, if INTENSET.DFLLLCKC or INTENSET.DFLLLCKF, respectively, are written to '1'.

The accuracy of the output frequency depends on which locks are set.

Note: Writing DFLLVAL.COARSE to a value close to the final value before entering closed-loop mode will reduce the time needed to get a lock on Coarse.

For a DFLL48M output frequency of 48MHz, the bit field "DFLL48M COARSE CAL" in the NVM Software Calibration Area provides a matching value for DFLL.COARSE, and will start DFLL with a frequency close to 48MHz.

This procedure will reduce the locking time to only the DFLL Fine Lock time:

1. Load the "DFLL48M COARSE CAL" value from the NVM Software Calibration Area into the DFLL.COARSE bit field.
2. Enable the Bypass Coarse Lock (DFLLCTRL.BPLCKC=1).
3. Start DFLL close loop (DFLLCTRL.MODE=1).

Important: Once the DFLL is configured in close loop (DFLLCTRL.MODE=1), it is forbidden to access the "DFLL48M COARSE CAL" value from the NVM Software Calibration Area.

20.6.5.1.5 Frequency Error Measurement

The ratio between GCLK_DFLL48M and CLK_DFLL48M is measured automatically when the DFLL48M is in closed-loop mode. The difference between this ratio and the value in DFLLMUL.MUL is stored in the DFLL Multiplication Ratio Difference bit group (DFLLVAL.DIFF) in the DFLL Value register.

The relative error of CLK_DFLL48M with respect to the target frequency is calculated as follows:

$$ERROR = \frac{\text{DFLLVAL.DIFF}}{\text{DFLLMUL.MUL}}$$

20.6.5.1.6 Drift Compensation

If the Stable DFLL Frequency bit (DFLLCTRL.STABLE) in the DFLL Control register is '0', the frequency tuner will automatically compensate for drift in the CLK_DFLL48M without losing either of the locks.

Note: This means that DFLLVAL.FINE can change after every measurement of CLK_DFLL48M.

The DFLLVAL.FINE value may overflow or underflow in closed-loop mode due to large drift/instability of the clock source reference, and the DFLL Out Of Bounds bit (STATUS.DFLLOOB) in the Status register will be set. After an Out of Bounds error condition, the user must rewrite DFLLMUL.MUL to ensure correct CLK_DFLL48M frequency.

A zero-to-one transition of STATUS.DFLLOOB will generate an interrupt, if the DFLL Out Of Bounds bit in the Interrupt Enable Set register (INTENSET.DFLLOOB) is '1'. This interrupt will also be set if the tuner is not able to lock on the correct Coarse value.

To avoid this out-of-bounds error, the reference clock must be stable; an external oscillator XOSC32K is recommended.

20.6.5.1.7 Reference Clock Stop Detection

If GCLK_DFLL48M stops or is running at a very low frequency (slower than CLK_DFLL48M/(2 * MUL_{MAX})), the DFLL Reference Clock Stopped bit in the Status register (STATUS.DFLLRCS) will be set.

Detecting a stopped reference clock can take a long time, in the order of $2^{(17)}$ CLK_DFLL48M cycles.

When the reference clock is stopped, the DFLL48M will operate as if in open-loop mode. Closed-loop mode operation will automatically resume when the GCLK_DFLL48M is restarted.

A zero-to-one transition of the DFLL Reference Clock Stopped bit in the Status register (STATUS.DFLLRCS) will generate an interrupt, if the DFLL Reference Clock Stopped bit in the Interrupt Enable Set register (INTENSET.DFLLRCS) is '1'.

20.6.5.2 Additional Features

20.6.5.2.1 Dealing with Settling Time in Closed-Loop Mode

The time from selecting a new CLK_DFLL48M output frequency until this frequency is output by the DFLL48M can be up to several microseconds. A small value in DFLLMUL.MUL can lead to instability in the DFLL48M locking mechanism, which can prevent the DFLL48M from achieving locks.

To avoid this, a chill cycle can be enabled, during which the CLK_DFLL48M frequency is not measured. The chill cycle is enabled by default, but can be disabled by writing '1' to the DFLL Chill Cycle Disable bit in the DFLL Control register (DFLLCTRL.CCDIS). Enabling chill cycles might double the lock time.

Another solution to this problem is using less strict lock requirements. This is called Quick Lock (QL). QL is enabled by default as well, but it can be disabled by writing '1' to the Quick Lock Disable bit in the DFLL Control register (DFLLCTRL.QLDIS). The Quick Lock might lead to a larger spread in the output frequency than chill cycles, but the average output frequency is the same.

20.6.5.2.2 USB Clock Recovery Mode

USB Clock Recovery mode can be used to create the 48MHz USB clock from the USB Start Of Frame (SOF). This mode is enabled by writing a '1' to both the USB Clock Recovery Mode bit and the Mode bit in DFLL Control register (DFLLCTRL.USBCRM and DFLLCTRL.MODE).

The SOF signal from USB device will be used as reference clock (GCLK_DFLL48M), ignoring the selected generic clock reference. When the USB device is connected, a SOF will be sent every 1ms, thus DFLLVAL.MUX bits should be written to 0xBB80 to obtain a 48MHz clock.

In USB clock recovery mode, the DFLLCTRL.BPLCKC bit state is ignored, and the value stored in the DFLLVAL.COARSE will be used as final Coarse Value. The COARSE calibration value can be loaded from NVM OTP row by software. The locking procedure will also go instantaneously to the fine lock search.

The DFLLCTRL.QLDIS bit must be cleared and DFLLCTRL.CCDIS should be set to speed up the lock phase. The DFLLCTRL.STABLE bit state is ignored, an auto jitter reduction mechanism is used instead.

Important: When the DFLL is in USB Clock Recovery mode and the USB is in Suspend mode, the STATUS.DFLLRCS (DFLL Reference Clock Stopped) and STATUS.DFLLLCKF (DFLL Lock Fine) status bits may be set due to the absence of SOF (Start Of Frame).
After the USB leaves the Suspend mode (Resume), these status bits are still set.

As a consequence, disregard STATUS.DFLLRCS and STATUS.DFLLLCKF status bits after a USB Suspend until the corresponding interrupt flags are cleared.

20.6.5.2.3 Wake from Sleep Modes

DFLL48M can optionally reset its lock bits when it is disabled. This is configured by the Lose Lock After Wake bit in the DFLL Control register (DFLLCTRL.LLAW).

If DFLLCTRL.LLAW is zero, the DFLL48M will be re-enabled and start running with the same configuration as before being disabled, even if the reference clock is not available. The locks will not be lost. After the reference clock has restarted, the fine lock tracking will quickly compensate for any frequency drift during sleep if DFLLCTRL.STABLE is zero.

If DFLLCTRL.LLAW is '1' when disabling the DFLL48M, the DFLL48M will lose all its locks, and needs to regain these through the full lock sequence.

20.6.5.2.4 Wait for lock

DFLL48M can optionally control the issued clock. This is configured by the Wait For Lock bit (DFLLCTRL.WAITLOCK) in the DFLL Control register. If DFLLCTRLB.WAITLOCK is zero, the DFLL48M will issue a clock immediately after the ready bit (STATUS.DFLLRDY) has risen. If DFLLCTRLB.WAITLOCK is one, the DFLL48M will issue a clock immediately after the fine lock bit (STATUS.DFLLLCKF) has risen. Using the wait for lock feature allows a better accuracy of the issued DFLL48M clock, conversely it increases the startup time of the DFLL48M clock.

20.6.5.2.5 Accuracy

There are three main factors that determine the accuracy of $F_{clkdfll48m}$. These can be tuned to obtain maximum accuracy when fine lock is achieved.

- Fine resolution. The frequency step between two Fine values. This is relatively smaller for higher output frequencies.
- Resolution of the measurement: If the resolution of the measured $F_{clkdfll48m}$ is low, i.e., the ratio between the CLK_DFLL48M frequency and the GCLK_DFLL48M frequency is small, the DFLL48M might lock at a frequency that is lower than the targeted frequency. It is recommended to use a reference clock frequency of 32.768 kHz or lower to avoid this issue for low target frequencies.
- The accuracy of the reference clock.

20.6.6 32 MHz - 96 MHz Fractional Digital Phase-Locked Loop (FDPLL96M) Operation

The task of the DPLL is to maintain coherence between the input (reference) signal and the respective output frequency, CLK_DPLL, via phase comparison. The DPLL controller supports three independent sources of reference clocks:

- CLK_XOSC32K: this clock is provided by the 32.768 kHz crystal oscillator (XOSC32K).
- CLK_XOSC: this clock is provided by the 0.4 to 32 MHz crystal oscillator (XOSC)
- GCLK_FDPLL96M: this clock is provided by the Generic Clock Controller (GCLK).

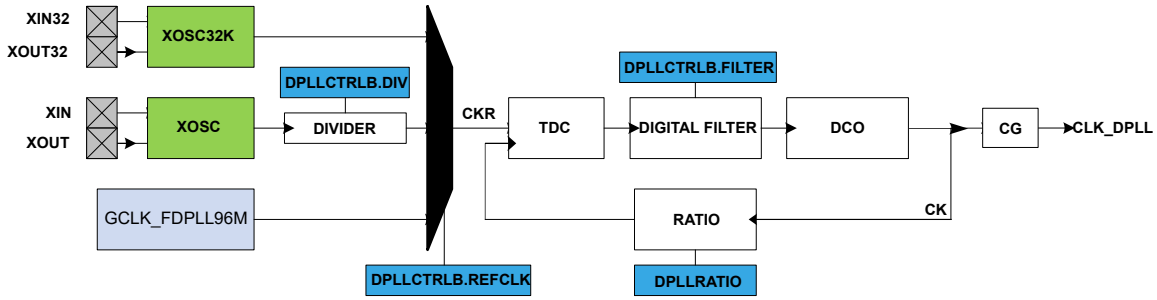
Note: Another clock (GCLK_FDPLL96M_32K) is used by the FDPLL96M, but as the FDPLL96M internal clock timer for counting the user defined lock time (refer to Initialization, Enabling, Disabling, and Resetting for more details).

When the controller is enabled, the relationship between the reference clock frequency and the output clock frequency is:

$$f_{CK} = f_{CKR} \times \left(\mathrm{LDR} + 1 + \frac{\mathrm{LDRFRAC}}{16}\right) \times \frac{1}{2^{\mathrm{PRESC}}}$$

Where f_{CK} is the frequency of the DPLL output clock, LDR is the loop divider ratio integer part, LDRFRAC is the loop divider ratio fractional part, f_{CKR} is the frequency of the selected reference clock, and PRESC is the output prescaler value.

Figure 20-3. DPLL Block Diagram



When the controller is disabled, the output clock is low. If the Loop Divider Ratio Fractional part bit field in the DPLL Ratio register (DPLLRATIO.LDRFRAC) is zero, the DPLL works in integer mode. Otherwise, the fractional mode is activated. The fractional part has a negative impact on the jitter of the DPLL.

For example (integer mode only): Assuming F_{CKR} = 32 kHz and F_{CK} = 48 MHz, the multiplication ratio is 1500. It means that LDR will be set to 1499.

For example (fractional mode): Assuming F_{CKR} = 32 kHz and F_{CK} = 48.006 MHz, the multiplication ratio is 1500.1875 (1500 + 3/16). Thus LDR is set to 1499 and LDRFRAC to 3.

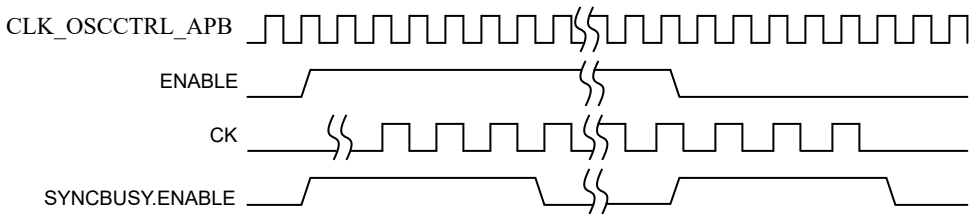
20.6.6.1 Basic Operation

20.6.6.1.1 Initialization, Enabling, Disabling, and Resetting

The DPLL is enabled by writing a '1' to the Enable bit in the DPLL Control A register (DPLLCTRLA.ENABLE). The DPLL is disabled by writing a zero to this bit.

The DPLLSYNCBUSY.ENABLE is set when the DPLLCTRLA.ENABLE bit is modified. It is cleared when the DPLL output clock CK has sampled the bit at the high level after enabling the DPLL. When disabling the DPLL, DPLLSYNCBUSY.ENABLE is cleared when the output clock is no longer running.

Figure 20-4. Enable Synchronization Busy Operation



The frequency of the DPLL output clock CK is stable when the module is enabled and when the Lock bit in the DPLL Status register is set (DPLLSTATUS.LOCK).

When the Lock Time bit field in the DPLL Control B register (DPLLCTRLB.LTIME) is non-zero, a user defined lock time is used to validate the lock operation. In this case the lock time is constant. If DPLLCTRLB.LTIME=0, the lock signal is linked with the status bit of the DPLL, and the lock time varies depending on the filter selection and the final target frequency.
Note: GCLK_FDPLL96M_32K is responsible for counting the user defined lock time (LTIME different from 0x0), hence must be enabled.

When the Wake Up Fast bit (DPLLCTRLB.WUF) is set, the wake up fast mode is activated. In this mode the clock gating cell is enabled at the end of the start-up time. At this time the final frequency is not stable, as it is still during the acquisition period, but it allows to save several milliseconds. After first acquisition, the clock gater (CG) generating the output clock CLK_DPLL is gated by the LOCK signal when the Lock Bypass bit (DPLLCTRLB.LBYPASS) is cleared or is not gated and delivers the output clock CLK_DPLL immediately when DPLLCTRLB.LBYPASS is set.

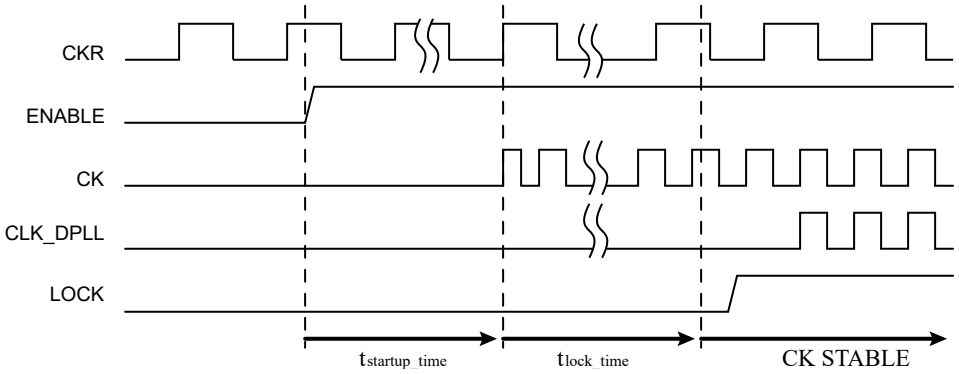
Table 20-4. CLK_DPLL Behavior from Startup to First Edge Detection

WUF	LTIME	CLK_DPLL Behavior
0	0	Normal Mode: First Edge when lock is asserted
0	Not Equal To Zero	Lock Timer Timeout mode: First Edge when the timer down-counts to 0.
1	X	Wake Up Fast Mode: First Edge when CK is active (startup time)

Table 20-5. CLK_DPLL Behavior after First Edge Detection

LBYPASS	CLK_DPLL Behavior
0	Normal Mode: the CLK_DPLL is turned off when lock signal is low.
1	Lock Bypass Mode: the CLK_DPLL is always running, lock is irrelevant.

Figure 20-5. CK and CLK_DPLL Output from DPLL Off Mode to Running Mode



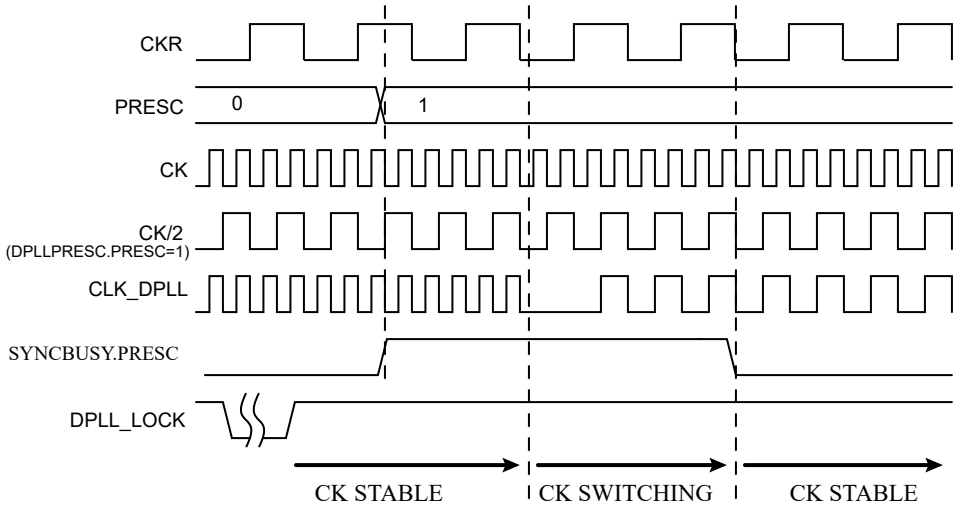
20.6.6.1.2 Reference Clock Switching

When a software operation requires reference clock switching, the recommended procedure is to turn the DPLL into the standby mode, modify the DPLLCTRLB.REFCLK to select the desired reference source, and activate the DPLL again.

20.6.6.1.3 Output Clock Prescaler

The DPLL controller includes an output prescaler. This prescaler provides three selectable output clocks CK, CK/2 and CK/4. The Prescaler bit field in the DPLL Prescaler register (DPLLPRESC.PRESC) is used to select a new output clock prescaler. When the prescaler field is modified, the DPLLSYNCBUSY.DPLLPRESC bit is set. It will be cleared by hardware when the synchronization is over.

Figure 20-6. Output Clock Switching Operation

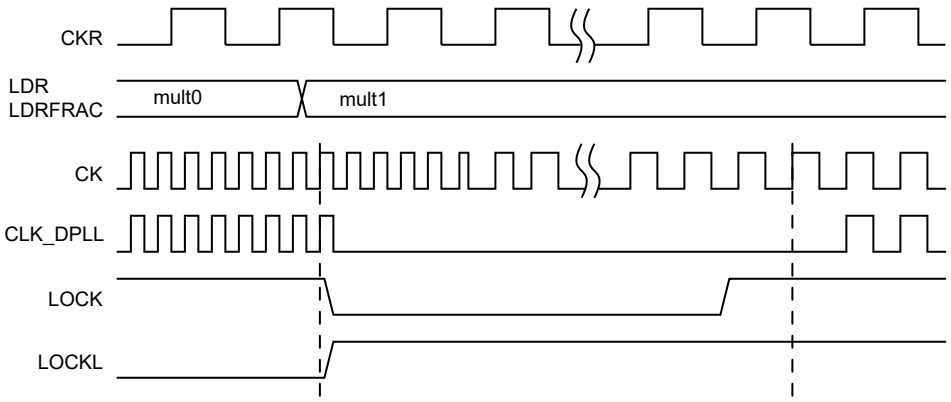


20.6.6.1.4 Loop Divider Ratio Updates

The DPLL Controller supports on-the-fly update of the DPLL Ratio Control (DPLLRATIO) register, allowing to modify the loop divider ratio and the loop divider ratio fractional part when the DPLL is enabled.

STATUS.DPLLLDRTO is set when the DPLLRATIO register has been modified and the DPLL analog cell has successfully sampled the updated value. At that time the DPLLSTATUS.LOCK bit is cleared and set again by hardware when the output frequency reached a stable state.

Figure 20-7. RATIOCTRL register update operation



20.6.6.1.5 Digital Filter Selection

The PLL digital filter (PI controller) is automatically adjusted in order to provide a good compromise between stability and jitter. Nevertheless a software operation can override the filter setting using the Filter bit field in the DPLL Control B register (DPLLCTRLB.FILTER). The Low Power Enable bit (DPLLCTRLB.LPEN) can be use to bypass the Time to Digital Converter (TDC) module.

20.6.7 Interrupts

The OSCCTRL has the following interrupt sources:

- XOSCRDY - Multipurpose Crystal Oscillator Ready: A 0-to-1 transition on the STATUS.XOSCRDY bit is detected
- XOSCFAIL - XOSC Clock Failure. A 0-to-1 transition on the STATUS.XOSCFAIL bit is detected
- OSC16MRDY - 16 MHz Internal Oscillator Ready: A 0-to-1 transition on the STATUS.OSC16MRDY bit is detected
- DFLL-related:
  - DFLLRDY - DFLL48M Ready: A 0-to-1 transition of the STATUS.DFLLRDY bit is detected
  - DFLLOOB - DFLL48M Out Of Boundaries: A 0-to-1 transition of the STATUS.DFLLOOB bit is detected
  - DFLLLOCKF - DFLL48M Fine Lock: A 0-to-1 transition of the STATUS.DFLLLCKF bit is detected
  - DFLLLOCKC - DFLL48M Coarse Lock: A 0-to-1 transition of the STATUS.DFLLLCKC bit is detected
  - DFLLRCS - DFLL48M Reference Clock has Stopped: A 0-to-1 transition of the STATUS.DFLLRCS bit is detected
- DFLLULP-related:
  - DFLLULPRDY - DFLLULP Ready: A 0-to1 transition of the STATUS.DFLLULPRDY bit is detected.
  - DFLLULPLOCK - DFLLULP Lock: A 0-to-1 transition of the STATUS.DFLLULPLOCK bit is detected.
  - DFLLULPNOLOCK - DFLLULP No Lock: A 0-to-1 transition of the STATUS.DFLLULPNOLOCK bit is detected.
- DPLL-related:
  - DPLLLCKR - DPLL Lock Rise: A 0-to-1 transition of the STATUS.DPLLLCKR bit is detected
  - DPLLLCKF - DPLL Lock Fall: A 0-to-1 transition of the STATUS.DPLLLCKF bit is detected
  - DPLLLTTO - DPLL Lock Timer Time-out: A 0-to-1 transition of the STATUS.DPLLLTTO bit is detected
  - DPLLLTO - DPLL Loop Divider Ratio Update Complete. A 0-to-1 transition of the STATUS.DPLLTO bit is detected

All these interrupts are synchronous wake-up source.

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).

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 OSCCTRL is reset. See the INTFLAG register for details on how to clear interrupt flags.

The OSCCTRL 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. Refer to the INTFLAG register for details.

Note: The interrupts must be globally enabled for interrupt requests to be generated.

20.6.8 Events

The CFD can generate the following output event:

- Clock Failure Detection (CFD): Generated when the Clock Failure status bit is set in the Status register (STATUS.XOSCFAIL). The CFD event is not generated when the Clock Switch Back Enable bit (STATUS.XOSCCKSW) in the Status register is set.

Writing a '1' to an Event Output bit in the Event Control register (EVCTRL.CFDEO) enables the CFD output event. Writing a '0' to this bit disables the CFD output event. Refer to the Event System chapter for details on configuring the event system.

The DFLLULP can take the following actions on an input event (TUNE):

- Unlock the DFLLULP close loop tuner and start over a frequency tuning, depending on the settings of the DFLLULP registers, until the tuner achieves a new lock.

Writing a '1' to the Event Input Enable bit in the Event Control register (EVCTRL.TUNEEI) enables the corresponding action on input event. Writing a '0' to this bit disables the corresponding action on input event. Refer to the Event System chapter for details on configuring the event system.

20.6.9 Synchronization

DFLL48M

Due to the multiple clock domains, values in the DFLL48M control registers need to be synchronized to other clock domains.

Once the DFLL is enabled, any read and write operation requires the DFLL Ready bit in the Status register (STATUS.DFLLRDY) to read '1'.

Note: Once the DFLL48M is enabled in on-demand mode (DFLLCTRL.ONDEMAND=1), the STATUS.DFLLRDY bit will keep to '0' until the DFLL48M is requested by a peripheral.

Before writing to any of the DFLL48M control registers, the user must check that the DFLL Ready bit (STATUS.DFLLRDY) is set to '1'. When this bit is set, the DFLL48M can be configured and CLK_DFLL48M is ready to be used. Any write to any of the DFLL48M control registers while DFLLRDY is '0' will be ignored.

In order to read from the DFLLVAL register in closed loop mode, the user must request a read synchronization by writing a '1' to the Read Request bit in the DFLL Synchronization register (DFLLSYNC.READREQ). This is required because the DFLL controller may change the content of the DFLLVAL register any time. If a read operation is issued while the DFLL controller is updating the DFLLVAL content, a zero will be returned.

Note: Issuing a read on any register while a write-synchronization is still on-going will return a zero.

Read-Synchronized registers using DFLLSYNC.READREQ:

- DFLL48M Value register (DFLLVAL)

Write-Synchronized registers:

- DFLL48M Control register (DFLLCTRL)
- DFLL48M Value register (DFLLVAL)
- DFLL48M Multiplier register (DFLLMUL)

DFLLULP

Due to the asynchronicity between the main clock domain (CLK_OSCCTRL_APB) and the internal clock domain, some registers are synchronized when written. When a write-synchronized register is written, the corresponding bit in the Synchronization Busy register (DFLLULPSYNCBUSY) is set immediately. When the write-synchronization is complete, this bit is cleared. Reading a write-synchronized register while the synchronization is ongoing will return the

value written, and not the current value in the peripheral clock domain. To read the current value in the peripheral clock domain after writing a register, the user must wait for the corresponding DFLLULPSYNCBUSY bit to be cleared before reading the value.

If a register is written while the corresponding bit in DFLLULPSYNCBUSY is one, the write is discarded and an error is generated.

The following bits and registers are write-synchronized:

- Delay Value register (DFLLULPDLY)

Write-synchronization is denoted by the Write-Synchronized property in the register description.

FDPLL96M
Due to the multiple clock domains, some registers in the FDPLL96M must be synchronized when accessed.

When executing an operation that requires synchronization, the relevant synchronization bit in the Synchronization Busy register (DPLLSYNCBUSY) will be set immediately, and cleared when synchronization is complete.

The following bits need synchronization when written:

- Enable bit in control register A (DPLLCTRL.ENABLE)
- DPLL Ratio register (DPLLRATIO)
- DPLL Prescaler register (DPLLPRESC)

20.6.10 Debug Operation

When the CPU is halted in debug mode the OSCCTRL continues normal operation. If the OSCCTRL 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.

20.7 Register Summary

Refer to the Registers Description section for more details on register properties and access permissions.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	EVCTRL	7:0						TUNEINV	TUNEEI	CFDEO
0x01 ... 0x03	Reserved									
0x04	INTENCLR	31:24				DFLLRCS	DFLLLCKC	DFLLLCKF	DFLLOOB	DFLLRDY
		23:16					DPLLLDRTO	DPLLLTO	DPLLLCKF	DPLLLCKR
		15:8						DFLLULPNOLOCK	DFLLULPLOCK	DFLLULPRDY
		7:0				OSC16MRDY			XOSCFAIL	XOSCRDY
0x08	INTENSET	31:24				DFLLRCS	DFLLLCKC	DFLLLCKF	DFLLOOB	DFLLRDY
		23:16					DPLLLDRTO	DPLLLTO	DPLLLCKF	DPLLLCKR
		15:8						DFLLULPNOLOCK	DFLLULPLOCK	DFLLULPRDY
		7:0				OSC16MRDY			XOSCFAIL	XOSCRDY
0x0C	INTFLAG	31:24				DFLLRCS	DFLLLCKC	DFLLLCKF	DFLLOOB	DFLLRDY
		23:16					DPLLLDRTO	DPLLLTO	DPLLLCKF	DPLLLCKR
		15:8						DFLLULPNOLOCK	DFLLULPLOCK	DFLLULPRDY
		7:0				OSC16MRDY			XOSCFAIL	XOSCRDY
0x10	STATUS	31:24				DFLLRCS	DFLLLCKC	DFLLLCKF	DFLLOOB	DFLLRDY
		23:16					DPLLLDRTO	DPLLLTO	DPLLLCKF	DPLLLCKR
		15:8						DFLLULPNOLOCK	DFLLULPLOCK	DFLLULPRDY
		7:0				OSC16MRDY		XOSCCKSW	XOSCFAIL	XOSCRDY
0x14	XOSCCTRL	15:8	STARTUP[3:0]				AMPGC	GAIN[2:0]		
		7:0	ONDEMAND	RUNSTDBY		SWBEN	CFDEN	XTALEN	ENABLE	
0x16	CFDPRESC	7:0						CFDPRESC[2:0]		
0x17	Reserved									
0x18	OSC16MCTRL	7:0	ONDEMAND	RUNSTDBY			FSEL[1:0]		ENABLE	
0x19 ... 0x1B	Reserved									
0x1C	DFLLULPCTRL	15:8						DIV[2:0]		
		7:0	ONDEMAND	RUNSTDBY	DITHER	SAFE	BINSE	Reserved	ENABLE	
0x1E	DFLLULPDITHER	7:0		PER[2:0]				STEP[2:0]		
0x1F	DFLLULPRREQ	7:0	RREQ							
0x20	DFLLULPDLY	31:24								
		23:16								
		15:8								
		7:0	DELAY[7:0]							
0x24	DFLLULPRATIO	31:24								
		23:16								
		15:8						RATIO[10:8]		
		7:0	RATIO[7:0]							
0x28	DFLLULPSYNCBUSY	31:24								
		23:16								
		15:8								
		7:0					DELAY	Reserved	ENABLE	
0x2C ... 0x2F	Reserved									
0x30	DFLLCTRL	15:8					WAITLOCK	BPLCKC	QLDIS	CCDIS
		7:0	ONDEMAND	RUNSTDBY	USBCRM	LLAW	STABLE	MODE	ENABLE	

.....continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x32 ... 0x33	Reserved									
0x34	DFLLVAL	31:24	DIFF[15:8]							
		23:16	DIFF[7:0]							
		15:8	COARSE[5:0]						FINE[9:8]	
		7:0	FINE[7:0]							
0x38	DFLLMUL	31:24	CSTEP[5:0]						FSTEP[9:8]	
		23:16	FSTEP[7:0]							
		15:8	MUL[15:8]							
		7:0	MUL[7:0]							
0x3C	DFLLSYNC	7:0	READREQ							
0x3D ... 0x3F	Reserved									
0x40	DPLLCTRLA	7:0	ONDEMAND	RUNSTDBY					ENABLE	
0x41 ... 0x43	Reserved									
0x44	DPLLRATIO	31:24								
		23:16					LDRFRAC[3:0]			
		15:8					LDR[11:8]			
		7:0	LDR[7:0]							
0x48	DPLLCTRLB	31:24						DIV[10:8]		
		23:16	DIV[7:0]							
		15:8				LBYPASS		LTIME[2:0]		
		7:0			REFCLK[1:0]		WUF	LPEN	FILTER[1:0]	
0x4C	DPLLPRESC	7:0							PRESC[1:0]	
0x4D ... 0x4F	Reserved									
0x50	DPLLSYNCBUSY	7:0					DPLLPRESC	DPLLRATIO	ENABLE	
0x51 ... 0x53	Reserved									
0x54	DPLLSTATUS	7:0							CLKRDY	LOCK

20.7.1 Event Control

Name: EVCTRL
Offset: 0x00
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
						TUNEINV	TUNEEI	CFDEO
Access						R/W	R/W	R/W
Reset						0	0	0

Bit 2 – TUNEINV Tune Event Input Invert
This bit is used to invert the input event of the DFLLULP tuner.

Value	Description
0	Tune event input source is not inverted.
1	Tune event input source is inverted.

Bit 1 – TUNEEI Tune Event Input Enable
This bit is used to enable the input event of the DFLLULP tuner.

Value	Description
0	A new closed loop tuning will not be triggered on any incoming event.
1	A new closed loop tuning will be triggered on any incoming event.

Bit 0 – CFDEO Clock Failure Detector Event Output Enable
This bit indicates whether the Clock Failure detector event output is enabled or not and an output event will be generated when the Clock Failure detector detects a clock failure

Value	Description
0	Clock Failure detector event output is disabled and no event will be generated.
1	Clock Failure detector event output is enabled and an event will be generated.

20.7.2 Interrupt Enable Clear

Name: INTENCLR
Offset: 0x04
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
				DFLLRCS	DFLLLCKC	DFLLLCKF	DFLLOOB	DFLLRDY
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
					DPLLLDRTO	DPLLLTO	DPLLLCKF	DPLLLCKR
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
						DFLLULPNOLOCK	DFLLULPLOCK	DFLLULPRDY
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	7	6	5	4	3	2	1	0
				OSC16MRDY			XOSCFAIL	XOSCRDY
Access				R/W			R/W	R/W
Reset				0			0	0

Bit 28 – DFLLRCS DFLL Reference Clock Stopped Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the DFLL Reference Clock Stopped Interrupt Enable bit, which disables the DFLL Reference Clock Stopped interrupt.

Value	Description
0	The DFLL Reference Clock Stopped interrupt is disabled.
1	The DFLL Reference Clock Stopped interrupt is enabled, and an interrupt request will be generated when the DFLL Reference Clock Stopped Interrupt flag is set.

Bit 27 – DFLLLCKC DFLL Lock Coarse Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the DFLL Lock Coarse Interrupt Enable bit, which disables the DFLL Lock Coarse interrupt.

Value	Description
0	The DFLL Lock Coarse interrupt is disabled.
1	The DFLL Lock Coarse interrupt is enabled, and an interrupt request will be generated when the DFLL Lock Coarse Interrupt flag is set.

Bit 26 – DFLLLCKF DFLL Lock Fine Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the DFLL Lock Fine Interrupt Enable bit, which disables the DFLL Lock Fine interrupt.

Value	Description
0	The DFLL Lock Fine interrupt is disabled.
1	The DFLL Lock Fine interrupt is enabled, and an interrupt request will be generated when the DFLL Lock Fine Interrupt flag is set.

Bit 25 – DFLLOOB DFLL Out Of Bounds Interrupt Enable

Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the DFLL Out Of Bounds Interrupt Enable bit, which disables the DFLL Out Of Bounds interrupt.

Value	Description
0	The DFLL Out Of Bounds interrupt is disabled.
1	The DFLL Out Of Bounds interrupt is enabled, and an interrupt request will be generated when the DFLL Out Of Bounds Interrupt flag is set.

Bit 24 – DFLLRDY DFLL Ready Interrupt Enable

Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the DFLL Ready Interrupt Enable bit, which disables the DFLL Ready interrupt.

Value	Description
0	The DFLL Ready interrupt is disabled.
1	The DFLL Ready interrupt is enabled, and an interrupt request will be generated when the DFLL Ready Interrupt flag is set.

Bit 19 – DPLLLDRTO DPLL Loop Divider Ratio Update Complete Interrupt Enable

Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the DPLL Loop Divider Ratio Update Complete Interrupt Enable bit, which disables the DPLL Loop Divider Ratio Update Complete interrupt.

Value	Description
0	The DPLL Loop Divider Ratio Update Complete interrupt is disabled.
1	The DPLL Loop Divider Ratio Update Complete interrupt is enabled, and an interrupt request will be generated when the DPLL Loop Divider Ratio Update Complete Interrupt flag is set.

Bit 18 – DPLLLTO DPLL Lock Timeout Interrupt Enable

Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the DPLL Lock Timeout Interrupt Enable bit, which disables the DPLL Lock Timeout interrupt.

Value	Description
0	The DPLL Lock Timeout interrupt is disabled.
1	The DPLL Lock Timeout interrupt is enabled, and an interrupt request will be generated when the DPLL Lock Timeout Interrupt flag is set.

Bit 17 – DPLLLCKF DPLL Lock Fall Interrupt Enable

Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the DPLL Lock Fall Interrupt Enable bit, which disables the DPLL Lock Fall interrupt.

Value	Description
0	The DPLL Lock Fall interrupt is disabled.
1	The DPLL Lock Fall interrupt is enabled, and an interrupt request will be generated when the DPLL Lock Fall Interrupt flag is set.

Bit 16 – DPLLLCKR DPLL Lock Rise Interrupt Enable

Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the DPLL Lock Rise Interrupt Enable bit, which disables the DPLL Lock Rise interrupt.

Value	Description
0	The DPLL Lock Rise interrupt is disabled.
1	The DPLL Lock Rise interrupt is enabled, and an interrupt request will be generated when the DPLL Lock Rise Interrupt flag is set.

Bit 10 – DFLLULPNOLOCK DFLLULP No Lock Interrupt Enable

Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the DFLLULP No Lock interrupt Enable bit, which disables the DFLLULP No Lock interrupt.

Value	Description
0	The DFLLULP No Lock is disabled.

Value	Description
1	The DFLLULP No Lock interrupt is enabled, and an interrupt request will be generated when the DFLLULP No Lock Interrupt flag is set.

Bit 9 – DFLLULPLOCK DFLLULP Lock Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the DFLLULP Lock Interrupt Enable bit, which disables the DFLLULP Lock interrupt.

Value	Description
0	The DFLLULP Lock interrupt is disabled.
1	The DFLLULP Lock interrupt is enabled, and an interrupt request will be generated when the DFLLULP Lock Interrupt flag is set.

Bit 8 – DFLLULPRDY DFLLULP Ready interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the DFLLULP Ready Interrupt Enable bit, which disables the DFLLULP Ready interrupt.

Value	Description
0	The DFLLULP Ready interrupt is disabled.
1	The DFLLULP Ready interrupt is enabled, and an interrupt request will be generated when the DFLLULP Ready Interrupt flag is set.

Bit 4 – OSC16MRDY OSC16M Ready Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the OSC16M Ready Interrupt Enable bit, which disables the OSC16M Ready interrupt.

Value	Description
0	The OSC16M Ready interrupt is disabled.
1	The OSC16M Ready interrupt is enabled, and an interrupt request will be generated when the OSC16M Ready Interrupt flag is set.

Bit 1 – XOSCFAIL XOSC Clock Failure Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the XOSC Clock Failure Interrupt Enable bit, which disables the XOSC Clock Failure interrupt.

Value	Description
0	The XOSC Clock Failure interrupt is disabled.
1	The XOSC Clock Failure interrupt is enabled, and an interrupt request will be generated when the XOSC Clock Failure Interrupt flag is set.

Bit 0 – XOSCRDY XOSC Ready Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the XOSC Ready Interrupt Enable bit, which disables the XOSC Ready interrupt.

Value	Description
0	The XOSC Ready interrupt is disabled.
1	The XOSC Ready interrupt is enabled, and an interrupt request will be generated when the XOSC Ready Interrupt flag is set.

20.7.3 Interrupt Enable Set

Name: INTENSET
Offset: 0x08
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 Clear register (INTENCLR).

Bit	31	30	29	28	27	26	25	24
				DFLLRCS	DFLLLCKC	DFLLLCKF	DFLLOOB	DFLLRDY
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
					DPLLLDRTO	DPLLLTO	DPLLLCKF	DPLLLCKR
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
						DFLLULPNOLOCK	DFLLULPLOCK	DFLLULPRDY
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	7	6	5	4	3	2	1	0
				OSC16MRDY			XOSCFAIL	XOSCRDY
Access				R/W			R/W	R/W
Reset				0			0	0

Bit 28 – DFLLRCS DFLL Reference Clock Stopped Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will set the DFLL Reference Clock Stopped Interrupt Enable bit, which enables the DFLL Reference Clock Stopped interrupt.

Value	Description
0	The DFLL Reference Clock Stopped interrupt is disabled.
1	The DFLL Reference Clock Stopped interrupt is enabled, and an interrupt request will be generated when the DFLL Reference Clock Stopped Interrupt flag is set.

Bit 27 – DFLLLCKC DFLL Lock Coarse Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will set the DFLL Lock Coarse Interrupt Enable bit, which enables the DFLL Lock Coarse interrupt.

Value	Description
0	The DFLL Lock Coarse interrupt is disabled.
1	The DFLL Lock Coarse interrupt is enabled, and an interrupt request will be generated when the DFLL Lock Coarse Interrupt flag is set.

Bit 26 – DFLLLCKF DFLL Lock Fine Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will set the DFLL Lock Fine Interrupt Disable/Enable bit, disable the DFLL Lock Fine interrupt and set the corresponding interrupt request.

Value	Description
0	The DFLL Lock Fine interrupt is disabled.
1	The DFLL Lock Fine interrupt is enabled, and an interrupt request will be generated when the DFLL Lock Fine Interrupt flag is set.

Bit 25 – DFLLOOB DFLL Out Of Bounds Interrupt Enable

Writing '0' to this bit has no effect.
Writing '1' to this bit will set the DFLL Out Of Bounds Interrupt Enable bit, which enables the DFLL Out Of Bounds interrupt.

Value	Description
0	The DFLL Out Of Bounds interrupt is disabled.
1	The DFLL Out Of Bounds interrupt is enabled, and an interrupt request will be generated when the DFLL Out Of Bounds Interrupt flag is set.

Bit 24 – DFLLRDY DFLL Ready Interrupt Enable

Writing '0' to this bit has no effect.
Writing '1' to this bit will set the DFLL Ready Interrupt Enable bit, which enables the DFLL Ready interrupt and set the corresponding interrupt request.

Value	Description
0	The DFLL Ready interrupt is disabled.
1	The DFLL Ready interrupt is enabled, and an interrupt request will be generated when the DFLL Ready Interrupt flag is set.

Bit 19 – DPLLLDRTO DPLL Loop Divider Ratio Update Complete Interrupt Enable

Writing '0' to this bit has no effect.
Writing '1' to this bit will set the DPLL Loop Divider Ratio Update Complete Interrupt Enable bit, which enables the DPLL Loop Divider Ratio Update Complete interrupt.

Value	Description
0	The DPLL Loop Divider Ratio Update Complete interrupt is disabled.
1	The DPLL Loop Divider Ratio Update Complete interrupt is enabled, and an interrupt request will be generated when the DPLL Loop Divider Ratio Update Complete Interrupt flag is set.

Bit 18 – DPLLLTO DPLL Lock Timeout Interrupt Enable

Writing '0' to this bit has no effect.
Writing '1' to this bit will set the DPLL Lock Timeout Interrupt Enable bit, which enables the DPLL Lock Timeout interrupt.

Value	Description
0	The DPLL Lock Timeout interrupt is disabled.
1	The DPLL Lock Timeout interrupt is enabled, and an interrupt request will be generated when the DPLL Lock Timeout Interrupt flag is set.

Bit 17 – DPLLLCKF DPLL Lock Fall Interrupt Enable

Writing '0' to this bit has no effect.
Writing '1' to this bit will set the DPLL Lock Fall Interrupt Enable bit, which enables the DPLL Lock Fall interrupt.

Value	Description
0	The DPLL Lock Fall interrupt is disabled.
1	The DPLL Lock Fall interrupt is enabled, and an interrupt request will be generated when the DPLL Lock Fall Interrupt flag is set.

Bit 16 – DPLLLCKR DPLL Lock Rise Interrupt Enable

Writing '0' to this bit has no effect.
Writing '1' to this bit will set the DPLL Lock Rise Interrupt Enable bit, which enables the DPLL Lock Rise interrupt.

Value	Description
0	The DPLL Lock Rise interrupt is disabled.
1	The DPLL Lock Rise interrupt is enabled, and an interrupt request will be generated when the DPLL Lock Rise Interrupt flag is set.

Bit 10 – DFLLULPNOLOCK DFLLULP No Lock Interrupt Enable

Writing '0' to this bit has no effect.
Writing '1' to this bit will set the DFLLULP No Lock interrupt Enable bit, which enables the DFLLULP No Lock interrupt.

Value	Description
0	The DFLLULP No Lock is disabled.
1	The DFLL No Lock interrupt is enabled, and an interrupt request will be generated when the DFLL No Lock Interrupt flag is set.

Bit 9 – DFLLULPLOCK DFLLULP Lock Interrupt Enable

Writing '0' to this bit has no effect.
Writing '1' to this bit will set the DFLLULP Lock Interrupt Enable bit, which enables the DFLLULP Lock interrupt.

Value	Description
0	The DFLLULP Lock interrupt is disabled.
1	The DFLLULP Lock interrupt is enabled, and an interrupt request will be generated when the DFLL Lock Interrupt flag is set.

Bit 8 – DFLLULPRDY DFLLULP Ready interrupt Enable

Writing '0' to this bit has no effect.
Writing '1' to this bit will set the DFLLULP Ready Interrupt Enable bit, which enables the DFLLULP Ready interrupt.

Value	Description
0	The DFLLULP Ready interrupt is disabled.
1	The DFLLULP Ready interrupt is enabled, and an interrupt request will be generated when the DFLLULP Ready Interrupt flag is set.

Bit 4 – OSC16MRDY OSC16M Ready Interrupt Enable

Writing '0' to this bit has no effect.
Writing '1' to this bit will set the OSC16M Ready Interrupt Enable bit, which enables the OSC16M Ready interrupt.

Value	Description
0	The OSC16M Ready interrupt is disabled.
1	The OSC16M Ready interrupt is enabled, and an interrupt request will be generated when the OSC16M Ready Interrupt flag is set.

Bit 1 – XOSCFAIL XOSC Clock Failure Interrupt Enable

Writing '0' to this bit has no effect.
Writing '1' to this bit will set the XOSC Clock Failure Interrupt Enable bit, which enables the XOSC Clock Failure Interrupt.

Value	Description
0	The XOSC Clock Failure Interrupt is disabled.
1	The XOSC Clock Failure Interrupt is enabled, and an interrupt request will be generated when the XOSC Clock Failure Interrupt flag is set.

Bit 0 – XOSCRDY XOSC Ready Interrupt Enable

Writing '0' to this bit has no effect.
Writing '1' to this bit will set the XOSC Ready Interrupt Enable bit, which enables the XOSC Ready interrupt.

Value	Description
0	The XOSC Ready interrupt is disabled.
1	The XOSC Ready interrupt is enabled, and an interrupt request will be generated when the XOSC Ready Interrupt flag is set.

20.7.4 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x0C
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
				DFLLRCS	DFLLLCKC	DFLLLCKF	DFLLOOB	DFLLRDY
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
					DPLLLDRTO	DPLLLTO	DPLLLCKF	DPLLLCKR
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
						DFLLULPNOLOCK	DFLLULPLOCK	DFLLULPRDY
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	7	6	5	4	3	2	1	0
				OSC16MRDY			XOSCFAIL	XOSCRDY
Access				R/W			R/W	R/W
Reset				0			0	0

Bit 28 – DFLLRCS DFLL Reference Clock Stopped
This flag is cleared by writing '1' to it.
This flag is set on 0-to-1 transition of the DFLL Reference Clock Stopped bit in the Status register (STATUS.DFLLRCS) and will generate an interrupt request if INTENSET.DFLLRCS is '1'.
Writing '0' to this bit has no effect.
Writing '1' to this bit clears the DFLL Reference Clock Stopped interrupt flag.

Bit 27 – DFLLLCKC DFLL Lock Coarse
This flag is cleared by writing '1' to it.
This flag is set on 0-to-1 transition of the DFLL Lock Coarse bit in the Status register (STATUS.DFLLLCKC) and will generate an interrupt request if INTENSET.DFLLLCKC is '1'.
Writing '0' to this bit has no effect.
Writing '1' to this bit clears the DFLL Lock Coarse interrupt flag.

Bit 26 – DFLLLCKF DFLL Lock Fine
This flag is cleared by writing '1' to it.
This flag is set on 0-to-1 transition of the DFLL Lock Fine bit in the Status register (STATUS.DFLLLCKF) and will generate an interrupt request if INTENSET.DFLLLCKF is '1'.
Writing '0' to this bit has no effect.
Writing '1' to this bit clears the DFLL Lock Fine interrupt flag.

Bit 25 – DFLLOOB DFLL Out Of Bounds
This flag is cleared by writing '1' to it.
This flag is set on 0-to-1 transition of the DFLL Out Of Bounds bit in the Status register (STATUS.DFLLOOB) and will generate an interrupt request if INTENSET.DFLLOOB is '1'.
Writing '0' to this bit has no effect.
Writing '1' to this bit clears the DFLL Out Of Bounds interrupt flag.

Bit 24 – DFLLRDY DFLL Ready
This flag is cleared by writing '1' to it.
This flag is set on 0-to-1 transition of the DFLL Ready bit in the Status register (STATUS.DFLLRDY) and will generate an interrupt request if INTENSET.DFLLRDY is '1'.
Writing '0' to this bit has no effect.
Writing '1' to this bit clears the DFLL Ready interrupt flag.

Bit 19 – DPLLLDRTO DPLL Loop Divider Ratio Update Complete
This flag is cleared by writing '1' to it.
This flag is set on a high to low transition of the DPLL Loop Divider Ratio Update Complete bit in the Status register (STATUS.DPLLLDRTO) and will generate an interrupt request if INTENSET.DPLLLDRTO is '1'.
Writing '0' to this bit has no effect.
Writing '1' to this bit clears the DPLL Loop Divider Ratio Update Complete interrupt flag.

Bit 18 – DPLLLTO DPLL Lock Timeout
This flag is cleared by writing '1' to it.
This flag is set on 0-to-1 transition of the DPLL Lock Timeout bit in the Status register (STATUS.DPLLLTO) and will generate an interrupt request if INTENSET.DPLLLTO is '1'.
Writing '0' to this bit has no effect.
Writing '1' to this bit clears the DPLL Lock Timeout interrupt flag.

Bit 17 – DPLLLCKF DPLL Lock Fall
This flag is cleared by writing '1' to it.
This flag is set on 0-to-1 transition of the DPLL Lock Fall bit in the Status register (STATUS.DPLLLCKF) and will generate an interrupt request if INTENSET.DPLLLCKF is '1'.
Writing '0' to this bit has no effect.
Writing '1' to this bit clears the DPLL Lock Fall interrupt flag.

Bit 16 – DPLLLCKR DPLL Lock Rise
This flag is cleared by writing '1' to it.
This flag is set on 0-to-1 transition of the DPLL Lock Rise bit in the Status register (STATUS.DPLLLCKR) and will generate an interrupt request if INTENSET.DPLLLCKR is '1'.
Writing '0' to this bit has no effect.
Writing '1' to this bit clears the DPLL Lock Rise interrupt flag.

Bit 10 – DFLLULPNOLOCK DFLLULP No Lock
This flag is cleared by writing '1' to it.
This flag is set on 0-to-1 transition of the DFLLULP No Lock bit in the Status register (STATUS.DFLLULPNOLOCK) and will generate an interrupt request if INTENSET.DFLLULPNOLOCK is '1'.
Writing '0' to this bit has no effect.
Writing '1' to this bit clears the DFLLULP No Lock interrupt flag.

Bit 9 – DFLLULPLOCK DFLLULP Lock
This flag is cleared by writing '1' to it.
This flag is set on 0-to-1 transition of the DFLLULP Lock bit in the Status register (STATUS.DFLLULPLOCK) and will generate an interrupt request if INTENSET.DFLLULPLOCK is '1'.
Writing '0' to this bit has no effect.
Writing '1' to this bit clears the DFLLULP Lock interrupt flag.

Bit 8 – DFLLULPRDY DFLLULP Ready
This flag is cleared by writing '1' to it.
This flag is set on 0-to-1 transition of the DFLLULP Ready bit in the Status register (STATUS.DFLLULPREADY) and will generate an interrupt request if INTENSET.DFLLULPREADY is '1'.
Writing '0' to this bit has no effect.
Writing '1' to this bit clears the DFLLULP Ready interrupt flag.

Bit 4 – OSC16MRDY OSC16M Ready
This flag is cleared by writing '1' to it.

This flag is set on 0-to-1 transition of the OSC16M Ready bit in the Status register (STATUS.OSC16MRDY) and will generate an interrupt request if INTENSET.OSC16MRDY is '1'.
Writing '0' to this bit has no effect.
Writing '1' to this bit clears the OSC16M Ready interrupt flag.

Bit 1 – XOSCFAIL XOSC Clock Failure

This flag is cleared by writing '1' to it.
This flag is set on a 0-to-1 transition of the XOSC Clock Failure bit in the Status register (STATUS.XOSCFAIL) and will generate an interrupt request if INTENSET.XOSCFAIL is '1'.
Writing '0' to this bit has no effect.
Writing '1' to this bit clears the XOSC Clock Failure interrupt flag.

Bit 0 – XOSCRDY XOSC Ready

This flag is cleared by writing '1' to it.
This flag is set on a 0-to-1 transition of the XOSC Ready bit in the Status register (STATUS.XOSCRDY) and will generate an interrupt request if INTENSET.XOSCRDY is '1'.
Writing '0' to this bit has no effect.
Writing '1' to this bit clears the XOSC Ready interrupt flag.

20.7.5 Status

Name: STATUS
Offset: 0x10
Reset: 0x00000010
Property: -

Bit	31	30	29	28	27	26	25	24
				DFLLRCS	DFLLLCKC	DFLLLCKF	DFLLOOB	DFLLRDY
Access				R	R	R	R	R
Reset				0	0	0	0	1

Bit	23	22	21	20	19	18	17	16
					DPLLLDRTO	DPLLLTO	DPLLLCKF	DPLLLCKR
Access					R	R	R	R
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
						DFLLULPNOLOCK	DFLLULPLOCK	DFLLULPRDY
Access						R	R	R
Reset						0	0	0

Bit	7	6	5	4	3	2	1	0
				OSC16MRDY		XOSCCKSW	XOSCFAIL	XOSCRDY
Access				R		R	R	R
Reset				1		0	0	0

Bit 28 – DFLLRCS DFLL Reference Clock Stopped

Value	Description
0	DFLL reference clock is running.
1	DFLL reference clock has stopped.

Bit 27 – DFLLLCKC DFLL Lock Coarse

Value	Description
0	No DFLL coarse lock detected.
1	DFLL coarse lock detected.

Bit 26 – DFLLLCKF DFLL Lock Fine

Value	Description
0	No DFLL fine lock detected.
1	DFLL fine lock detected.

Bit 25 – DFLLOOB DFLL Out Of Bounds

Value	Description
0	No DFLL Out Of Bounds detected.
1	DFLL Out Of Bounds detected.

Bit 24 – DFLLRDY DFLL Ready

Value	Description
0	DFLL registers update is ongoing. Registers update is requested through DFLLSYNC.READREQ, or after a write access in DFLLCTRL, DFLLVAL or DFLLMUL register.
1	DFLL registers are stable and ready for read/write access.

Bit 19 – DPLLLDRTO DPLL Loop Divider Ratio Update Complete

Value	Description
0	DPLL Loop Divider Ratio Update Complete not detected.
1	DPLL Loop Divider Ratio Update Complete detected.

Bit 18 – DPLLLTO DPLL Lock Timeout

Value	Description
0	DPLL Lock time-out not detected.
1	DPLL Lock time-out detected.

Bit 17 – DPLLLCKF DPLL Lock Fall

Value	Description
0	DPLL Lock fall edge not detected.
1	DPLL Lock fall edge detected.

Bit 16 – DPLLLCKR DPLL Lock Rise

Value	Description
0	DPLL Lock rise edge not detected.
1	DPLL Lock rise edge detected.

Bit 10 – DFLLULPNOLOCK DFLLULP No Lock

Value	Description
0	DFLLULP Tuner no lock state is not detected.
1	DFLLULP Tuner no lock state is detected.

Bit 9 – DFLLULPLOCK DFLLULP Lock

Value	Description
0	DFLLULP Tuner lock state is not detected.
1	DFLLULP Tuner lock state is detected.

Bit 8 – DFLLULPRDY DFLLULP Ready

Value	Description
0	DFLLULP is not ready.
1	DFLLULP is stable and ready to be used as a clock source.

Bit 4 – OSC16MRDY OSC16M Ready

Value	Description
0	OSC16M is not ready.
1	OSC16M is stable and ready to be used as a clock source.

Bit 2 – XOSCCKSW XOSC Clock Switch

Value	Description
0	XOSC is not switched and provides the external clock or crystal oscillator clock.
1	XOSC is switched and provides the safe clock.

Bit 1 – XOSCFAIL XOSC Clock Failure

Value	Description
0	No XOSC failure is detected.
1	A XOSC failure is detected.

Bit 0 – XOSCRDY XOSC Ready

Note: The XOSC Ready bit (STATUS.XOSCRDY) is not cleared when a clock failure is detected: STATUS.XOSCFAIL must always be checked before STATUS.XOSCRDY, and STATUS.XOSCRDY must always be ignored when STATUS.XOSCFAIL=1.

Value	Description
0	XOSC is not ready.
1	XOSC is stable and ready to be used as a clock source.

20.7.6 External Multipurpose Crystal Oscillator (XOSC) Control

Name: XOSCCTRL
Offset: 0x14
Reset: 0x0080
Property: PAC Write-Protection

Bit	15	14	13	12	11	10	9	8
	STARTUP[3:0]				AMPGC	GAIN[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
	ONDEMAND	RUNSTDBY		SWBEN	CFDEN	XTALEN	ENABLE	
Access	R/W	R/W		R/W	R/W	R/W	R/W	
Reset	1	0		0	0	0	0	

Bits 15:12 – STARTUP[3:0] Start-Up Time
This bit field selects the XOSC crystal oscillator stabilization time.

Important: This stabilization time is for guidance only. A major component of crystal start-up time is based on the second party crystal MFG parasitics that are outside the scope of this specification. If this is a major concern, the customer would need to characterize this based on their design choices.

Table 20-6. Stabilization Time for External Multipurpose Crystal Oscillator (1)

Value	Number of OSCULP32K Clock Cycles	Approximate Equivalent Time (2) (µs)
0x0	1	31
0x1	2	61
0x2	4	122
0x3	8	244
0x4	16	488
0x5	32	977
0x6	64	1953
0x7	128	3906
0x8	256	7813
0x9	512	15625
0xA	1024	31250
0xB	2048	62500
0xC	4096	125000
0xD	8192	250000
0xE	16384	500000
0xF	32768	1000000

Notes:
1. The OSCULP32K oscillator is used to clock the start-up counter.
2. Actual Start-Up time is the number of selected OSCULP32K cycles + 3 XOSC cycles.

Bit 11 – AMPGC Automatic Amplitude Gain Control
Note: The configuration of the Oscillator Gain is mandatory even if AMPGC feature is enabled at startup.

Value	Description
0	The automatic amplitude gain control is disabled.
1	The automatic amplitude gain control is enabled. Amplitude gain will be automatically adjusted during Crystal Oscillator operation.

Bits 10:8 – GAIN[2:0] Oscillator Gain

These bits select the gain for the oscillator. The listed maximum frequencies are recommendations, and might vary based on capacitive load and crystal characteristics. Those bits must be properly configured even when the Automatic Amplitude Gain Control is active.

Value	Recommended Max Frequency [MHz]
0x0	2
0x1	4
0x2	8
0x3	16
0x4	30
0x5-0x7	Reserved

Bit 7 – ONDEMAND On Demand Control

The On Demand operation mode allows the oscillator to be enabled or disabled, depending on peripheral clock requests.
If the ONDEMAND bit has been previously written to '1', the oscillator will be running only when requested by a peripheral. If there is no peripheral requesting the oscillator's clock source, the oscillator will be in a disabled state.
If On Demand is disabled, the oscillator will always be running when enabled.
In standby sleep mode, the On Demand operation is still active.

Value	Description
0	The oscillator is always on, if enabled.
1	The oscillator is enabled when a peripheral is requesting the oscillator to be used as a clock source. The oscillator is disabled if no peripheral is requesting the clock source.

Bit 6 – RUNSTDBY Run in Standby

This bit controls how the XOSC behaves during Standby Sleep mode, together with the ONDEMAND bit:

Value	Description
0	The XOSC is not running in Standby sleep mode if no peripheral requests the clock.
1	The XOSC is running in Standby sleep mode. If ONDEMAND=1, the XOSC will be running when a peripheral is requesting the clock. If ONDEMAND=0, the clock source will always be running in Standby sleep mode.

Bit 4 – SWBEN Clock Switch Back Enable

This bit controls the XOSC output switch back to the external clock or crystal oscillator in case of clock recovery:

Value	Description
0	The clock switch back is disabled.
1	The clock switch back is enabled. This bit is reset once the XOSC output clock is switched back to the external clock or crystal oscillator.

Bit 3 – CFDEN Clock Failure Detector Enable

This bit controls the clock failure detector:

Value	Description
0	The Clock Failure Detector is disabled.
1	the Clock Failure Detector is enabled.

Bit 2 – XTALEN Crystal Oscillator Enable

This bit controls the connections between the I/O pads and the external clock or crystal oscillator:

Value	Description
0	External clock connected on XIN. XOUT can be used as general-purpose I/O.
1	Crystal connected to XIN/XOUT.

Bit 1 – ENABLE Oscillator Enable

Value	Description
0	The oscillator is disabled.
1	The oscillator is enabled.

20.7.7 Clock Failure Detector Prescaler

Name: CFDPRESC
Offset: 0x16
Reset: 0x00
Property: PAC Write-Protection

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

Bits 2:0 – CFDPRESC[2:0] Clock Failure Detector Prescaler
These bits select the prescaler for the clock failure detector.
The OSC16M oscillator is used to clock the CFD prescaler. The CFD safe clock frequency is the OSC16M frequency divided by 2^CFDPRESC.

20.7.8 16MHz Internal Oscillator (OSC16M) Control

Name: OSC16MCTRL
Offset: 0x18
Reset: 0x82
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
	ONDEMAND	RUNSTDBY			FSEL[1:0]		ENABLE	
Access	R/W	R/W			R/W	R/W	R/W	
Reset	1	0			0	0	1	

Bit 7 – ONDEMAND On Demand Control

The On Demand operation mode allows the oscillator to be enabled or disabled depending on peripheral clock requests.
If the ONDEMAND bit has been previously written to '1', the oscillator will only be running when requested by a peripheral. If there is no peripheral requesting the oscillator's clock source, the oscillator will be in a disabled state.
If On Demand is disabled the oscillator will always be running when enabled.
In standby sleep mode, the On Demand operation is still active.

Value	Description
0	The oscillator is always on, if enabled.
1	The oscillator is enabled when a peripheral is requesting the oscillator to be used as a clock source. The oscillator is disabled if no peripheral is requesting the clock source.

Bit 6 – RUNSTDBY Run in Standby

This bit controls how the OSC16M behaves during standby sleep mode.

Value	Description
0	The OSC16M is disabled in standby sleep mode if no peripheral requests the clock.
1	The OSC16M is not stopped in standby sleep mode. If ONDEMAND=1, the OSC16M will be running when a peripheral is requesting the clock. If ONDEMAND=0, the clock source will always be running in standby sleep mode.

Bits 3:2 – FSEL[1:0] Oscillator Frequency Selection

These bits control the oscillator frequency range.

Value	Name	Description
0	4MHZ	4 MHz
1	8MHZ	8 MHz
2	12MHZ	12 MHz
3	16MHZ	16 MHz

Bit 1 – ENABLE Oscillator Enable

Value	Description
0	The oscillator is disabled.
1	The oscillator is enabled.

20.7.9 DFLLULP Control

Name: DFLLULPCTRL
Offset: 0x1C
Reset: 0x0504
Property: PAC Write-Protection, Enable-Protected Bits, Write-synchronized Bits

Bit	15	14	13	12	11	10	9	8
						DIV[2:0]		
Access						R/W	R/W	R/W
Reset						1	0	1

Bit	7	6	5	4	3	2	1	0
	ONDEMAND	RUNSTDBY	DITHER	SAFE	BINSE	Reserved	ENABLE	
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	0	0	0	0	0	1	0	

Bits 10:8 – DIV[2:0] Division Factor
This field defines the division factor for the output frequency of the DFLLULP which depends on PL0 and PL2 modes. This value which is determined and written during production, must be copied from the NVM software calibration row (DFLLULP Division Factor in PL0 or DFLLULP Division Factor in PL2) into the DFLLULPCTRL register by software. The value must be changed before switching on a new Performance Level mode (PL0 or PL2).
Note: This bit field is enable-protected. This bit field is not synchronized.

Value	Name	Description
0x0	DIV1	Frequency divided by 1
0x1	DIV2	Frequency divided by 2
0x2	DIV4	Frequency divided by 4
0x3	DIV8	Frequency divided by 8
0x4	DIV16	Frequency divided by 16
0x5	DIV32	Frequency divided by 32
0x6 – 0x7	-	Reserved

Bit 7 – ONDEMAND On Demand
The On Demand operation mode allows the oscillator to be enabled or disabled, depending on peripheral clock requests.
If the ONDEMAND bit has been previously written to '1', the oscillator will be running only when requested by a peripheral. If there is no peripheral requesting the oscillator's clock source, the oscillator will be in a disabled state.
If On Demand is disabled, the oscillator will always be running when enabled.
In standby sleep mode, the On Demand operation is still active.
Note: This bit is not enable-protected. This bit is not synchronized.

Bit 6 – RUNSTDBY Run in Standby
This bit controls how the DFLLULP behaves during standby sleep mode, together with the ONDEMAND bit.
Note: This bit is not enable-protected. This bit is not synchronized.

Bit 5 – DITHER Tuner Dither Mode
Note: This bit is enable-protected. This bit is not synchronized.

Value	Description
0	The dither mode is disabled.
1	The dither mode is enabled.

Bit 4 – SAFE Tuner Safe Mode
Note: This bit is enable-protected. This bit is not synchronized.

Value	Description
0	The clock output is not masked while binary search tuning is ongoing.
1	The clock output is masked while binary search tuning is ongoing (DFLLULPCTRL.BINSE = 1).

Bit 3 – BINSE Binary Search Enable

Note: This bit is enable-protected. This bit is not synchronized.

Value	Description
0	Binary search tuning is disabled. Maximum number of reference clock cycles to acquire lock is 256.
1	Binary search tuning is enabled. Maximum number of reference clock cycles to acquire lock is 8.

Bit 2 – Reserved Must Be Set to 1

Bit 2 must always be set to ‘1’ when programming the DFLLULPCTRL register.

Bit 1 – ENABLE Enable

Note: This bit is write-synchronized: DFLLULPSYNCBUSY.ENABLE must be checked to ensure the DFLLULPCTRL.ENABLE synchronization is complete.

Note: This bit is not enable-protected.

Value	Description
0	The DFLLULP is disabled.
1	The DFLLULP is enabled.

20.7.10 DFLLULP Dither Control

Name: DFLLULPDITHER
Offset: 0x1E
Reset: 0x00
Property: PAC Write-Protection, Enable-Protected

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

Bits 6:4 – PER[2:0] Dither Period
These bits define the number of reference clock periods over which dithering is applied.

Value	Name	Description
0x0	PER1	Dither over 1 reference clock period
0x1	PER2	Dither over 2 reference clock periods
0x2	PER4	Dither over 4 reference clock periods
0x3	PER8	Dither over 8 reference clock periods
0x4	PER16	Dither over 16 reference clock periods
0x5	PER32	Dither over 32 reference clock periods
0x6 – 0x7	-	Reserved

Bits 2:0 – STEP[2:0] Dither Step
This field defines the dithering step size.

Value	Name	Description
0x0	STEP1	Dither step = 1
0x1	STEP2	Dither step = 2
0x2	STEP4	Dither step = 4
0x3	STEP8	Dither step = 8
0x4 – 0x7	-	Reserved

20.7.11 DFLLULP Read Request

Name: DFLLULPRREQ
Offset: 0x1F
Reset: 0x00
Property: -

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

Bit 7 – RREQ Read Request
Writing a zero to this bit has no effect.
Writing a one to this bit requests synchronization of the DFLLULPDLY register with the current oscillator delay value and sets the Delay Busy bit in the Synchronization Busy register (DFLLULPSYNCBUSY.DELAY).
This bit is cleared automatically when synchronization is complete.

20.7.12 DFLLULP Delay Value

Name: DFLLULPDLY
Offset: 0x20
Reset: 0x00000080
Property: PAC Write-Protection, Write-Synchronized

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
	DELAY[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	0	0	0	0	0	0	0

Bits 7:0 – DELAY[7:0] Delay Value
Writing a value to this field sets the oscillator delay. A small value will produce a fast clock and a large value will produce a slow clock. Writing to this field will cause the tuner to start tuning from the written value. Reading this value will return the last written delay or the oscillator delay when a synchronization was requested from the DFLLULPRREQ register. Writing a value to this register while a write synchronization or a read request synchronization is on-going will have no effect and produce a PAC error.
Note: This bit field is write-synchronized: DFLLULPSYNCBUSY.DELAY must be checked to ensure the DFLLULPDLY.DELAY synchronization is complete.

20.7.13 DFLLULP Target Ratio

Name: DFLLULPRATIO
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
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
						RATIO[10:8]		
Access						R/W	R/W	R/W
Reset						0	0	0

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

Bits 10:0 – RATIO[10:0] Target Tuner Ratio
Writing a value to this field sets the target ratio between the DFLLULP output clock and the reference clock. The DFLLULPDLY.DELAY value will be updated in such a way that the target ratio and the actual ratio are as close as possible.

20.7.14 DFLLULP Synchronization Busy

Name: DFLLULPSYNCBUSY
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
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
					DELAY	Reserved	ENABLE	
Access					R	R	R	
Reset					0	0	0	

Bit 3 – DELAY Delay Register Synchronization Busy
This bit is cleared when the synchronization of DFLLULPDLY is complete.
This bit is set when the synchronization of DFLLULPDLY is started.
Writing this bit has no effect.

Bit 2 – Reserved Reserved

Bit 1 – ENABLE Enable Bit Synchronization Busy
This bit is cleared when the synchronization of DFLLULPCTRL.ENABLE is complete.
This bit is set when the synchronization of DFLLULPCTRL.ENABLE is started.
Writing this bit has no effect.

20.7.15 DFLL48M Control

Name: DFLLCTRL
Offset: 0x30
Reset: 0x0080
Property: PAC Write-Protection, Write-Synchronized

Note: This register is write-synchronized: STATUS.DFLLRDY must be checked to ensure the DFLLCTRL register synchronization is complete.

Bit	15	14	13	12	11	10	9	8
					WAITLOCK	BPLCKC	QLDIS	CCDIS
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ONDEMAND	RUNSTDBY	USBCRM	LLAW	STABLE	MODE	ENABLE	
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	1	0	0	0	0	0	0	

Bit 11 – WAITLOCK Wait Lock

This bit controls the DFLL output clock, depending on lock status.

Value	Description
0	Output clock before the DFLL is locked.
1	Output clock when DFLL is locked.

Bit 10 – BPLCKC Bypass Coarse Lock

This bit controls the coarse lock procedure.

Value	Description
0	Bypass coarse lock is disabled.
1	Bypass coarse lock is enabled.

Bit 9 – QLDIS Quick Lock Disable

Value	Description
0	Quick Lock is enabled.
1	Quick Lock is disabled.

Bit 8 – CCDIS Chill Cycle Disable

Value	Description
0	Chill Cycle is enabled.
1	Chill Cycle is disabled.

Bit 7 – ONDEMAND On Demand Control

The On Demand operation mode allows the DFLL to be enabled or disabled depending on peripheral clock requests. If the ONDEMAND bit has been previously written to '1', the DFLL will only be running when requested by a peripheral. If there is no peripheral requesting the DFLL clock source, the DFLL will be in a disabled state.
If On Demand is disabled, the DFLL will always be running when enabled.
In standby sleep mode, the On Demand operation is still active.

Value	Description
0	The DFLL is always on, if enabled.
1	The DFLL is enabled when a peripheral is requesting the DFLL to be used as a clock source. The DFLL is disabled if no peripheral is requesting the clock source.

Bit 6 – RUNSTDBY Run in Standby

This bit controls how the DFLL behaves during standby sleep mode:

Value	Description
0	The DFLL is disabled in standby sleep mode if no peripheral requests the clock.

Value	Description
1	The DFLL is not stopped in standby sleep mode. If ONDEMAND is one, the DFLL will be running when a peripheral is requesting the clock. If ONDEMAND is zero, the clock source will always be running in standby sleep mode.

Bit 5 – USBCRM USB Clock Recovery Mode

Value	Description
0	USB Clock Recovery Mode is disabled.
1	USB Clock Recovery Mode is enabled.

Bit 4 – LLAW Lose Lock After Wake

Value	Description
0	Locks will not be lost after waking up from sleep modes if the DFLL clock has been stopped.
1	Locks will be lost after waking up from sleep modes if the DFLL clock has been stopped.

Bit 3 – STABLE Stable DFLL Frequency

Value	Description
0	FINE calibration tracks changes in output frequency.
1	FINE calibration register value will be fixed after a fine lock.

Bit 2 – MODE Operating Mode Selection

Value	Description
0	The DFLL operates in open-loop operation.
1	The DFLL operates in closed-loop operation.

Bit 1 – ENABLE DFLL Enable

Due to synchronization, there is delay from updating the register until the peripheral is enabled/disabled. The value written to DFLLCTRL.ENABLE will read back immediately after written.

Value	Description
0	The DFLL oscillator is disabled.
1	The DFLL oscillator is enabled.

20.7.16 DFLL48M Value

Name: DFLLVAL
Offset: 0x34
Reset: 0x00000000
Property: PAC Write-Protection, Write-Synchronized, Read-Synchronized

Note: This register is read-synchronized: prior to any read access, this register must be synchronized by user by writing the according value to the DFLLSYNC register (DFLLSYNC.READREQ=1).

Note: This register is write-synchronized: STATUS.DFLLRDY must be checked to ensure the DFLLVAL register synchronization is complete.

Bit	31	30	29	28	27	26	25	24
	DIFF[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
	DIFF[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
	COARSE[5:0]						FINE[9:8]	
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
	FINE[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 – DIFF[15:0] Multiplication Ratio Difference
In closed-loop mode (DFLLCTRL.MODE=1), this bit group indicates the difference between the ideal number of DFLL cycles and the counted number of cycles. In open-loop mode, this value is not updated and hence, invalid.

Bits 15:10 – COARSE[5:0] Coarse Value
Set the value of the Coarse Calibration register. In closed-loop mode, this field is read-only.
Using the "DFLL48M COARSE" value from the Non Volatile Memory Software Calibration Area helps to output a frequency close to 48MHz.

Bits 9:0 – FINE[9:0] Fine Value
Set the value of the Fine Calibration register. In closed-loop mode, this field is read-only.

20.7.17 DFLL48M Multiplier

Name: DFLLMUL
Offset: 0x38
Reset: 0x00000000
Property: PAC Write-Protection, Write-Synchronized

Note: This register is write-synchronized: STATUS.DFLLRDY must be checked to ensure the DFLLMUL register synchronization is complete.

Bit	31	30	29	28	27	26	25	24
	CSTEP[5:0]						FSTEP[9:8]	
Access	R	R	R	R	R	R	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	FSTEP[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
	MUL[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
	MUL[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 31:26 – CSTEP[5:0] Coarse Maximum Step
This bit group indicates the maximum step size allowed during coarse adjustment in closed-loop mode. When adjusting to a new frequency, the expected output frequency overshoot depends on this step size.

Bits 25:16 – FSTEP[9:0] Fine Maximum Step
This bit group indicates the maximum step size allowed during fine adjustment in closed-loop mode. When adjusting to a new frequency, the expected output frequency overshoot depends on this step size.

Bits 15:0 – MUL[15:0] DFLL Multiply Factor
This field determines the ratio of the CLK_DFLL output frequency to the GCLK_DFLL48M input frequency. Writing to the MUL bits will cause locks to be lost and the fine calibration value to be reset to its midpoint.

20.7.18 DFLL48M Synchronization

Name: DFLLSYNC
Offset: 0x3C
Reset: 0x00
Property: PAC Write-Protection

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

Bit 7 – READREQ Read Request
To be able to read the current value of the DFLLVAL register in closed-loop mode, this bit must be written to '1'.

20.7.19 DPLL Control A

Name: DPLLCTRLA
Offset: 0x40
Reset: 0x80
Property: PAC Write-Protection, Write-Synchronized Bits

Bit	7	6	5	4	3	2	1	0
	ONDEMAND	RUNSTDBY					ENABLE	
Access	R/W	R/W					R/W	
Reset	1	0					0	

Bit 7 – ONDEMAND On Demand Clock Activation

The On Demand operation mode allows the DPLL to be enabled or disabled depending on peripheral clock requests. If the ONDEMAND bit has been previously written to '1', the DPLL will only be running when requested by a peripheral. If there is no peripheral requesting the DPLL's clock source, the DPLL will be in a disabled state.
If On Demand is disabled the DPLL will always be running when enabled.
In standby sleep mode, the On Demand operation is still active.
Note: This bit is not write-synchronized.

Value	Description
0	The DPLL is always on, if enabled.
1	The DPLL is enabled when a peripheral is requesting the DPLL to be used as a clock source. The DPLL is disabled if no peripheral is requesting the clock source.

Bit 6 – RUNSTDBY Run in Standby

This bit controls how the DPLL behaves during standby sleep mode:
Note: This bit is not write-synchronized.

Value	Description
0	The DPLL is disabled in standby sleep mode if no peripheral requests the clock.
1	The DPLL is not stopped in standby sleep mode. If ONDEMAND=1, the DPLL will be running when a peripheral is requesting the clock. If ONDEMAND=0, the clock source will always be running in standby sleep mode.

Bit 1 – ENABLE DPLL Enable

Note: This bit is write-synchronized: DPLLSYNCBUSY.ENABLE must be checked to ensure the DPLLCTRLA.ENABLE synchronization is complete.

Value	Description
0	The DPLL is disabled.
1	The DPLL is enabled.

20.7.20 DPLL Ratio Control

Name: DPLLRATIO
Offset: 0x44
Reset: 0x00000000
Property: PAC Write-Protection, Write-Synchronized

Note: This register is write-synchronized: DPLLSYNCBUSY.DPLLRATIO must be checked to ensure the DPLLRATIO register synchronization is complete.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					LDRFRAC[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
					LDR[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
	LDR[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 19:16 – LDRFRAC[3:0] Loop Divider Ratio Fractional Part

Writing these bits selects the fractional part of the frequency multiplier. Due to synchronization there is a delay between writing these bits and the effect on the DPLL output clock. The value written will read back immediately and the DPLLRATIO bit in the DPLL Synchronization Busy register (DPLLSYNCBUSY.DPLLRATIO) will be set. DPLLSYNCBUSY.DPLLRATIO will be cleared when the operation is completed.

Bits 11:0 – LDR[11:0] Loop Divider Ratio

Writing these bits selects the integer part of the frequency multiplier. The value written to these bits will read back immediately, and the DPLLRATIO bit in the DPLL Synchronization busy register (DPLLSYNCBUSY.DPLLRATIO), will be set. DPLLSYNCBUSY.DPLLRATIO will be cleared when the operation is completed.

20.7.21 DPLL Control B

Name: DPLLCTRLB
Offset: 0x48
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
						DIV[10:8]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	23	22	21	20	19	18	17	16
	DIV[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
				LBYPASS		LTIME[2:0]		
Access				R/W		R/W	R/W	R/W
Reset				0		0	0	0

Bit	7	6	5	4	3	2	1	0
			REFCLK[1:0]		WUF	LPEN	FILTER[1:0]	
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bits 26:16 – DIV[10:0] Clock Divider
These bits set the XOSC clock division factor and can be calculated with following formula:

$$f_{DIV} = \frac{f_{XOSC}}{2x(DIV + 1)}$$

Bit 12 – LBYPASS Lock Bypass

Value	Description
0	DPLL Lock signal drives the DPLL controller internal logic.
1	DPLL Lock signal is always asserted.

Bits 10:8 – LTIME[2:0] Lock Time
These bits select the lock time-out value:
Note: GCLK_FDPLL96M_32K is responsible for counting the user defined lock time (LTIME different from 0x0), hence must be enabled.

Value	Name	Description
0x0	DEFAULT	No time-out. Automatic lock.
0x1-0x3	Reserved	-
0x4	8MS	Time-out if no lock within 8ms
0x5	9MS	Time-out if no lock within 9ms
0x6	10MS	Time-out if no lock within 10ms
0x7	11MS	Time-out if no lock within 11ms

Bits 5:4 – REFCLK[1:0] Reference Clock Selection
Write these bits to select the DPLL clock reference:

Value	Name	Description
0x0	XOSC32K	XOSC32K clock reference
0x1	XOSC	XOSC clock reference
0x2	GCLK	GCLK_FDPLL96M clock reference

Value	Name	Description
0x3	Reserved	-

Bit 3 – WUF Wake Up Fast

Value	Description
0	DPLL clock is output after startup and lock time.
1	DPLL clock is output after startup time.

Bit 2 – LPEN Low-Power Enable

Value	Description
0	The low-power mode is disabled. Time to Digital Converter is enabled.
1	The low-power mode is enabled. Time to Digital Converter is disabled. This will improve power consumption but increase the output jitter.

Bits 1:0 – FILTER[1:0] Proportional Integral Filter Selection

These bits select the DPLL filter type:

Value	Name	Description
0x0	DEFAULT	Default filter mode
0x1	LBFILT	Low bandwidth filter
0x2	HBFILT	High bandwidth filter
0x3	HDFILT	High damping filter

20.7.22 DPLL Prescaler

Name: DPLLPRESC
Offset: 0x4C
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized

Note: This register is write-synchronized: DPLLSYNCBUSY.DPLLPRESC must be checked to ensure the DPLLPRESC register synchronization is complete.

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

Bits 1:0 – PRESC[1:0] Output Clock Prescaler
These bits define the output clock prescaler setting.

Value	Name	Description
0x0	DIV1	DPLL output is divided by 1
0x1	DIV2	DPLL output is divided by 2
0x2	DIV4	DPLL output is divided by 4
0x3	Reserved	-

20.7.23 DPLL Synchronization Busy

Name: DPLLSYNCBUSY
Offset: 0x50
Reset: 0x00
Property: –

Bit	7	6	5	4	3	2	1	0
					DPLLPRESC	DPLLRATIO	ENABLE	
Access					R	R	R	
Reset					0	0	0	

Bit 3 – DPLLPRESC DPLL Prescaler Synchronization Status

Value	Description
0	The DPLLRESC register has been synchronized.
1	The DPLLRESC register value has changed and its synchronization is in progress.

Bit 2 – DPLLRATIO DPLL Loop Divider Ratio Synchronization Status

Value	Description
0	The DPLLRATIO register has been synchronized.
1	The DPLLRATIO register value has changed and its synchronization is in progress.

Bit 1 – ENABLE DPLL Enable Synchronization Status

Value	Description
0	The DPLLCTRLA.ENABLE bit has been synchronized.
1	The DPLLCTRLA.ENABLE bit value has changed and its synchronization is in progress.

20.7.24 DPLL Status

Name: DPLLSTATUS
Offset: 0x54
Reset: 0x00
Property: –

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

Bit 1 – CLKRDY DPLL Clock Ready

Value	Description
0	The DPLL output clock is off.
1	The DPLL output clock in on.

Bit 0 – LOCK DPLL Lock

Value	Description
0	The DPLL Lock signal is cleared, when the DPLL is disabled or when the DPLL is trying to reach the target frequency.
1	The DPLL Lock signal is asserted when the desired frequency is reached.

21. Supply Controller (SUPC)

21.1 Overview

The Supply Controller (SUPC) manages the voltage references and the power supplies of the device.

The SUPC controls:

- The voltage regulators (VDDCORE domain)
- The FDPLL96M, DFLL48M and DFLLULP clock sources (VDDPLL domain)

It sets the voltage regulators according to the sleep modes, or the user configuration. In active mode, the voltage regulators can be selected on the fly between LDO (low-dropout) type regulator or Buck converter (BUCK).

The SUPC embeds different features to monitor, warn and reset the device:

- A Power-on Reset (POR) on VDD and AVDD:
  - Monitoring is always activated, including during device start-up or during any sleep modes.
  - Having VDD/AVDD below a fixed threshold voltage will reset the whole device.

  Note: POR minimum and maximum threshold voltages can be found in Power Supply Electrical Specifications in the Electrical Characteristics chapter.
- A Brown-out Detector (BOD33) on VDD and AVDD:
  - The BOD33 can monitor VDD/AVDD continuously (continuous mode) or periodically (sampled mode) with a programmable sample frequency in active mode as in any sleep modes.
  - A programmable threshold loaded from the NVM User Row is used to trigger an interrupt and/or reset the whole device.
- A Brown-out Detector (BOD12) on VDDCORE.
- A Brown-out Detector (BOD12PLL) on VDDPLL.

⚠ CAUTION BOD12 and BOD12PLL are calibrated in production and their calibration parameters are stored in the NVM User Row. These data must not be changed to ensure correct device behavior.

The SUPC generates also a selectable reference voltage which can be used by analog modules like the ADC or DAC.

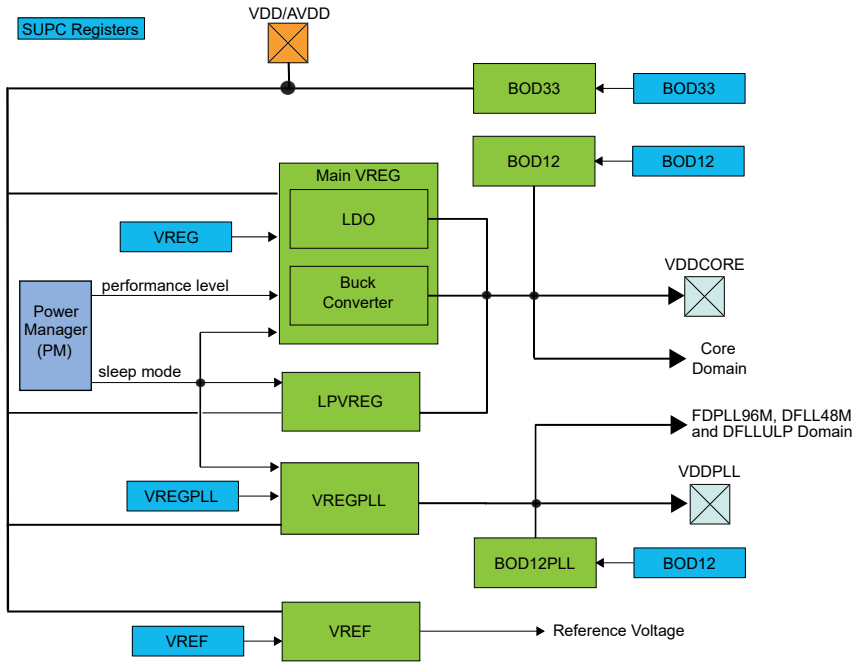
21.2 Features

- Voltage Regulators System
  - Main voltage regulator: LDO or Buck Converter in Active mode (MAINVREG)
  - LDO regulator for the FDPLL96M, DFLL48M and DFLLULP clock sources (VREGPLL)
  - Low-Power voltage regulator in Standby mode (LPVREG)
  - Adjustable VDDCORE and VDDPLL to the Sleep mode or the performance level
  - Controlled VDDCORE and VDDPLL voltage slope when changing VDDCORE and VDDPLL respectively
- Voltage Reference System
  - Reference voltage for ADC and DAC
- 3.3V Brown-Out Detector (BOD33)
  - Programmable threshold
  - Threshold value loaded from NVM User Row at start-up
  - Triggers resets or interrupts or event. Action loaded from NVM User Row
  - Operating modes:
    - Continuous mode
    - Sampled mode for low power applications with programmable sample frequency

  - Hysteresis value from Flash User Calibration
- 1.2V Brown-Out Detectors (BOD12 and BOD12PLL)
  - Internal non-configurable Brown-Out Detector for VDDCORE
  - Internal non-configurable Brown-Out Detector for VDDPLL

21.3 Block Diagram

Figure 21-1. SUPC Block Diagram



21.4 Peripheral Dependencies

Table 21-1. SUPC Configuration Summary

Peripheral name	Base address	NVIC IRQ Index	MCLK AHB/APB Clocks	PAC Peripheral Identifier Index (PAC.WRCTRL)	Events (EVSYS)		Power Domain (PM.STDBYCFG)
					Users (EVSYS.USER)	Generators (EVSYS.CHANNEL)	
SUPC	0x40001800	0: BOD33RDY,BOD33DET, B33SRDY, VREGRDY, VCORERDY, ULPVREFRDY,VCOREPLLRDY	CLK_SUPC_APB	6	—	3: BOD33DET	PDAO

21.5 Functional Description

21.5.1 Voltage Regulators

Two embedded voltage regulators are used to provide VDDCORE to the device:

- The Main voltage regulator (MAINVREG)
- The Low-Power voltage regulator (LPVREG) used when the device is in Standby Sleep mode

The Main Voltage Regulator has two modes:

- Linear (LDO) mode: The default mode after reset.
- Switching (BUCK) mode: The most power efficient mode when the CPU and peripherals are running (Active mode) and which requires an external inductor between VDDOUT and VDDCORE.
  Note: In Active mode, the voltage regulator can be selected on the fly between LDO (low-dropout) type regulator and Buck converter using the Supply Controller (SUPC)

One embedded voltage regulator (VREGPLL) is used to provide VDDPLL to the device.

21.5.1.1 Enabling, Disabling, and Resetting

The LDO main voltage regulator is enabled after any Reset. The main voltage regulator (MAINVREG) can not be disabled, hence the Enable bit in the VREG register (VREG.ENABLE) must be set to one. The main voltage regulator output supply level is automatically defined by the performance level or the Sleep mode selected in the Power Manager module.

The FDPLL96M/DFLL48M/DFLLULP voltage regulator (VREGPLL) is enabled by writing the Enable bit in VREGPLL register (VREGPLL.ENABLE) to one. As for the main regulator, the VREGPLL output supply level is automatically defined by the performance level or the sleep mode selected in the Power Manager module.

The VREGPLL is disabled by writing the Enable bit in VREGPLL register (VREGPLL.ENABLE) to zero. When the regulator is disabled, the power domain power domain is shut-off and pending FDPLL96M/DFLL48M/DFLLULP clock requests will not be serviced. As a consequence, it is recommended to shut-down the peripherals using these clock sources before disabling the VREGPLL.

21.5.1.2 Initialization

After a Reset, the LDO voltage regulator supplying VDDCORE is enabled. VREGPLL is disabled.

21.5.1.3 Selecting MAINVREG and VREGPLL Voltage Regulators

In Active mode, the type of the main voltage regulator supplying VDDCORE can be switched on the fly. The two alternatives are a LDO regulator and a Buck converter.

The main voltage regulator switching sequences are as follows:

- The user changes the value of the Voltage Regulator Selection bit in the Voltage Regulator System Control register (VREG.SEL)
- The start of the switching sequence is indicated by clearing the Voltage Regulator Ready bit in the STATUS register (STATUS.VREGRDY=0)
- Once the switching sequence is completed, STATUS.VREGRDY will read '1'

The Voltage Regulator Ready (VREGRDY) interrupt can also be used to detect a zero-to-one transition of the STATUS.VREGRDY bit.

If one of the FDPLL96M DFLL48M or DFLLULP clock sources must be used by the application, the user must enable the VREGPLL regulator. To configure in a safe way the system, the user must follow the sequence:

- Wait until the main regulator is ready (STATUS.VREGRDY = 1)
- Optionally enable the PLL Oscillators Core Voltage Regulator Ready Interrupt (INTENSET.VCOREPLLRDY)
- Enable the PLL oscillators regulator (VREGPLL.ENABLE)
- Wait for VCOREPLLRDY interrupt or poll the VCOREPLLRDY bit STATUS register (STATUS.VCOREPLLRDY)

The clock sources must be enabled only when the VDDPLL supply is established. When enabled, the VREGPLL regulator remains enabled in all sleep modes, except Standby sleep mode, where the regulator supports the Sleep Walking capability.

21.5.1.4 Voltage Scaling Control

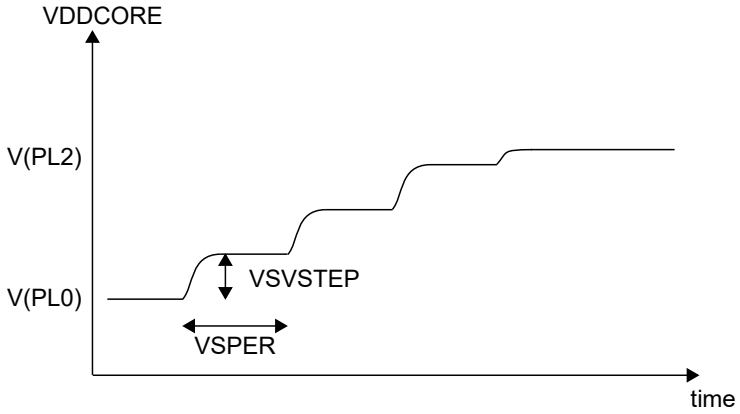
The VDDCORE supply will change under certain circumstances:

- When a new performance level (PL) is set
- When the Standby Sleep mode is entered or left
- When a sleepwalking task is requested in Standby Sleep mode

To prevent high peak current on the main power supply and to have a smooth transition of VDDCORE, both the voltage scaling step size and the voltage scaling frequency can be controlled: VDDCORE is changed by the selected step size of the selected period until the target voltage is reached.

The Voltage Scaling Voltage Step field is in the VREG register, VREG.VSVSTEP. The Voltage Scaling Period field is VREG.VSPER.

The following waveform shows an example of changing performance level from PL0 to PL2.



Setting VREG.VSVSTEP to the maximum value allows to transition in one voltage step.

The STATUS.VCORERDY bit is set to '1' as soon as the VDDCORE voltage has reached the target voltage. During voltage transition, STATUS.VCORERDY will read '0'. The Voltage Ready interrupt (VCORERDY) can be used to detect a 0-to-1 transition of STATUS.VCORERDY, see also 21.5.4. Interrupts.

When entering the Standby Sleep mode and when no sleepwalking task is requested, the VDDCORE Voltage scaling control is not used.

Note: If VREGPLL voltage regulator is enabled (VREGPLL.ENABLE = 1), the Voltage Scaling Voltage Step (VREG.VSVSTEP) and the Voltage Scaling Period (VREG.VSPER) settings also apply to VREGPLL. The STATUS.VCOREPLLRDY status bit or VCOREPLLRDY interrupt flag are set when the VDDPLL reached the voltage target.

21.5.1.5 Sleep Mode Operation

In Standby mode, the low-power voltage regulator (LPVREG) is used to supply VDDCORE.

When the Run in Standby bit in the VREG register (VREG.RUNSTDBY) is written to '1', VDDCORE is supplied by the main voltage regulator. Depending on the Standby in PL0 bit in the Voltage Regulator register (VREG.STDBYPL0), the VDDCORE level is either set to the PL0 voltage level, or remains in the current performance level.

Table 21-2. VDDCORE Level in Standby Mode

VREG.RUNSTDBY	VREG.STDBYPL0	VDDCORE Supply in Standby Mode
0	-	LPVREG
1	0	MAINVREG in current performance level(1)
1	1	MAINVREG in PL0

Note:

1. When the device is in PL0 but VREG.STDBYPL0=0, the MAINVREG is operating in normal power mode. To minimize power consumption, operate MAINVREG in PL0 mode by selecting VREG.STDBYPL0=1.

By writing the Low-Power mode Efficiency bit in the VREG register (VREG.LPEFF) to '1', the efficiency of the regulator in LPVREG can be improved when the application uses a limited VDD/AVDD range (2.5 to 3.63V). It is also possible to use the BOD33 in order to monitor the VDD/AVDD and change this LPEFF value on the fly according to VDD/AVDD level.

The VREGPLL voltage regulator has a similar behavior as the main voltage regulator, with some differences:

- VREGPLL continuously run in standby mode if the Run in Standby bit in the VREGPLL register is (VREGPLL.RUNSTDBY=1) set and the regulator is enabled (VREGPLL.ENABLE = 1). This mode allows faster start-up times when sleep walking tasks must be executed.

- When VREGPLL is enabled (VREGPLL.ENABLE = 1) and Standby bit in the VREGPLL register is cleared (VREGPLL.RUNSTDBY=0), the VREGPLL is internally enabled when a sleep walking task must be executed, using the DFLLULP, DFLL or FDPLL clock sources. In this operating mode, the start-up time needed before executing a sleep walking task will be longer, as the voltage level must be stable before execution starts.

Table 21-3. VDDPLL Level in Standby Mode

VREGPLL.RUNSTDBY	VREG.STDBYPL0	VDDPLL Supply in Standby Mode
0	-	ON/OFF(1)
1	0	VREGPLL in current performance level(2)
1	1	VREGPLL in PL0

Notes:

1. Depending on sleep walking requests and VREGPLL.RUNSTDBY bit setting.
2. When the device is in PL0 but VREG.STDBYPL0=0, the VREGPLL is operating in normal power mode. To minimize power consumption, operate VREGPLL in PL0 mode by selecting VREG.STDBYPL0=1.

21.5.2 Voltage Reference System Operation

The reference voltages are generated by a functional block DETREF inside of the SUPC. DETREF is providing a fixed-voltage source, BANDGAP=1.1V, and a variable voltage, INTREF.

For further information, refer to 43.1. Reference Voltages.

21.5.2.1 Initialization

The voltage reference output is disabled after any Reset.

21.5.2.2 Enabling, Disabling, and Resetting

The voltage reference output is enabled/disabled by setting/clearing the Voltage Reference Output Enable bit in the Voltage Reference register (VREF.VREFOE).

21.5.2.3 Selecting a Voltage Reference

The Voltage Reference Selection bit field in the VREF register (VREF.SEL) selects the voltage of INTREF to be applied to analog modules, e.g. the ADC.

21.5.2.4 Sleep Mode Operation

The Voltage Reference output behavior during sleep mode can be configured using the Run in Standby bit and the On Demand bit in the Voltage Reference register (VREF.RUNSTDBY, VREF.ONDEMAND), see the following table:

Table 21-4. VREF Sleep Mode Operation

VREF.ONDEMAND	VREF.RUNSTDBY	Voltage Reference Sleep behavior
-	-	Disable
0	0	Always run in all sleep modes *except* standby sleep mode
0	1	Always run in all sleep modes *including* standby sleep mode
1	0	Only run if requested by the ADC, in all sleep modes *except* standby sleep mode
1	1	Only run if requested by the ADC, in all sleep modes *including* standby sleep mode

21.5.3 Brown-Out Detectors

21.5.3.1 Initialization

Before a Brown-Out Detector (BOD33) is enabled, it must be configured, as outlined by the following:

- Set the BOD threshold level (BOD33.LEVEL)
- Set the configuration in Active, Standby (BOD33.ACTION, BOD33.STDBYCFG)
- Set the prescaling value if the BOD will run in sampling mode (BOD33.PSEL)
- Set the action and hysteresis (BOD33.ACTION and BOD33.HYST)

The BOD33 register is Enable-Protected (excluding BOD33.ENABLE bit), meaning that they can only be written when the BOD is disabled (BOD33.ENABLE=0 and STATUS.B33SRDY=0). As long as the Enable bit is '1', any writes to Enable-Protected registers (EVCTRL) will be discarded, and an APB error will be generated. The Enable bits are not Enable-Protected.

21.5.3.2 Enabling, Disabling, and Resetting

After power or user reset, the BOD33 and BOD12 register values are loaded from the NVM User Page.

The BOD33 is enabled by writing a '1' to the Enable bit in the BOD control register (BOD33.ENABLE). The BOD33 is disabled by writing a '0' to the BOD33.ENABLE.

21.5.3.3 3.3V Brown-Out Detector (BOD33)

The 3.3V Brown-Out Detector (BOD33) is able to monitor the VDD/AVDD supply and compares the voltage with the brown-out threshold level set in the BOD33 Level field (BOD33.LEVEL) in the BOD33 register.

When VDD/AVDD crosses below the brown-out threshold level, the BOD33 can generate either an interrupt or a Reset, depending on the BOD33 Action bit field (BOD33.ACTION).

The BOD33 detection status can be read from the BOD33 Detection bit in the Status register (STATUS.BOD33DET).

At start-up or at Power-On Reset (POR), the BOD33 register values are loaded from the NVM User Row.

21.5.3.4 1.2V Brown-Out Detector (BOD12 and BOD12PLL)

The BOD12 and BOD12PLL are calibrated in production and their calibration configuration is stored in the NVM User Row. This configuration must not be changed to assure their correct behavior. The BOD12 and BOD12PLL generate a reset when 1.2V crosses below the preset brown-out level. The BOD12 and BOD12PLL are always disabled in Standby Sleep mode.

21.5.3.5 Continuous Mode

Continuous mode is the default mode for BOD33.

The BOD33 is continuously monitoring the VDD/AVDD supply voltage if it is enabled (BOD33.ENABLE=1) and if the BOD33 Configuration bit in the BOD33 register is cleared (BOD33.ACTCFG=0 for active mode, BOD33.STDBYCFG=0 for standby mode).

21.5.3.6 Sampling Mode

The Sampling Mode is a low-power mode where the BOD33 is being repeatedly enabled on a sampling clock's ticks. The BOD33 will monitor the supply voltage for a short period of time and then go to a low-power disabled state until the next sampling clock tick.

Sampling mode is enabled in Active mode for BOD33 by writing the ACTCFG bit (BOD33.ACTCFG=1). Sampling mode is enabled in Standby mode by writing to the STDBYCFG bit (BOD33.STBYCFG=1). The frequency of the clock ticks ($F_{clksampling}$) is controlled by the Prescaler Select bit groups in the BOD33 register (BOD33.PSEL).

$$F_{clksampling} = \frac{F_{clkprescaler}}{2^{(\mathrm{PSEL}+1)}}$$

The prescaler signal ($F_{clkprescaler}$) is a 1024 Hz clock, output by the 32.768 kHz Ultra Low Power Oscillator OSCULP32K.

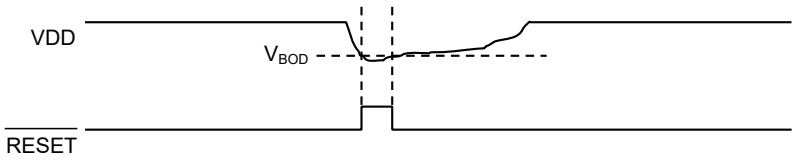
As the sampling clock is different from the APB clock domain, synchronization among the clocks is necessary. See also 21.5.6. Synchronization.

21.5.3.7 Hysteresis

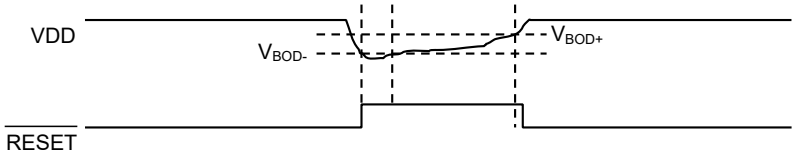
A hysteresis on the trigger threshold of a BOD will reduce the sensitivity to ripples on the monitored voltage: instead of switching $\overline{\text{RESET}}$ at each crossing of V_{BOD}, the thresholds for switching $\overline{\text{RESET}}$ on and off are separated (V_{BOD-} and V_{BOD+}, respectively).

Figure 21-2. BOD Hysteresis Principle

Hysteresis OFF:



Hysteresis ON:



Enabling the BOD33 hysteresis by writing the Hysteresis bit in the BOD33 register (BOD33.HYST) to '1' will add hysteresis to the BOD33 threshold level.

The hysteresis functionality can be used in both Continuous and Sampling Mode.

21.5.3.8 Sleep Mode Operation

21.5.3.8.1 Standby Mode

The BOD33 can be used in standby mode if the BOD is enabled and the corresponding Run in Standby bit is written to '1' (BOD33.RUNSTDBY).

The BOD33 can be configured to work in either Continuous or Sampling Mode by writing a '1' to the Configuration in Standby Sleep Mode bit (BOD33.STDBYCFG).

21.5.4 Interrupts

The SUPC has the following interrupt sources, which are either synchronous or asynchronous wake-up sources:

- VDDCORE Voltage Ready (VCORERDY), asynchronous
- Voltage Regulator Ready (VREGRDY) asynchronous
- VDDPLL Voltage Regulator Ready (VCOREPLLRDY) asynchronous
- BOD33 Ready (BOD33RDY), synchronous
- BOD33 Detection (BOD33DET), asynchronous
- BOD33 Synchronization Ready (B33SRDY), synchronous
- DFLLULP Low-Power Voltage Reference Ready (ULPVREFRDY), asynchronous

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).

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 SUPC is reset. See the INTFLAG register for details on how to clear interrupt flags. The SUPC 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: Interrupts must be globally enabled for interrupt requests to be generated.

21.5.5 Events

The SUPC can generate the following output event:

- BOD33 Detection (BOD33DET): Generated when the VDD/AVDD crosses below the brown-out threshold level.

Writing a one to the Event Output bit in the Event Control Register (EVCTRL.BOD33DETEO) enables the corresponding output event. Writing a zero to this bit disables the corresponding output event. Refer to the Event System chapter for details on configuring the event system.

21.5.6 Synchronization

The prescaler counters that are used to trigger brown-out detections operate asynchronously from the peripheral bus. As a consequence, the BOD33 Enable bit (BOD33.ENABLE) need synchronization when written.

The Write-Synchronization of the Enable bit is triggered by writing a '1' to the Enable bit of the BOD33 Control register. The Synchronization Ready bit (STATUS.B33SRDY) in the STATUS register will be cleared when the Write-Synchronization starts, and set again when the Write-Synchronization is complete. Writing to the same register while the Write-Synchronization is ongoing (STATUS.B33SRDY is '0') will generate a PAC error without stalling the APB bus.

21.5.7 Debug Operation

When the CPU is halted in debug mode, the SUPC continues normal operation. If the SUPC 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.

If debugger cold-plugging is detected by the system, BOD33, BOD12 and BOD12PLL resets will be masked. The BOD resets keep running under hot-plugging. This allows to correct a BOD33 user level too high for the available supply.

21.5.8 Low-Power VREF in Active Mode

During active functional mode, the Brown-out Detector (BOD33) and the main voltage regulator (VREG) can reduce their power consumption by using a low-power voltage reference (ULPVREF).

The low-power voltage reference is started after power-up and is available when the ULPVREFRDY bit in the STATUS register is high. The ULPVREF Ready (ULPVREFRDY) interrupt can also be used to detect a zero-to-one transition of the STATUS.ULPVREFRDY bit.

Writing the VREF bit in the BOD33 register to '1' selects ULPVREF as voltage reference for the BOD33.

If the chip operated in PL0 ((PM->PLCFG.PLSEL=0) or Performance Level is disabled (PM->PLCFG.PLDIS=1), writing the VREFSEL bit in the VREG register to '1' selects ULPVREF as voltage reference for the main voltage regulator.
Note: The ULPVREF reference cannot be used in PL2 mode.

21.6 Register Summary

Refer to the Registers Description section for more details on register properties and access permissions.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	INTENCLR	31:24								
		23:16				Reserved	Reserved	VCOREPLLRDY		
		15:8					ULPVREFRDY	VCORERDY		VREGRDY
		7:0			Reserved	Reserved	Reserved	B33SRDY	BOD33DET	BOD33RDY
0x04	INTENSET	31:24								
		23:16				Reserved	Reserved	VCOREPLLRDY		
		15:8					ULPVREFRDY	VCORERDY		VREGRDY
		7:0			Reserved	Reserved	Reserved	B33SRDY	BOD33DET	BOD33RDY
0x08	INTFLAG	31:24								
		23:16				Reserved	Reserved	VCOREPLLRDY		
		15:8					ULPVREFRDY	VCORERDY		VREGRDY
		7:0			Reserved	Reserved	Reserved	B33SRDY	BOD33DET	BOD33RDY
0x0C	STATUS	31:24								
		23:16				Reserved	Reserved	VCOREPLLRDY		
		15:8			Reserved	ULPVREFRDY		VCORERDY		VREGRDY
		7:0			Reserved	Reserved	Reserved	B33SRDY	BOD33DET	BOD33RDY
0x10	BOD33	31:24								
		23:16			LEVEL[5:0]					
		15:8	PSEL[3:0]				VREFSEL			ACTCFG
		7:0		RUNSTDBY	STDBYCFG	ACTION[1:0]		HYST	ENABLE	
0x14 ... 0x17	Reserved									
0x18	VREG	31:24	VSPER[7:0]							
		23:16					VSVSTEP[3:0]			
		15:8							VREFSEL	LPEFF
		7:0		RUNSTDBY	STDBYPL0		SEL[1:0]		ENABLE	
0x1C	VREF	31:24								
		23:16					SEL[3:0]			
		15:8								
		7:0	ONDEMAND	RUNSTDBY			Reserved	VREFOE	Reserved	
0x20	VREGPLL	31:24								
		23:16								
		15:8					STARTUP[3:0]			
		7:0		RUNSTDBY					ENABLE	
0x24 ... 0x2B	Reserved									
0x2C	EVCTRL	31:24								
		23:16								
		15:8								
		7:0				Reserved			BOD33DETEO	

21.6.1 Interrupt Enable Clear

Name: INTENCLR
Offset: 0x00
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
				Reserved	Reserved	VCOREPLLRDY		
Access				R/W	R/W	R/W		
Reset				0	0	0		

Bit	15	14	13	12	11	10	9	8
					ULPVREFRDY	VCORERDY		VREGRDY
Access					R/W	R/W		R/W
Reset					0	0		0

Bit	7	6	5	4	3	2	1	0
			Reserved	Reserved	Reserved	B33SRDY	BOD33DET	BOD33RDY
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit 20 – Reserved

Bit 19 – Reserved

Bit 18 – VCOREPLLRDY VDDPLL Voltage Ready Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the VDDPLL Voltage Ready Interrupt Enable bit, which disables the VDDPLL Voltage Ready interrupt.

Value	Description
0	The VDDPLL Voltage Ready interrupt is disabled.
1	The VDDPLL Voltage Ready interrupt is enabled, and an interrupt request will be generated when the VCOREPLLRDY Voltage Ready Interrupt flag is set.

Bit 11 – ULPVREFRDY Low Power Voltage Reference Ready Interrupt Enable
Writing a '0' to this bit has no effect.
The ULPVREFRDY bit will clear on a zero-to-one transition of the Low Power Voltage Reference Ready bit in the Status register (STATUS.ULPVREFRDY).

Value	Description
0	The Low Power Voltage Ready interrupt is disabled.
1	The Low Power Voltage Ready interrupt is enabled and an interrupt request will be generated when the ULPVREFRDY Voltage Interrupt Flag is set.

Bit 10 – VCORERDY VDDCORE Voltage Ready Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the VDDCORE Voltage Ready Interrupt Enable bit, which disables the VDDCORE Ready interrupt.

Value	Description
0	The VDDCORE Voltage Ready interrupt is disabled.
1	The VDDCORE Voltage Ready interrupt is enabled and an interrupt request will be generated when the VCORERDY Voltage Interrupt Flag is set.

Bit 8 – VREGRDY Voltage Regulator Ready Interrupt Enable

Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Voltage Regulator Ready Interrupt Enable bit, which disables the Voltage Regulator Ready interrupt.

Value	Description
0	The Voltage Regulator Ready interrupt is disabled.
1	The Voltage Regulator Ready interrupt is enabled and an interrupt request will be generated when the Voltage Regulator Ready Interrupt Flag is set.

Bit 5 – Reserved

Bit 4 – Reserved

Bit 3 – Reserved

Bit 2 – B33SRDY BOD33 Synchronization Ready Interrupt Enable

Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the BOD33 Synchronization Ready Interrupt Enable bit, which disables the BOD33 Synchronization Ready interrupt.

Value	Description
0	The BOD33 Synchronization Ready interrupt is disabled.
1	The BOD33 Synchronization Ready interrupt is enabled, and an interrupt request will be generated when the BOD33 Synchronization Ready Interrupt flag is set.

Bit 1 – BOD33DET BOD33 Detection Interrupt Enable

Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the BOD33 Detection Interrupt Enable bit, which disables the BOD33 Detection interrupt.

Value	Description
0	The BOD33 Detection interrupt is disabled.
1	The BOD33 Detection interrupt is enabled, and an interrupt request will be generated when the BOD33 Detection Interrupt flag is set.

Bit 0 – BOD33RDY BOD33 Ready Interrupt Enable

Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the BOD33 Ready Interrupt Enable bit, which disables the BOD33 Ready interrupt.

Value	Description
0	The BOD33 Ready interrupt is disabled.
1	The BOD33 Ready interrupt is enabled, and an interrupt request will be generated when the BOD33 Ready Interrupt flag is set.

21.6.2 Interrupt Enable Set

Name: INTENSET
Offset: 0x04
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 Clear register (INTENCLR).

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
				Reserved	Reserved	VCOREPLLRDY		
Access				R/W	R/W	R/W		
Reset				0	0	0		

Bit	15	14	13	12	11	10	9	8
					ULPVREFRDY	VCORERDY		VREGRDY
Access					R/W	R/W		R/W
Reset					0	0		0

Bit	7	6	5	4	3	2	1	0
			Reserved	Reserved	Reserved	B33SRDY	BOD33DET	BOD33RDY
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit 20 – Reserved

Bit 19 – Reserved

Bit 18 – VCOREPLLRDY VDDPLL Voltage Ready Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the VDDPLL Voltage Ready Interrupt Enable bit, which enables the Voltage Regulator PLL Ready interrupt.

Value	Description
0	The VDDPLL Voltage Ready interrupt is disabled.
1	The VDDPLL Voltage Ready interrupt is enabled, and an interrupt request will be generated when the VDDPLL Voltage Ready Interrupt flag is set.

Bit 11 – ULPVREFRDY Low Power Voltage Reference Ready Interrupt Enable
Writing a '0' to this bit has no effect.
The ULPVREFRDY bit is set on a zero-to-one transition of the Low Power Voltage Reference Ready bit in the Status register (STATUS.ULPVREFRDY).

Value	Description
0	The Low Power Voltage Ready interrupt is disabled.
1	The Low Power Voltage Ready interrupt is enabled and an interrupt request will be generated when the ULPVREFRDY Voltage Interrupt Flag is set.

Bit 10 – VCORERDY VDDCORE Voltage Ready Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the VDDCORE Voltage Ready Interrupt Enable bit, which enables the VDDCORE Voltage Ready interrupt.

Value	Description
0	The VDDCORE Voltage Ready interrupt is disabled.
1	The VDDCORE Voltage Ready interrupt is enabled and an interrupt request will be generated when the VCORERDY Voltage Interrupt Flag is set.

Bit 8 – VREGRDY Voltage Regulator Ready Interrupt Enable

Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Voltage Regulator Ready Interrupt Enable bit, which enables the Voltage Regulator Ready interrupt.

Value	Description
0	The Voltage Regulator Ready interrupt is disabled.
1	The Voltage Regulator Ready interrupt is enabled and an interrupt request will be generated when the Voltage Regulator Ready Interrupt Flag is set.

Bit 5 – Reserved

Bit 4 – Reserved

Bit 3 – Reserved

Bit 2 – B33SRDY BOD33 Synchronization Ready Interrupt Enable

Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the BOD33 Synchronization Ready Interrupt Enable bit, which enables the BOD33 Synchronization Ready interrupt.

Value	Description
0	The BOD33 Synchronization Ready interrupt is disabled.
1	The BOD33 Synchronization Ready interrupt is enabled, and an interrupt request will be generated when the BOD33 Synchronization Ready Interrupt flag is set.

Bit 1 – BOD33DET BOD33 Detection Interrupt Enable

Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the BOD33 Detection Interrupt Enable bit, which enables the BOD33 Detection interrupt.

Value	Description
0	The BOD33 Detection interrupt is disabled.
1	The BOD33 Detection interrupt is enabled, and an interrupt request will be generated when the BOD33 Detection Interrupt flag is set.

Bit 0 – BOD33RDY BOD33 Ready Interrupt Enable

Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the BOD33 Ready Interrupt Enable bit, which enables the BOD33 Ready interrupt.

Value	Description
0	The BOD33 Ready interrupt is disabled.
1	The BOD33 Ready interrupt is enabled, and an interrupt request will be generated when the BOD33 Ready Interrupt flag is set.

21.6.3 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x08
Reset: x initially determined from NVM User Row after reset
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
				Reserved	Reserved	VCOREPLLRDY		
Access				R/W	R/W	R/W		
Reset				0	0	0		

Bit	15	14	13	12	11	10	9	8
					ULPVREFRDY	VCORERDY		VREGRDY
Access					R/W	R/W		R/W
Reset					0	0		1

Bit	7	6	5	4	3	2	1	0
			Reserved	Reserved	Reserved	B33SRDY	BOD33DET	BOD33RDY
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	x

Bit 20 – Reserved

Bit 19 – Reserved

Bit 18 – VCOREPLLRDY Voltage Regulator PLL Ready
This flag is cleared by writing a '1' to it.
This flag is set on a zero-to-one transition of the VDDPLL Voltage Ready bit in the Status register (STATUS.VCOREPLLRDY) and will generate an interrupt request if INTENSET.VCOREPLLRDY=1.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the VCOREPLLRDY Voltage Ready interrupt flag.
The Voltage Regulator PLL can be enabled.

Bit 11 – ULPVREFRDY Low Power Voltage Reference Ready Interrupt Enable
Writing a '0' to this bit has no effect.
The ULPVREFRDY bit will clear on a zero-to-one transition of the Low Power Voltage Reference Ready bit in the Status register (STATUS.ULPVREFRDY) and will generate an interrupt request if INTENSET.ULPVREFRDY=1.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the ULPVREFRDY Voltage Ready interrupt flag.

Bit 10 – VCORERDY VDDCORE Voltage Ready
This flag is cleared by writing a '1 to it.
This flag is set on a zero-to-one transition of the VDDCORE Voltage Ready bit in the Status register (STATUS.VCORERDY) and will generate an interrupt request if INTENSET.VCORERDY=1.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the VCORERDY Voltage interrupt flag.

Bit 8 – VREGRDY Voltage Regulator Ready
This flag is cleared by writing a '1' to it.

This flag is set on a zero-to-one transition of the Voltage Regulator Ready bit in the Status register (STATUS.VREGRDY) and will generate an interrupt request if INTENSET.VREGRDY=1.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the VREGRDY interrupt flag.

Bit 5 – Reserved

Bit 4 – Reserved

Bit 3 – Reserved

Bit 2 – B33SRDY BOD33 Synchronization Ready
This flag is cleared by writing a '1' to it.
This flag is set on a zero-to-one transition of the BOD33 Synchronization Ready bit in the Status register (STATUS.B33SRDY) and will generate an interrupt request if INTENSET.B33SRDY=1.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the BOD33 Synchronization Ready interrupt flag.

Bit 1 – BOD33DET BOD33 Detection
This flag is cleared by writing a '1' to it.
This flag is set on a zero-to-one transition of the BOD33 Detection bit in the Status register (STATUS.BOD33DET) and will generate an interrupt request if INTENSET.BOD33DET=1.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the BOD33 Detection interrupt flag.

Bit 0 – BOD33RDY BOD33 Ready
This flag is cleared by writing a '1' to it.
This flag is set on a zero-to-one transition of the BOD33 Ready bit in the Status register (STATUS.BOD33RDY) and will generate an interrupt request if INTENSET.BOD33RDY=1.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the BOD33 Ready interrupt flag.
The BOD33 can be enabled.

21.6.4 Status

Name: STATUS
Offset: 0x0C
Reset: x initially determined from NVM User Row after reset
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
				Reserved	Reserved	VCOREPLLRDY		
Access				R	R	R		
Reset				0	0	0		

Bit	15	14	13	12	11	10	9	8
			Reserved	ULPVREFRDY		VCORERDY		VREGRDY
Access			R	R		R		R
Reset			1	1		1		1

Bit	7	6	5	4	3	2	1	0
			Reserved	Reserved	Reserved	B33SRDY	BOD33DET	BOD33RDY
Access			R	R	R	R	R	R
Reset			1	0	1	1	0	x

Bit 20 – Reserved

Bit 19 – Reserved

Bit 18 – VCOREPLLRDY VDDPLL Voltage Ready

Value	Description
0	The VDDPLL voltage is not as expected.
1	The VDDPLL voltage is the target voltage.

Bit 13 – Reserved

Bit 12 – ULPVREFRDY Low Power Voltage Reference Ready

Value	Description
0	The ULPVREF voltage is not as expected.
1	The ULPVREF voltage is the target voltage.

Bit 10 – VCORERDY VDDCORE Voltage Ready

Value	Description
0	The VDDCORE voltage is not as expected.
1	The VDDCORE voltage is the target voltage.

Bit 8 – VREGRDY Voltage Regulator Ready

Value	Description
0	The selected voltage regulator in VREG.SEL is not ready.
1	The voltage regulator selected in VREG.SEL is ready and the core domain is supplied by this voltage regulator.

Bit 5 – Reserved

Bit 4 – Reserved

Bit 3 – Reserved

Bit 2 – B33SRDY BOD33 Synchronization Ready

Value	Description
0	BOD33 synchronization is ongoing.
1	BOD33 synchronization is complete.

Bit 1 – BOD33DET BOD33 Detection

Value	Description
0	No BOD33 detection.
1	BOD33 has detected that the I/O power supply is going below the BOD33 reference value.

Bit 0 – BOD33RDY BOD33 Ready

The BOD33 can be enabled at start-up from NVM User Row.

Value	Description
0	BOD33 is not ready.
1	BOD33 is ready.

21.6.5 3.3V Brown-Out Detector (BOD33) Control

Name: BOD33
Offset: 0x10
Reset: x initially determined from NVM User Row after reset
Property: PAC Write-Protection, Enable-Protected Bits, Write-synchronized Bits

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
			LEVEL[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			x	x	x	x	x	x

Bit	15	14	13	12	11	10	9	8
	PSEL[3:0]				VREFSEL			ACTCFG
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	STDBYCFG	ACTION[1:0]		HYST	ENABLE	
Access		R/W	R/W	R/W	R/W	R/W	R/W	
Reset		0	0	x	x	x	x	

Bits 21:16 – LEVEL[5:0] BOD33 Threshold Level on VDD/AVDD
These bits set the triggering voltage threshold for the BOD33 when the BOD33 monitors the VDD/AVDD. These bits are loaded from NVM User Row at start-up.
Note: This bit field is enable-protected. This bit field is not synchronized.

Bits 15:12 – PSEL[3:0] Prescaler Select
Selects the prescaler divide-by output for the BOD33 sampling mode. The input clock comes from the OSCULP32K 1024 Hz output.
Note: This bit field is enable-protected. This bit field is not synchronized.

Value	Name	Description
0x0	DIV2	Divide clock by 2
0x1	DIV4	Divide clock by 4
0x2	DIV8	Divide clock by 8
0x3	DIV16	Divide clock by 16
0x4	DIV32	Divide clock by 32
0x5	DIV64	Divide clock by 64
0x6	DIV128	Divide clock by 128
0x7	DIV256	Divide clock by 256
0x8	DIV512	Divide clock by 512
0x9	DIV1024	Divide clock by 1024
0xA	DIV2048	Divide clock by 2048
0xB	DIV4096	Divide clock by 4096
0xC	DIV8192	Divide clock by 8192
0xD	DIV16384	Divide clock by 16384
0xE	DIV32768	Divide clock by 32768
0xF	DIV65536	Divide clock by 65536

Bit 11 – VREFSEL BOD33 Voltage Reference Selection

Note: This bit is enable-protected. This bit is not synchronized.

Value	Description
0	Selects VREF for the BOD33.
1	Selects ULPVREF for the BOD33.

Bit 8 – ACTCFG BOD33 Configuration in Active Sleep Mode

Note: This bit is enable-protected. This bit is not synchronized.

Value	Description
0	In active mode, the BOD33 operates in continuous mode.
1	In active mode, the BOD33 operates in sampling mode.

Bit 6 – RUNSTDBY Run in Standby

Note: This bit is enable-protected. This bit is not synchronized.

Value	Description
0	In standby sleep mode, the BOD33 is disabled.
1	In standby sleep mode, the BOD33 is enabled.

Bit 5 – STDBYCFG BOD33 Configuration in Standby Sleep Mode

If the RUNSTDBY bit is set to '1', the STDBYCFG bit sets the BOD33 configuration in standby sleep mode.

Note: This bit is enable-protected. This bit is not synchronized.

Value	Description
0	In standby sleep mode, the BOD33 is enabled and configured in continuous mode.
1	In standby sleep mode, the BOD33 is enabled and configured in sampling mode.

Bits 4:3 – ACTION[1:0] BOD33 Action

These bits are used to select the BOD33 action when the supply voltage crosses below the BOD33 threshold.
These bits are loaded from NVM User Row at start-up.

Note: This bit field is enable-protected. This bit field is not synchronized.

Value	Name	Description
0x0	NONE	No action
0x1	RESET	The BOD33 generates a reset
0x2	INT	The BOD33 generates an interrupt
0x3	-	Reserved

Bit 2 – HYST Hysteresis

This bit indicates whether hysteresis is enabled for the BOD33 threshold voltage.
This bit is loaded from NVM User Row at start-up.

Note: This bit field is enable-protected. This bit field is not synchronized.

Value	Description
0	No hysteresis.
1	Hysteresis enabled.

Bit 1 – ENABLE Enable

This bit is loaded from NVM User Row at start-up.

Note: This bit is write-synchronized: STATUS.B33SRDY must be checked to ensure the BOD33.ENABLE synchronization is complete.

Note: This bit is not enable-protected.

Value	Description
0	BOD33 is disabled.
1	BOD33 is enabled.

21.6.6 Voltage Regulators System (VREG) Control

Name: VREG
Offset: 0x18
Reset: 0x00000002
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
	VSPER[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
					VSVSTEP[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
							VREFSEL	LPEFF
Access							R/W	R/W
Reset							0	0

Bit	7	6	5	4	3	2	1	0
		RUNSTDBY	STDBYPL0		SEL[1:0]		ENABLE	
Access		R/W	R/W		R/W	R/W	R/W	
Reset		0	0		0	0	1	

Bits 31:24 – VSPER[7:0] Voltage Scaling Period
This bitfield sets the period between the voltage steps when the VDDCORE voltage is changing in µs.
If VSPER=0, the period between two voltage steps is 1µs.
Note: This setting is common to both MAINVREG and VREGPLL voltage regulators.

Bits 19:16 – VSVSTEP[3:0] Voltage Scaling Voltage Step
This field sets the voltage step height when the VDDCORE voltage is changing to reach the target VDDCORE voltage.
The voltage step is equal to $2^{VSVSTEP*}$ min_step.
Voltage Scaling Minimum Voltage Step (min_step) can be found on the Power Supply Electrical Specifications from Electrical Characteristics chapter.
Note: This setting is common to both MAINVREG and VREGPLL voltage regulators.

Bit 9 – VREFSEL Voltage Regulator Voltage Reference Selection
This bit provides support of using ULPVREF during active function mode.
Note: This setting is common to both MAINVREG and VREGPLL voltage regulators.

Value	Description
0	Selects VREF for the voltage regulator.
1	Selects ULPVREF for the voltage regulator.

Bit 8 – LPEFF Low power Mode Efficiency

Value	Description
0	The voltage regulator in Low power mode has the default efficiency and supports the whole VDD/AVDD range (1.62V to 3.63V).
1	The voltage regulator in Low power mode has the highest efficiency and supports a limited VDD/AVDD range (2.5V to 3.63V).

Bit 6 – RUNSTDBY Run in Standby

Value	Description
0	The voltage regulator is in low power mode in Standby sleep mode.
1	The voltage regulator is in normal mode in Standby sleep mode.

Bit 5 – STDBYPL0 Standby in PL0

This bit selects the performance level (PL) of the main voltage regulator for the Standby sleep mode. This bit is only considered when RUNSTDBY=1.

Note: This setting is common to both MAINVREG and VREGPLL voltage regulators

Value	Description
0	In Standby sleep mode, the voltage regulator remains in the current performance level.
1	In Standby sleep mode, the voltage regulator is used in PL0.

Bits 3:2 – SEL[1:0] Voltage Regulator Selection

Value	Description
0	The voltage regulator in active mode is a LDO voltage regulator.
1	The voltage regulator in active mode is a buck converter.
2-3	Reserved.

Bit 1 – ENABLE Must Be Set to 1.

Bit 1 must always be set to ‘1’ when programming the VREG register.

21.6.7 Voltage References System (VREF) Control

Name: VREF
Offset: 0x1C
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
					SEL[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	ONDEMAND	RUNSTDBY			Reserved	VREFOE	Reserved	
Access	R/W	R/W			R/W	R/W	R/W	
Reset	0	0			0	0	0	

Bits 19:16 – SEL[3:0] Voltage Reference Selection

These bits select the Voltage Reference for the ADC/DAC.

Value	Name	Description
0x0 - 0x5	Reserved	Reserved
0x6	2V4	2.4V voltage reference typical value
0x7	2V5	2.5V voltage reference typical value
0x8-0xF	Reserved	Reserved

Bit 7 – ONDEMAND On Demand Control

The On Demand operation mode allows to enable or disable the voltage reference depending on peripheral requests.

Value	Description
0	The voltage reference is always on, if enabled.
1	The voltage reference is enabled when a peripheral is requesting it. The voltage reference is disabled if no peripheral is requesting it.

Bit 6 – RUNSTDBY Run In Standby

The bit controls how the voltage reference behaves during standby sleep mode.

Value	Description
0	The voltage reference is halted during standby sleep mode.
1	The voltage reference is not stopped in standby sleep mode. If VREF.ONDEMAND=1, the voltage reference will be running when a peripheral is requesting it. If VREF.ONDEMAND=0, the voltage reference will always be running in standby sleep mode.

Bit 3 – Reserved

Bit 2 – VREFOE Voltage Reference Output Enable

Value	Description
0	The Voltage Reference output (INTREF) is not available as an ADC input channel.

Value	Description
1	The Voltage Reference output (INTREF) is routed to an ADC input channel.

Bit 1 – Reserved

21.6.8 VREGPLL Voltage Regulator System Control

Name: VREGPLL
Offset: 0x20
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
					STARTUP[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
		RUNSTDBY					ENABLE	
Access		R/W					R/W	
Reset		0					0	

Bits 11:8 – STARTUP[3:0] Startup Time

These bits define the VREGPLL wake-up counts, using the 32.768 kHz Ultra Low Power Oscillator (OSCULP32K) output.
The startup time is calculated as: Startup_Time = (STARTUP + 1) * osculp32k_period.
The STARTUP register value must comply with the requirement: STARTUP ≥ (CEXT_TYP + CEXT_ERR)/ 1µF, where CEXT_TYP represents the typical external capacitor connected to VDDPLL, and CEXT_ERR represents the accuracy error of the external capacitor.

Bit 6 – RUNSTDBY Run In Standby

Value	Description
0	In Standby sleep mode, the VREGPLL voltage regulator is enabled only during a sleep walking task execution and if VREGPLL.ENABLE = 1.
1	In Standby sleep mode, the VREGPLL voltage regulator is always enabled if VREGPLL.ENABLE = 1.

Bit 1 – ENABLE Enable

Value	Description
0	VREGPLL voltage regulator is disabled.
1	VREGPLL voltage regulator is enabled.

21.6.9 Event Control

Name: EVCTRL
Offset: 0x2C
Reset: 0x0000000
Property: Enable-Protected, PAC Write-Protection

As long as BOD33.ENABLE=1, any writes to this register will be discarded, and an APB error will be generated.

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
				Reserved			BOD33DETEO	
Access				R/W			R/W	
Reset				0			0	

Bit 4 – Reserved

Bit 1 – BOD33DETEO BOD33 Detection Event Output Enable

Value	Description
0	BOD33 detection event output is disabled and event will not be generated
1	BOD33 detection event output is enabled and event will be generated

22. Power Manager (PM)

22.1 Overview

The Power Manager (PM) controls the sleep modes and the power domain gating of the device.

Various sleep modes are provided in order to fit power consumption requirements. This enables the PM to stop unused modules in order to save power. In active mode, the CPU is executing application code. When the device enters a sleep mode, program execution is stopped and some modules and clock domains are automatically switched off by the PM according to the sleep mode. The application code decides which sleep mode to enter and when. Interrupts from enabled peripherals and all enabled reset sources can restore the device from a sleep mode to active mode.

Performance level technique consists of adjusting the regulator output voltage to reduce power consumption. The user can select on the fly the performance level configuration which best suits the application.

The power domain gating technique enables the PM to turn off unused power domain supplies individually, while keeping others powered up. Based on activity monitoring, power domain gating is managed automatically by hardware without software intervention. This technique is transparent for the application while minimizing the static consumption. The user can also manually control which power domains will be turned on and off in Standby Sleep mode.

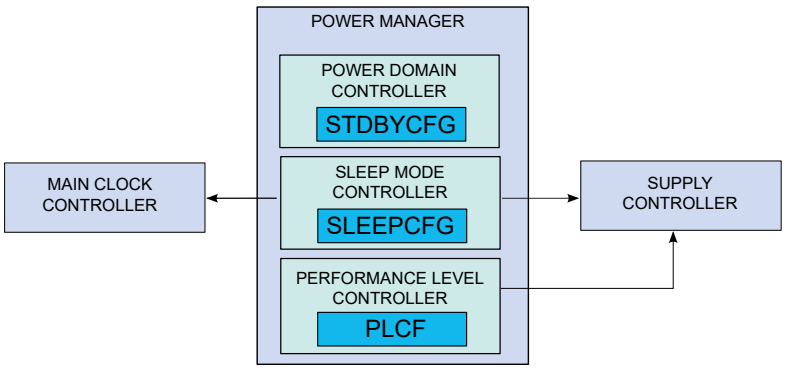
The internal state of the logic is retained (retention state) allowing the application context to be kept in non-active states.

22.2 Features

- Power management control
  - Sleep modes: Idle, Standby and Off
  - Performance levels: PL0 and PL2
  - SleepWalking available in Standby mode.
  - Full retention state in Standby mode
- Power Domain Control
  - Standby Sleep Mode with static power gating
  - SleepWalking extension to power gating
  - SRAM sub-blocks retention

22.3 Block Diagram

Figure 22-1. PM Block Diagram



22.4 Peripheral Dependencies

Table 22-1. PM Configuration Summary

Peripheral name	Base address	NVIC IRQ Index	MCLK AHB/APB Clocks	PAC Peripheral Identifier Index (PAC.WRCTRL)	Power Domain (PM.STDBYCFG)
PM	0x40000400	0: PLRDY	CLK_PM_APB	1	PDAO

22.5 Functional Description

22.5.1 Terminology

The following is a list of terms used to describe the Power Managemement features of this microcontroller.

22.5.1.1 Performance Levels

To help balance between performance and power consumption, the device has two performance levels. Each of the performance levels has a maximum operating frequency and a corresponding maximum consumption in µA/MHz.

It is the application's responsibility to configure the appropriate PL depending on the application activity level. When the application selects a new PL, the voltage applied on the full logic area moves from one value to another. This voltage scaling technique allows to reduce the active power consumption while decreasing the maximum frequency of the device.

22.5.1.1.1 PL0

Performance Level 0 (PL0) provides the maximum energy efficiency configuration.

Maximum CPU and Peripherals operating frequencies in PL0 performance level mode can be found in Maximum Clock Frequencies Electrical Specifications in the Electrical Characteristics chapter.

22.5.1.1.2 PL2

Performance Level 2 (PL2) provides the maximum operating frequency.

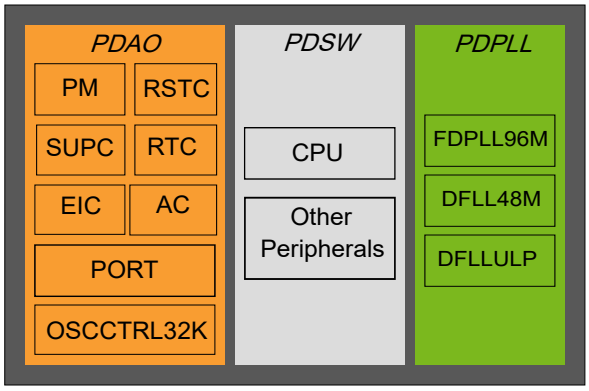
Maximum CPU and Peripherals operating frequencies in PL2 performance level mode can be found in Maximum Clock Frequencies Electrical Specifications in the Electrical Characteristics chapter.

22.5.1.2 Power Domains

The device provides these power domains:

- PDAO (Power Domain Always On)
- PDPLL (Power Domain of FPDLL96M, DFLL48M and DFLLULP)
- PDSW (Power Domain Switchable)

Figure 22-2. PIC32CM LE00/LS00/LS60 Power Domain Partitioning



22.5.1.2.1 PDAO - Power Domain Always On

PDAO contains all peripherals located in the always-on domain. It is always powered in Active, Idle, or Standby modes:

- Power Manager (PM)
- Reset Controller (RSTC)
- 32.768 kHz Oscillators Controller (OSCCTRL32K)
- Supply Controller (SUPC)
- Real Time Controller (RTC)
- External Interrupt Controller (EIC)
- Port Controller (PORT)
- Analog Controller (AC)

22.5.1.2.2 PDSW - Power Domain Switchable

PDSW is a "switchable power domain": in standby sleep mode, it can be turned off to save leakage consumption according to user configuration. PDSW contains all the other peripherals and the Cortex-M23.

22.5.1.2.3 PDPLL - Power Domain of FPDLL96M, DFLL48M and DFLLULP

PDPLL is the FPDLL96M, DFLL48M and DFLLULP clock sources power domain.

In standby sleep mode, it can be turned off to save leakage consumption according to user configuration.

Depending on peripheral settings, the FDPLL96M and DFLL48M can also be switched off if no clock activity is required.

22.5.1.3 Sleep Modes

The device can be set in a Sleep mode. In Sleep mode, the CPU is stopped and the peripherals are either active or idle, according to the Sleep mode depth and their on-demand and run-in-standby clocks settings.

- Idle Sleep mode: The CPU is stopped. Synchronous clocks are stopped except when requested. The logic is retained.
- Standby Sleep mode: The CPU is stopped. Both Synchronous and Asynchronous clocks are stopped except when requested. The logic is retained, and power domain gating can be used to reduce power consumption further.
- Off Sleep mode: The entire device is powered off.

22.5.1.4 Power Domain States and Gating

In Standby sleep mode, the Power Domain Gating technique allows for selecting the state of PDSW power domain automatically (for example, for executing sleepwalking tasks) or manually:

Active State	The power domain is powered according to the performance level
Retention State	The main voltage supply for the power domain is switched off, while maintaining a secondary low-power supply for sequential cells. The logic context is restored when waking up.

22.5.2 Principle of Operation

In Active mode, all clock domains and power domains are active, allowing software execution and peripheral operation. The PM Sleep Mode Controller allows to save power by choosing between different sleep modes depending on application requirements, see 22.5.3.3. Sleep Mode Controller.

The PM Performance Level Controller allows to optimize either for low power consumption or high performance.

The PM Power Domain Controller allows to reduce the power consumption in standby mode even further.

22.5.3 Basic Operation

22.5.3.1 Initialization

After a Power-on Reset (POR), the PM is enabled, the device is in Active mode, the performance level is PL0 (the lowest power consumption) and all the power domains are in active state.

22.5.3.2 Enabling, Disabling and Resetting

The PM is always enabled and can not be reset.

22.5.3.3 Sleep Mode Controller

Sleep mode is entered by executing the Wait For Interrupt instruction (WFI). The Sleep Mode bits in the Sleep Configuration register (SLEEPCFG.SLEEPMODE) select the level of the sleep mode.

Note: A small latency happens between the store instruction and actual writing of the SLEEPCFG register due to bridges. Software must ensure that the SLEEPCFG register reads the desired value before issuing a WFI instruction.

Note: After power-up, the MAINVREG low power mode takes some time to stabilize. Once stabilized, the SUPC->STATUS.ULPVREFRDY bit is set. Before entering Standby, software must ensure that the SUPC->STATUS.ULPVREFRDY bit is set.

Table 22-2. Sleep Mode Entry and Exit Table

Mode	Mode Entry	Wake-Up Sources
IDLE	SLEEPCFG.SLEEPMODE = IDLE	Synchronous (2) (APB, AHB), asynchronous (1)
STANDBY	SLEEPCFG.SLEEPMODE = STANDBY	Synchronous(3), Asynchronous
OFF	SLEEPCFG.SLEEPMODE = OFF	External Reset

Notes:

1. Asynchronous: interrupt generated on generic clock, external clock, or external event.
2. Synchronous: interrupt generated on the APB clock.
3. Synchronous interrupt only for peripherals configured to run in standby.

Note: The type of wake-up sources (synchronous or asynchronous) is given in each module interrupt section.

The sleep modes (idle, standby and off) and their effect on the clocks activity, the regulator and the NVM state are described in the table and the sections below. Refer to Power Domain Controller for the power domain gating effect.

Table 22-3. Sleep Mode Overview

Mode	Main clock	CPU	AHBx and APBx clock	GCLK clocks	Oscillators		VDDCORE Regulator	VDDPLL Regulator	NVM
					ONDEMAND = 0	ONDEMAND = 1			
Active	Run	Run	Run	Run(3)	Run	Run if requested	MAINVREG	ON(4)	active
IDLE	Run	Stop	Stop(1)	Run(3)	Run	Run if requested	MAINVREG	ON(4)	active
STANDBY	Stop(1)	Stop	Stop(1)	Stop(1)	Run if requested or RUNSTDBY=1	Run if requested	MAINVREG in low power mode	ON(5)	Ultra Low- power
OFF	Stop	Stop	Stop	OFF	OFF	OFF	OFF	OFF	OFF

Notes:

1. Running if requested by peripheral during SleepWalking.
2. Running during SleepWalking.
3. Following On-Demand Clock Request principle.
4. Running if enabled (SUPC.VREGPLL.ENABLE = 1)
5. Running if enabled (VREGPLL.ENABLE = 1) and allowed to run in Standby sleep mode: (SUPC.VREGPLL.RUNSTDBY = 1) or (SUPC.VREGPLL.RUNSTDBY = 0) and following the sleep walking request. SUPCVREG.RUNSTDBY must be set to allow VREGPLL running in Standby sleep mode. Refer to OSCCTRL module for details on oscillators behavior in STANDBY sleep mode.

22.5.3.3.1 IDLE Mode

IDLE mode allows power optimization with the fastest wake-up time.

The CPU is stopped, and peripherals are still working. As in Active mode, the AHBx and APBx clocks for peripheral are still provided if requested. As the main clock source is still running, wake-up time is very fast.

- Entering Idle mode: The Idle mode is entered by executing the WFI instruction. Additionally, if the SLEEPONEXIT bit in the Cortex System Control register (SCR) is set, the Idle mode will be entered when the CPU exits the lowest priority ISR (Interrupt Service Routine, refer to the ARM Cortex documentation for details). This mechanism can be useful for applications that only require the processor to run when an interrupt

occurs. Before entering the Idle mode, the user must select the Idle Sleep mode in the Sleep Configuration register (SLEEPCFG.SLEEPMODE=IDLE).

- Exiting Idle mode: The processor wakes the system up when it detects any non-masked interrupt with sufficient priority to cause exception entry. The system goes back to the Active mode. The CPU and affected modules are restarted.

GCLK clocks, regulators and SRAM are not affected by the Idle Sleep mode and operate in normal mode.

22.5.3.3.2 STANDBY Mode

The Standby mode is the lowest power configuration while keeping the state of the logic and the content of the SRAM.

In this mode, all clocks are stopped except those configured to be running sleepwalking tasks. The clocks can also be active on request or at all times, depending on their on-demand and run-in-standby settings. Either synchronous (CLK_APBx or CLK_AHBx) or generic (GCLK_x) clocks or both can be involved in sleepwalking tasks. This is the case when for example the SERCOM RUNSTDBY bit is written to '1'.

- Entering Standby mode: This mode is entered by executing the WFI instruction after writing the Sleep Mode bit in the Sleep Configuration register (SLEEPCFG.SLEEPMODE=STANDBY). The SLEEPONEXIT feature is also available as in Idle mode.
- Exiting Standby mode: Any peripheral able to generate an asynchronous interrupt can wake up the system. For example, a peripheral running on a GCLK clock can trigger an interrupt. When the enabled asynchronous wake-up event occurs and the system is woken up, the device will either execute the interrupt service routine or continue the normal program execution according to the Priority Mask Register (PRIMASK) configuration of the CPU.

Refer to 22.5.3.5. Power Domain Controller for the SRAM state.

The main regulator operates in Low-Power mode by default and switches automatically to the normal mode in case of a sleepwalking task requiring more power. It returns automatically to low power mode when the sleepwalking task is completed.

The VREGPLL regulator is enabled in the following cases:

- When (VREGPLL.RUNSTDBY = 1)
- When (VREGPLL.RUNSTDBY = 0) and a sleep walking task must be executed

22.5.3.3.3 OFF Mode

In Off mode, the device is entirely powered-off.

- Entering Off mode: This mode is entered by selecting the Off mode in the Sleep Configuration register by writing the Sleep Mode bits (SLEEPCFG.SLEEPMODE=OFF), and subsequent execution of the WFI instruction.
- Exiting Off mode: This mode is left by pulling the $\overline{\text{RESET}}$ pin low, or when a power Reset is done.

22.5.3.4 Performance Level

The application can change the performance level on the fly writing to the by Performance Level Select bit in the Performance Level Configuration register (PLCFG.PLSEL).

When changing to a lower performance level, the bus frequency must be reduced before writing PLCFG.PLSEL in order to avoid exceeding the limit of the target performance level.

When changing to a higher performance level, the bus frequency can be increased only after the Performance Level Ready flag in the Interrupt Flag Status and Clear (INTFLAG.PLRDY) bit set to '1', indicating that the performance level transition is complete.

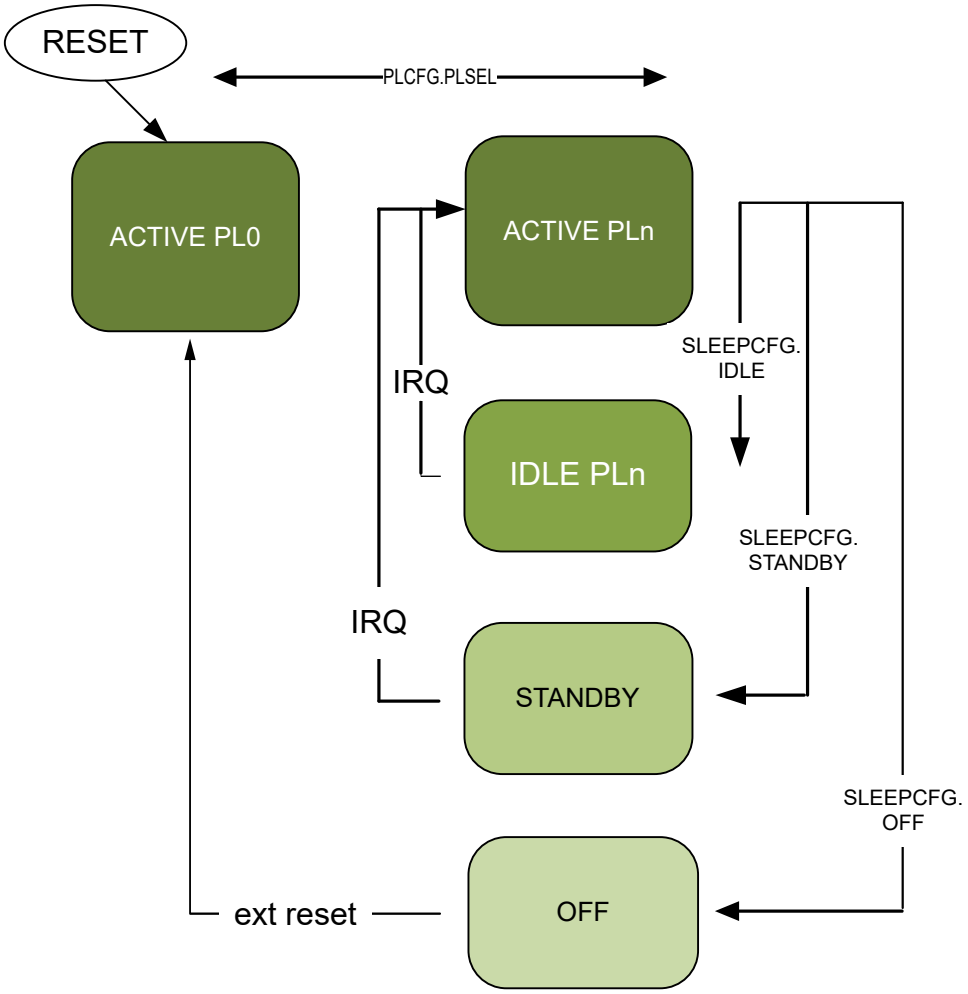
After a reset, the device starts in the lowest PL (lowest power consumption and lowest max frequency). The application can then switch to another PL at anytime without any stop in the code execution. As shown in Figure 22-3, performance level transition is possible only when the device is in active mode.

The Performance Level Disable bit in the Performance Level Configuration register (PLCFG.PLDIS) can be used to freeze the performance level to PL0. This disables the performance level hardware mechanism in order to reduce both the power consumption and the wake-up startup time from standby sleep mode.

Note: This bit PLCFG.PLDIS must be changed only when the current performance level is PL0.

Any attempt to modify this bit while the performance level is not PL0 is discarded and a violation is reported to the PAC module. Any attempt to change the performance level to PLn (with n>0) while PLCFG.PLDIS=1 is discarded and a violation is reported to the PAC module.

Figure 22-3. Sleep Modes and Performance Level Transitions



22.5.3.5 Power Domain Controller

The Power Domain Controller provides several ways of how power domains are handled while the device is in Standby mode or entering Standby mode:

- Default operation - all peripherals idle
  When entering Standby mode, the power domain PDSW is set in retention state. This allows for very low power consumption while retaining all the logic content of these power domains. When exiting Standby mode, all power domains are set back to active state.
- Default operation - Standby Sleep Mode with static power gating
  Static Power Domain Gating is a technique that allows to automatically turn off the PDSW power domain supply when not used while keeping PDAO powered up.
- SleepWalking extension to power gating (SleepWalking with dynamic power gating)
  SleepWalking is the capability for a device in Standby Sleep mode, to temporarily wake-up clocks for a peripheral to perform a task without waking-up the CPU. The SleepWalking feature has been expanded to control power gating in addition to clock gating. The power domain PDSW can be automatically controlled (active or retention state) depending on peripheral requirements (PDCFG bit from the STDBYCFG register).

The static and dynamic power gating features are fully transparent for the user.

Table 22-4. Sleep Modes versus Power Domain States Overview

	Power Domain State		
Sleep Mode	**PDSW**	**PDAO**	**PDPLL**
Active	active	active	active(2)
Idle	active	active	active(2)
Standby - At least one peripheral from PDSW with RUNSTDBY = 1 OR PDCFG = 1	active (1)	active	active(2)
Standby - No peripheral from PDSW with RUNSTDBY = 1	retention	active	off
Off	off	off	off

Notes:

1. PDSW can be switched automatically in retention mode if the dynamic power gating feature is enabled.
2. If enabled (SUPC.VREGPLL.ENABLE = 1)

22.5.3.6 Regulators, SRAM, and NVM State in Sleep Mode

By default, in Standby Sleep mode, the SRAM, NVM, and regulators are automatically set in Low-Power mode to reduce power consumption:

- The SRAM is in Low-Power mode if its power domain is in retention or off state.
- Non-Volatile Memory - the NVM is located in the power domain PDSW. By default, the NVM is automatically set in low power mode in these conditions:
  - When the power domain PDSW is in retention or off state.
  - When the device is in Standby Sleep mode and the NVM is not accessed. This behavior can be changed by software by configuring the SLEEPPRM bit group of the CTRLB register in the NVMCTRL peripheral.
  - When the device is in Idle Sleep mode and the NVM is not accessed. This behavior can be changed by software by configuring the SLEEPPRM bit group of the CTRLB register in the NVMCTRL peripheral.
- Regulators: by default, in Standby Sleep mode, the PM analyzes the device activity to use either the main or the low-power voltage regulator to supply the VDDCORE. The VREGPLL regulator is enabled only if one of the FDPLL96M, DFLL48M or DFLLULP clocks are requested by a peripheral.

GCLK clocks, regulators and SRAM are not affected in Idle Sleep mode and will operate as normal.

Table 22-5. Regulators, SRAM, and NVM state in Sleep Mode

Sleep Mode	PDSW	SRAM Mode(1)	NVM	Regulators		
				VDDCORE		VDDPLL
				MAINVREG	LPVREG	
Active	active	normal	normal	on	on	on(7)
Idle	active	auto(2)	on	on	on	on(7)
Standby - PDSW in Active mode	active	normal(6)	auto(2)	auto(3)	on(5)	auto(8)/on
Standby - PDSW in Retention mode	retention	low power(6)	low power	auto(4)	on(5)	off
OFF	off	off	off	off	off	off

Notes:

1. SRAM mode by default: STDBYCFG.BBIAS bits are set to their default value.
2. auto: by default, NVM is in low-power mode if not accessed.
3. auto: by default, the main voltage regulator is on if GCLK, APBx, or AHBx clock is running during SleepWalking.
4. auto: by default ULP regulator is selected in retention, but main regulator will be selected if VREG RUNSTDBY register bit in Supply Controller is set to 1.
5. on: low power voltage reference must be ready, and this is confirmed if STATUS.ULPVREFRDY register bit in SUPC equals to 1.
6. SRAM can be partially retained in STANDBY using SRAM Power Switch.
7. If enabled (SUPC VREGPLL.ENABLE = 1).
8. Auto: if SUPC VREGPLL.RUNSTDBY = 0.

22.5.4 Advanced Features

22.5.4.1 Power Domain Configuration

When entering Standby Sleep mode, a power domain is set automatically to retention state if no activity is required in it, refer to 22.5.3.5. Power Domain Controller for details. This behavior can be changed by writing the Power Domain Configuration bit group in the Standby Configuration register (STDBYCFG.PDCFG). For example, all power domains can be forced to remain in active state during Standby Sleep mode, this will accelerate wake-up time.

22.5.4.2 SRAM Automatic Low Power Mode

The SRAM is by default put in Low-Power mode (back-biased) if its power domain is in retention state and the device is in Standby Sleep mode.

This behavior can be changed by configuring BBIASxx bit groups in the Standby Configuration register (STDBYCFG.BBIASxx), refer to the table below for details.

Note: in Standby Sleep mode, the DMAC can access the SRAM in Standby Sleep mode only when the power domain PDSW is not in retention and PM.STDBYCFG.BBIASxx=0x0.

Table 22-6. SRAM Back-Biasing Mode

STBYCDFG.BBIASxx config		SRAM
0x0	Retention Back Biasing mode	SRAM is back-biased if its power domain is in retention state
0x1	Standby Back Biasing mode	SRAM is back-biased if the device is in Standby Sleep mode

22.5.4.3 SRAM Power Switch Configuration

The SRAM is divided in sub-blocks which can be switched OFF to optimize power consumption.

By default, all sub-blocks are switched ON but it is possible to switch them OFF (excluding the first 16 kB sub-block which remains always powered) depending on the SRAM memory size use.

This behavior can be changed by configuring RAMPSWC bit field in the Power Configuration register (PWCFG).

⚠ CAUTION This configuration takes effect immediately. So, this is the responsibility of the user to ensure that no access is performed on a SRAM sub-block which is switched OFF.
When a SRAM sub-block is switched from OFF to ON state, 1us delay is required before re-accessing it.

The first sub-block to be switched OFF is always at the top of the SRAM block memory and the same behavior applies for the next switched OFF sub-blocks.

22.5.4.4 Regulator Automatic Low Power Mode

In standby mode, the PM selects either the main or the low power voltage regulator to supply the VDDCORE. If switchable power domain is in retention state, the low power voltage regulator is used.

If a sleepwalking task is working on either asynchronous clocks (generic clocks) or synchronous clock (APB/AHB clocks), the main voltage regulator is used. This behavior can be changed by writing the Voltage Regulator Standby Mode bits in the Standby Configuration register (STDBYCFG.VREGSMOD). Refer to the following table for details.

Table 22-7. Regulator State in Sleep Mode

Sleep Modes	STDBYCFG. VREGSMOD	SleepWalking(1)	Regulator state for VDDCORE
Active	-	-	main voltage regulator
Idle	-	-	main voltage regulator
Standby (active)	0x0: AUTO	NO	low power regulator
		YES	main voltage regulator
	0x1: PERFORMANCE	-	main voltage regulator
	0x2: LP(2)	-	low power regulator
Standby (retention)	-	-	low power regulator

Notes:

1. SleepWalking is running on GCLK clock or synchronous clock. This is not related to XOSC32K or OSCULP32K clocks.
2. Must only be used when SleepWalking is running on GCLK with 32.768 kHz source.

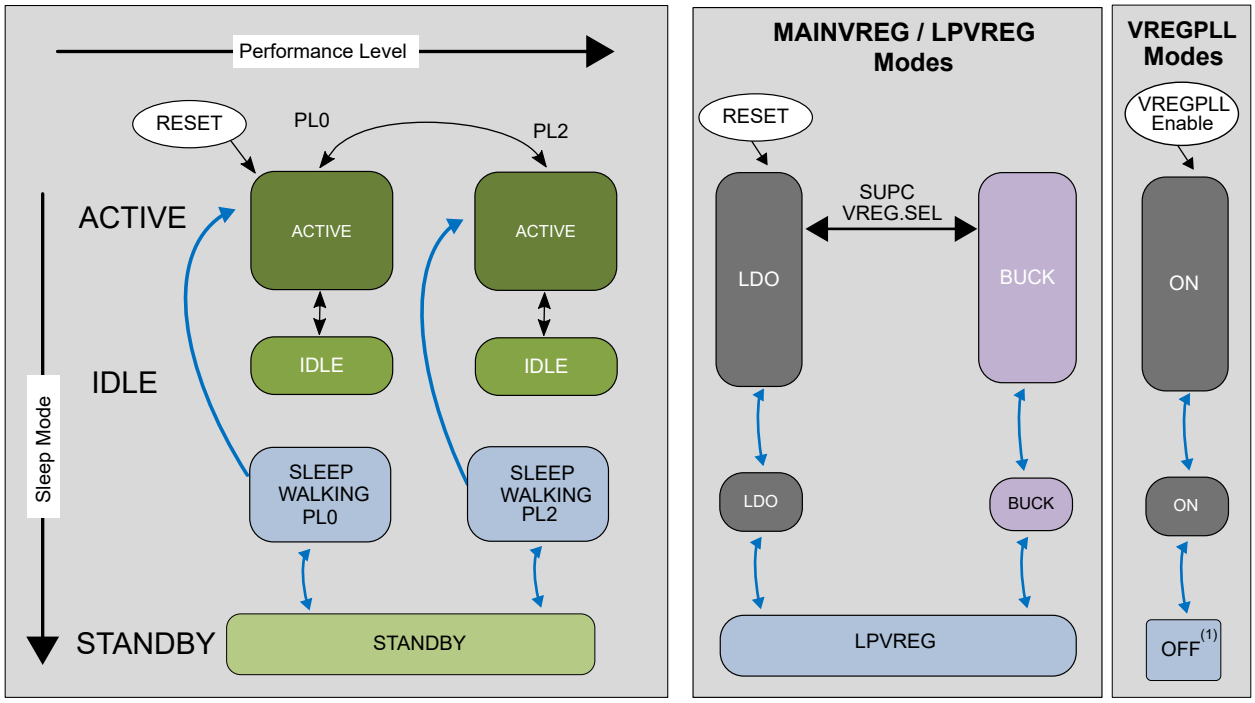
22.5.4.5 SleepWalking and Performance Level

SleepWalking is the capability for a device to temporarily wake up clocks for a peripheral to perform a task without waking up the CPU from STANDBY sleep mode. At the end of the sleepwalking task, the device can either be woken up by an interrupt (from a peripheral involved in SleepWalking) or enter again into STANDBY sleep mode. In this device, SleepWalking is supported only on GCLK clocks by using the on-demand clock principle of the clock sources.

In standby mode, when SleepWalking is ongoing, the performance level used to execute the sleepwalking task is the current configured performance level (used in active mode), and the main voltage regulator used to execute the SleepWalking task is the selected regulator used in active mode (LDO or Buck converter).

These are illustrated in the figure below.

Figure 22-4. Operating Conditions and SleepWalking



Note:

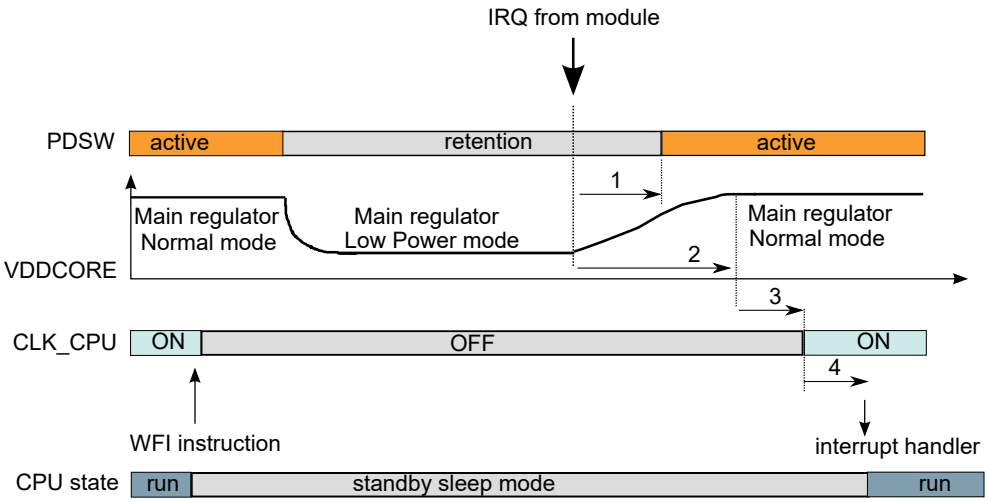
1. If SUPC VREGPLL.RUNSTDBY = 0

22.5.4.6 Wake-Up Time

The total wake-up time depends on the following:

- Latency due to Power Domain Gating:
  Usually, wake-up time is measured with the assumption that the power domain is already in active state. When using Power Domain Gating, changing a power domain from retention to active state will take a certain time (refer to Wake-Up Timing Electrical Specifications from Electrical Characteristics chapter). If power domain was already in active state in standby sleep mode, this latency is zero. If wake-up time is critical for the application, power domain can be forced to active state in Standby Sleep mode, refer to 22.5.4.1. Power Domain Configuration for details.
- Latency due to Performance Level and Regulator effect:
  Performance Level has to be taken into account for the global wake-up time. As example, if PL2 is selected and the device is in Standby Sleep mode, the voltage level supplied by the ULP voltage regulator is lower than the one used in Active mode. When the device wakes up, it takes a certain amount of time for the main regulator to transition to the voltage level corresponding to PL2, causing additional wake-up time.
- Latency due to the CPU clock source wake-up time.
- Latency due to the NVM memory access.
- Latency due to Switchable Power Domain back-bias wake-up time:
  If back-bias is enabled, and the device wakes up from retention, it takes a certain amount of time for the regulator to settle.

Figure 22-5. Total Wake-up Time from Standby Sleep Mode



1: latency due to power domain gating
2: latency due to regulator wakeup time
3: latency due to VREGPLL and clock source wakeup time
4: latency due to flash memory code access

22.5.5 Standby with Static Power Domain Gating in Details

In Standby Sleep mode, the switchable power domain (PDSW) of a peripheral can remain in active state to perform the peripheral's tasks. This Static Power Domain Gating feature is supported by all peripherals. For some peripherals it must be enabled by writing a Run in Standby bit in the respective Control A register (CTRLA.RUNSTDBY) to '1'. Refer to each peripheral chapter for details.

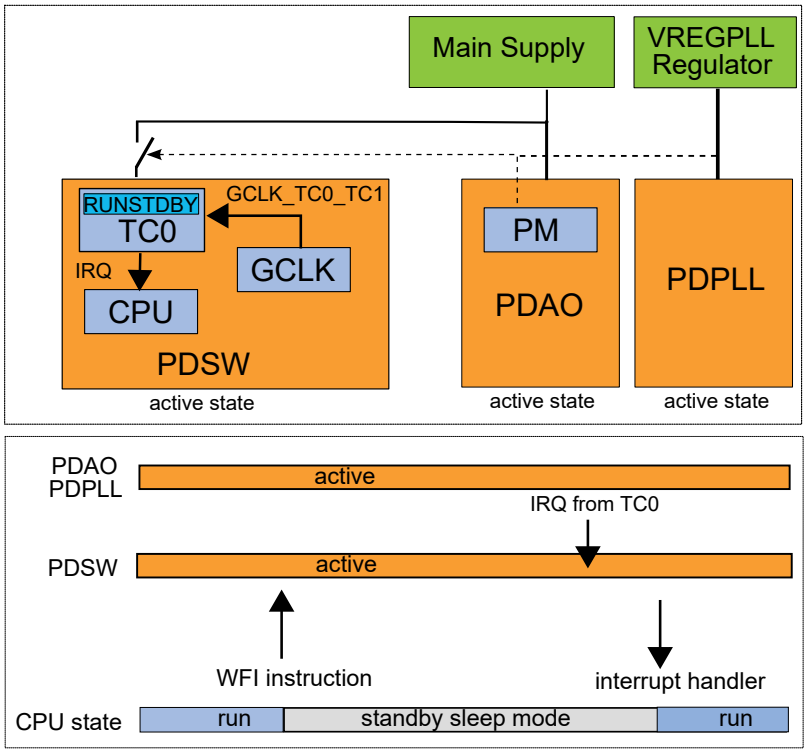
The following examples illustrate Standby with static Power Domain Gating:

TC0 Standby with Static Power Domain Gating

TC0 peripheral is used in counter operation mode. An interrupt is generated to wake-up the device based on the TC0 peripheral configuration. To make the TC0 peripheral continue to run in Standby Sleep mode, the RUNSTDBY bit is written to '1'.

- Entering Standby mode: As shown in Figure 22-6, PDSW remains active. Refer to 22.5.3.5. Power Domain Controller for details.
- Exiting Standby mode: When conditions are met, the TC0 peripheral generates an interrupt to wake-up the device, and the CPU is able to operate normally and execute the TC0 interrupt handler accordingly.
- Wake-up time:
  - The required time to set PDSW to active state has to be considered for the global wake-up time, refer to 22.5.4.6. Wake-Up Time for details.
  - In this case, the VDDCORE voltage is still supplied by the main voltage regulator, refer to 22.5.4.4. Regulator Automatic Low Power Mode for details. Thus, global wake-up time is not affected by the regulator.
  - In case TC0 is running with a clock provided by the FDPLL96M, DFLL48M or DFLLULP, and the SUPC -> VREGPLL.RUNSTDBY = 1, the PDPLL is kept on, and the global wake-up time is not affected by the VREGPLL regulator. If SUPC -> VREGPLL.RUNSTDBY = 0, the time required to set the PDPLL to active state has to be considered for the global wake-up time.

Figure 22-6. TC0 in Standby with Static Power Domain Gating

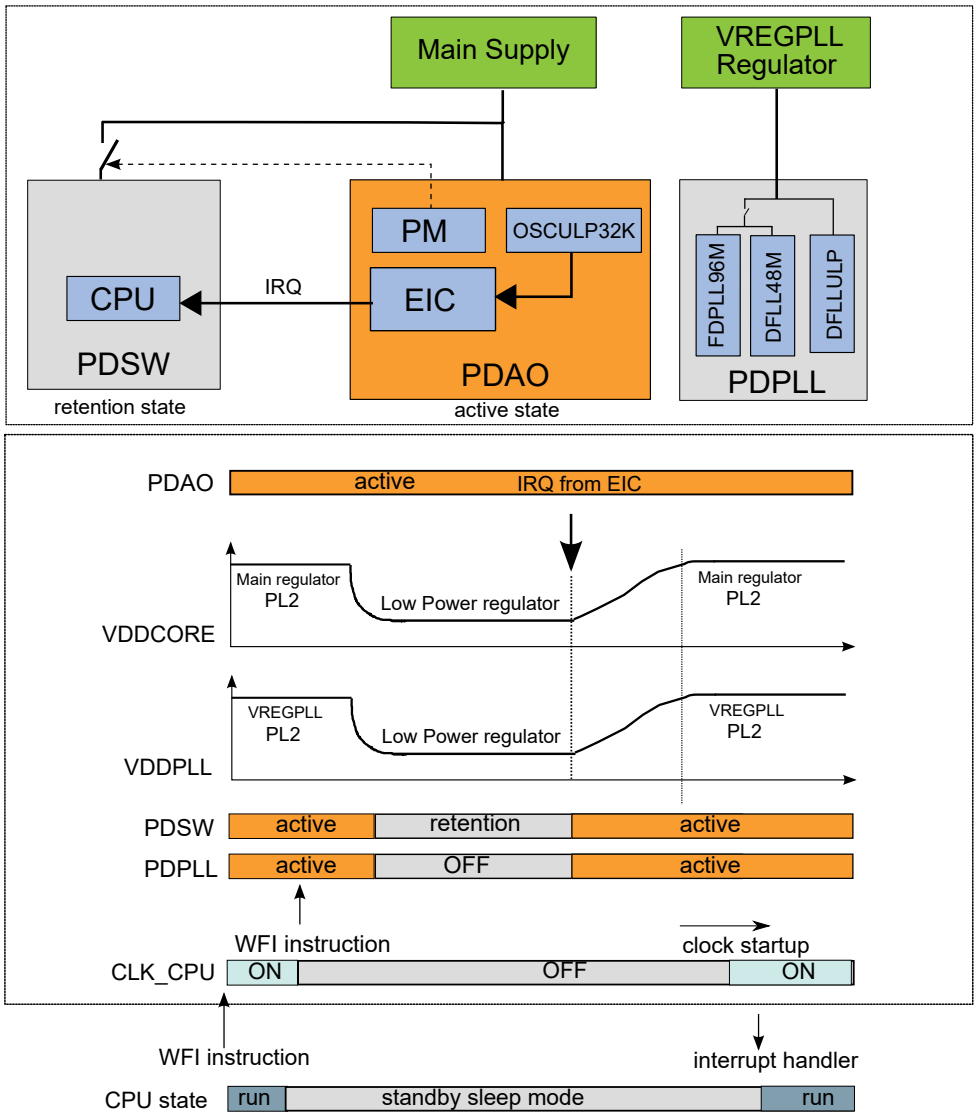


EIC in Standby with Static Power Domain Gating

In this example, EIC peripheral is used to detect an edge condition to generate interrupt to the CPU. An External interrupt pin is filtered by the CLK_ULP32K clock, GCLK peripheral is not used. Refer to Chapter 28. External Interrupt Controller (EIC) for details. The EIC peripheral is located in the power domain PDAO (which is not switchable), and there is no RUNSTDBY bit in the EIC peripheral.

- Entering Standby mode: As shown in Figure 22-7, the switchable power domain is set in retention state by the Power Manager peripheral. The low power regulator supplies the VDDCORE voltage level. The PDPLL is shut-off if no clock requests are present.
- Exiting Standby mode: When conditions are met, the EIC peripheral generates an interrupt to wake the device up. Successively, the PM peripheral sets PDSW to active state, and the main voltage regulator restarts. In the same way, if the CPU is clocked by any clock source from the PDPLL power domain, the VREGPLL regulator restarts. Once PDSW and PDPLL are in active state and both the main voltage and VREGPLL regulators are ready, the CPU is able to operate normally and execute the EIC interrupt handler accordingly.
- Wake-up time:
  - The required time to set the switchable power domains to active state has to be considered for the global wake-up time, refer to 22.5.4.6. Wake-Up Time for details.
  - When in standby Sleep mode, the GCLK peripheral is not used, allowing the VDDCORE to be supplied by the low power regulator to reduce consumption, see 22.5.4.4. Regulator Automatic Low Power Mode. In the same way, if no peripheral is requesting one of the FDPLL96M, DFLL48M or DFLLULP clock sources, the VREGPLL regulator is not required, and it will be internally disabled to save power if SUPC -> VREGPLL.RUNSTDBY = 0. Consequently, main voltage regulator wake-up time has to be considered for the global wake-up time as shown in Figure 22-7.

Figure 22-7. EIC in Standby with Static Power Domain Gating



22.5.6 Sleepwalking with Dynamic Power Domain Gating in Details

To reduce power consumption even further, Sleepwalking with dynamic Power Domain Gating (also referred to as "Dynamic Sleepwalking") is used to turn power domain state from retention to active and vice-versa, based on event or DMA trigger.

22.5.6.1 Dynamic SleepWalking based on Event

To enable SleepWalking with dynamic power domain gating, the Dynamic Power Gating for Power Domain SW bit in the Standby Configuration register (STDBYCFG.DPGPDSW) has to be written to '1'.

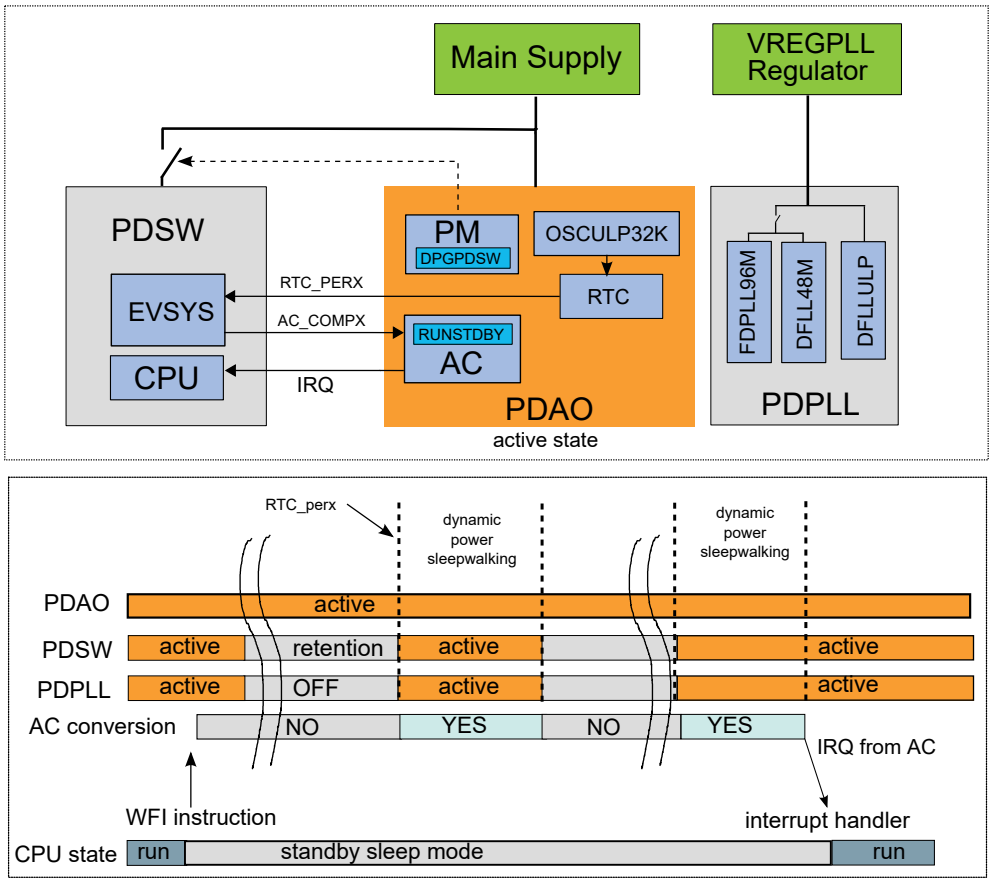
When in retention state, a power domain can be automatically set to active state by the PM if an event is directed to this power domain. In this device, this concerns the event users located in power domain PDSW.

- When PDSW is in retention state, dynamic SleepWalking can be triggered by:
  - AC output event
  - RTC output event
  - EIC output event (if using the CLK_ULP32K clock and debouncing is enabled)

Refer also to 22.5.1.2. Power Domains.

Dynamic SleepWalking based on event is illustrated in the following example:

Figure 22-8. Dynamic SleepWalking based on Event: AC Periodic Comparison



The Analog Comparator (AC) peripheral is used in single shot mode to monitor voltage levels on input pins. A comparator interrupt, based on the AC peripheral configuration, is generated to wake up the device. In the GCLK module, the AC generic clock (GCLK_AC) source is routed a 32.768kHz oscillator (for low power applications, OSC32KULP is recommended). RTC and EVSYS modules are configured to generate periodic events to the AC. To make the comparator continue to run in standby sleep mode, the RUNSTDBY bit is written to '1'. To enable the dynamic SleepWalking for PDSW power domain, STDBYCFG.PDSW must be written to '1'.

Entering standby mode: The Power Manager sets the PDSW power domain in retention state. The AC comparators, COMPx, are OFF. The GCLK_AC clock is stopped. The VDDCORE is supplied by the low power regulator.

Dynamic SleepWalking: The RTC event (RTC_PERX) is routed by the Event System to the Analog Comparator to trigger a single-shot measurement. This event is detected by the Power Manager, which sets the PDSW power domain to active state and starts the main voltage regulator.

After enabling the AC comparator and starting the GCLK_AC, the single-shot measurement can be performed during Sleep mode (sleepwalking task), refer to 45.6.13.2. Single-Shot Measurement during Sleep for details. At the end of the conversion, if conditions to generate an interrupt are not met, the GCLK_AC clock is stopped again, as well as the AC comparator.

The low-power regulator starts again and the PDSW power domain is set back to retention state by the PM. During this dynamic SleepWalking period, the CPU is still sleeping.

Exiting standby mode: during the dynamic SleepWalking sequence, if conditions are met, the AC module generates an interrupt to wake up the device.
Note: if the AC GCLK clock or the CPU are set to use one of the FDPLL96M, DFLL48M or DFLLULP clock sources, the VREGPLL will be enabled and the sleep walking task has to be executed. To avoid longer wake-up times, the regulator RUNSTDBY bit must be set to '1' (SUPC -> VREGPLL.RUNSTDBY.1).

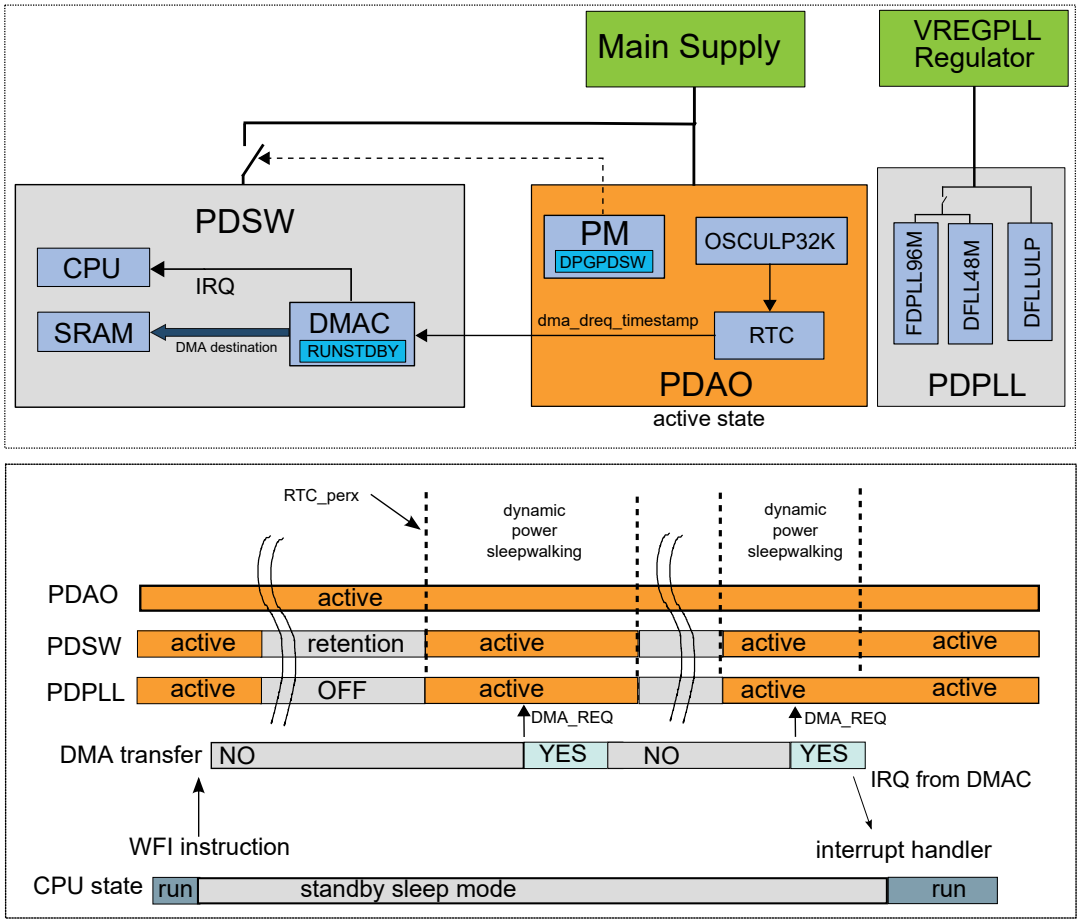
22.5.6.2 Dynamic SleepWalking Based on Peripheral DMA Trigger

To enable this advanced feature, the Dynamic Power Gating for Power Domain SW bit in the Standby Configuration register (STDBYCFG.DPGPDSW) have to be written to '1'.

When in retention state, the power domain PDSW (containing the DMAC) can be automatically set to active state if the PM detects a valid DMA trigger that is coming from a peripheral located in PDAO. A peripheral DMA trigger is valid if the corresponding DMA channel is enabled and its Run in Standby bit (RUNSTDBY) is written to '1'.

This is illustrated in the following example:

Figure 22-9. Dynamic Sleepwalking based on Peripheral DMA Trigger



The DMAC is configured to operate in standby sleep mode by using its respective RUNSTDBY bit. A DMAC channel is configured to set the DMA destination. The Run in Standby bit of this DMAC channel is written to '1' to allow it running in Standby Sleep mode.

Entering Standby mode: The Power Manager peripheral sets PDSW to retention state. The VDDCORE is supplied by the low-power regulator.

Dynamic SleepWalking: based on RTC conditions, an RTC output signal (DMA request for timestamp) triggers DMAC to put timestamp value at configured DMA destination.

This event is detected by the Power Manager which sets the PDSW power domain to active state and starts the main voltage regulator.

This DMA transfer request is detected by the PM, which sets PDSW (containing the DMAC) to active state. The DMAC requests the CLK_DMAC_AHB clock and transfer the timestamp value to the memory. When the DMA beat transfer is completed, the CLK_DMAC_AHB clock is stopped again.

The low-power regulator starts again and the PDSW power domain is set back to retention state by the PM. Note that during this dynamic SleepWalking period, the CPU is still sleeping.

Exiting Standby mode: during SleepWalking with Dynamic Power Gating sequence, if conditions are met, the DMAC generates an interrupt to wake up the device.
Note: if the AHB clock is set to use one of the FDPLL, DFLL or DFLLULP clock sources, the VREGPLL will be enabled and the sleep walking task has to be executed. To avoid longer wake-up times, the regulator RUNSTDBY bit must be set to '1' (SUPC -> VREGPLL.RUNSTDBY=1).

22.5.7 Interrupts

The peripheral has the following interrupt sources:

- Performance Level Ready (PLRDY)
  This interrupt is a synchronous wake-up source. See Table 22-2 for details.

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 peripheral is reset.

An interrupt flag is cleared by writing a '1' to the corresponding bit in the INTFLAG register. Each peripheral can have one interrupt request line per interrupt source or one common interrupt request line for all the interrupt sources. Refer to the Nested Vector Interrupt Controller (NVIC) for details. If the peripheral 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.

22.5.8 Sleep Mode Operation

The Power Manager is always active.

22.5.9 Debug Operation

When the CPU is halted in debug mode, the PM continues normal operation. If standby sleep mode is requested by the system while in debug mode, the power domains are not turned off. As a consequence, power measurements while in debug mode are not relevant.

If OFF sleep mode is requested by the system while in debug mode, the core domains are kept on, and the debug modules are kept running to allow the debugger to access internal registers. When exiting the OFF mode upon a reset condition, the core domains are reset except the debug logic, allowing users to keep using their current debug session.

Hot plugging in standby mode is supported except if the power domain PDSW is in retention state.

Cold plugging in OFF mode is supported as long as the reset duration is superior to (Tmin).

22.6 Register Summary

Refer to the Registers Description section for more details on register properties and access permissions.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	Reserved									
0x01	SLEEPCFG	7:0						SLEEPMODE[2:0]		
0x02	PLCFG	7:0	PLDIS						PLSEL[1:0]	
0x03	PWCFG	7:0							RAMPSWC[1:0]	
0x04	INTENCLR	7:0								PLRDY
0x05	INTENSET	7:0								PLRDY
0x06	INTFLAG	7:0								PLRDY
0x07	Reserved									
0x08	STDBYCFG	15:8				BBIASTR		BBIASHS		
		7:0	VREGSMOD[1:0]			DPGPDSW				PDCFG

22.6.1 Sleep Configuration

Name: SLEEPCFG
Offset: 0x01
Reset: 0x2
Property: PAC Write-Protection

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

Bits 2:0 – SLEEPMODE[2:0] Sleep Mode
Note: A small latency happens between the store instruction and actual writing of the SLEEPCFG register due to bridges. Software has to make sure the SLEEPCFG register reads the wanted value before issuing WFI instruction.

Value	Name	Description
0x0-0x1	Reserved	-
0x2	IDLE	CPU, AHBx, and APBx clocks are OFF
0x3	Reserved	-
0x4	STANDBY	ALL clocks are OFF, unless requested by sleepwalking peripheral
0x5	Reserved	-
0x6	OFF	All power domains are powered OFF
0x7	Reserved	-

22.6.2 Performance Level Configuration

Name: PLCFG
Offset: 0x02
Reset: 0x00
Property: PAC Write-Protection

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

Bit 7 – PLDIS Performance Level Disable

Disabling the PL selection forces the device to run in PL0, reducing the power consumption, and the wake-up time from Standby Sleep mode.
Changing this bit when the current performance level is not PL0 is discarded, and a violation is reported to the PAC module.

Value	Description
0	The Performance Level mechanism is enabled.
1	The Performance Level mechanism is disabled.

Bits 1:0 – PLSEL[1:0] Performance Level Select

Value	Name	Description
0x0	PL0	Performance Level 0
0x1	Reserved	-
0x2	PL2	Performance Level 2
0x3	Reserved	-

22.6.3 Power Configuration

Name: PWCFG
Offset: 0x03
Reset: 0x00
Property: PAC Write-Protection

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

Bits 1:0 – RAMPSWC[1:0] SRAM Power Switch Configuration

⚠ CAUTION This configuration takes effect immediately. So, this is the responsibility of the user to ensure that no access is performed on a SRAM sub-block which is switched OFF.
When a SRAM sub-block is switched from OFF to ON state, 1µs delay is required before re-accessing it.

Table 22-8. 512-KB Flash

Value	Name	Definition
0x0	64KB	64KB Available
0x1	48KB	48KB Available
0x2	32KB	32KB Available
0x3	16KB	16KB Available

Table 22-9. 256-KB Flash

Value	Name	Definition
0x0	32KB	32KB Available
0x1	32KB	32KB Available
0x2	32KB	32KB Available
0x3	16KB	16KB Available

22.6.4 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 (INTENSET) register.

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

Bit 0 – PLRDY Performance Level Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Performance Ready Interrupt Enable bit and the corresponding interrupt request.

Value	Description
0	The Performance Ready interrupt is disabled.
1	The Performance Ready interrupt is enabled and will generate an interrupt request when the Performance Ready Interrupt Flag is set.

22.6.5 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 (INTENCLR) register.

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

Bit 0 – PLRDY Performance Level Ready Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Performance Ready Interrupt Enable bit and enable the Performance Ready interrupt.

Value	Description
0	The Performance Ready interrupt is disabled.
1	The Performance Ready interrupt is enabled.

22.6.6 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x06
Reset: 0x00
Property: –

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

Bit 0 – PLRDY Performance Level Ready
This flag is set when the performance level is ready and will generate an interrupt if INTENSET.PLRDY is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Performance Ready interrupt flag.

22.6.7 Standby Configuration

Name: STDBYCFG
Offset: 0x08
Reset: 0x0000
Property: PAC Write-Protection

Bit	15	14	13	12	11	10	9	8
				BBIASTR		BBIASHS		
Access				R/W		R/W		
Reset				0		0		

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

Bit 12 – BBIASTR Back Bias for TrustRAM
Refer to 22.5.4.2. SRAM Automatic Low Power Mode for details.

Value	Description
0	Retention Back Biasing mode
1	Standby Back Biasing mode

Bit 10 – BBIASHS Back Bias for SRAM
Refer to 22.5.4.2. SRAM Automatic Low Power Mode for details.

Value	Description
0	Retention Back Biasing mode
1	Standby Back Biasing mode

Bits 7:6 – VREGSMOD[1:0] VREG Switching Mode
Refer to 22.5.4.4. Regulator Automatic Low Power Mode for details.

Value	Name	Description
0x0	AUTO	Automatic Mode
0x1	PERFORMANCE	Performance oriented
0x2	LP	Low Power consumption oriented

Bit 4 – DPGPDSW Dynamic Power Gating for Switchable Power Domain

Value	Description
0	Dynamic SleepWalking for switchable power domain is disabled
1	Dynamic SleepWalking for switchable power domain PDSW is enabled

Bit 0 – PDCFG Power Domain Configuration

Value	Name	Description
0x0	DEFAULT	In standby mode, all power domain switching are handled by hardware.
0x1	PDSW	In standby mode, PDSW is forced ACTIVE.

23. Reset Controller (RSTC)

23.1 Overview

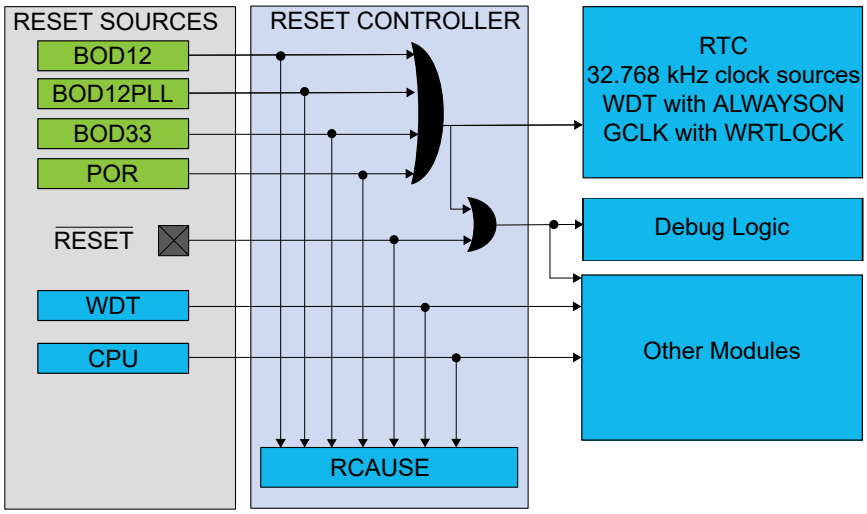
The Reset Controller (RSTC) manages the reset of the microcontroller. It issues a microcontroller reset, sets the device to its initial state and allows the reset source to be identified by software.

23.2 Features

- Reset the microcontroller and set it to an initial state according to the reset source
- Reset cause register for reading the reset source from the application code
- Multiple reset sources:
  - Power supply reset sources: POR, BOD12, BOD12PLL and BOD33
  - User reset sources: External reset ($\overline{\text{RESET}}$), Watchdog reset, and System Reset Request

23.3 Block Diagram

Figure 23-1. Reset System



23.4 Signal Description

Signal Name	Type	Description
$\overline{\text{RESET}}$	Digital input	External reset

One signal can be mapped on several pins.

23.5 Peripheral Dependencies

Table 23-1. RSTC Configuration Summary

Peripheral name	Base address	MCLK AHB/APB Clocks	PAC Peripheral Identifier Index (PAC.WRCTRL)	Power Domain (PM.STDBYCFG)
RSTC	0x40000C00	CLK_RSTC_APB	3	PDAO

23.6 Functional Description

23.6.1 Principle of Operation

The Reset Controller collects the various Reset sources and generates Reset for the device.

23.6.2 Basic Operation

23.6.2.1 Initialization

After a Power-on Reset (POR), the RSTC is enabled and the Reset Cause (RCAUSE) register indicates the POR source.

23.6.2.2 Enabling, Disabling, and Resetting

The RSTC module is always enabled.

23.6.2.3 Reset Causes and Effects

The latest Reset cause is available in RCAUSE register, and can be read during the application boot sequence in order to determine proper action.

These are the groups of Reset sources:

- Power supply Reset: Resets caused by an electrical issue. It covers POR and BODs Resets
- User Reset: Resets caused by the application. It covers external Resets, system Reset requests and watchdog Resets

The following table lists the parts of the device that are reset, depending on the Reset type.

Table 23-2. Effects of the Different Reset Causes

	Power Supply Reset	User Reset	
	POR, BOD33, BOD12, BOD12PLL	External Reset	WDT Reset, System Reset Request
RTC, OSC32KCTRL, RSTC	Y	N	N
GCLK with WRTLOCK	Y	N	N
Debug logic	Y	Y	N
Others	Y	Y	Y

The external Reset is generated when pulling the $\overline{\text{RESET}}$ pin low.

Note: External RESET valid active pulse width can be found on the Power Supply Electrical Specifications from Electrical Characteristics chapter.

The POR, BOD12, BOD12PLL and BOD33 Reset sources are generated by their corresponding module in the Supply Controller Interface (SUPC).

The WDT Reset is generated by the Watchdog Timer.

The System Reset Request is a Reset generated by the CPU when asserting the SYSRESETREQ bit located in the Reset Control register of the CPU (for details refer to the ARM® Cortex™ Technical Reference Manual on http://www.arm.com).

23.6.3 Sleep Mode Operation

The RSTC module is active in all sleep modes.

23.6.4 Debug Operation

When the CPU is halted in Debug mode, the RSTC continues normal operation.

23.7 Register Summary

Refer to the Registers Description section for more details on register properties and access permissions.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	RCAUSE	7:0		SYST	WDT	EXT	BOD12PLL	BOD33	BOD12	POR

23.7.1 Reset Cause

Name: RCAUSE
Offset: 0x00
Property: –

When a Reset occurs, the bit corresponding to the Reset source is set to '1' and all other bits are written to '0'.

Bit	7	6	5	4	3	2	1	0
		SYST	WDT	EXT	BOD12PLL	BOD33	BOD12	POR
Access		R	R	R	R	R	R	R
Reset		x	x	x	x	x	x	x

Bit 6 – SYST System Reset Request
This bit is set if a System Reset Request has occurred. Refer to the Cortex processor documentation for more details.

Bit 5 – WDT Watchdog Reset
This bit is set if a Watchdog Timer Reset has occurred.

Bit 4 – EXT External Reset
This bit is set if an external Reset has occurred.

Bit 3 – BOD12PLL Brown Out on VDDPLL Detector Reset
This bit is set if a BOD12PLL (VDDPLL) Reset has occurred.

Bit 2 – BOD33 Brown Out on AVDD/VDD Detector Reset
This bit is set if a BOD33 (AVDD/VDD) Reset has occurred.

Bit 1 – BOD12 Brown Out on VDDCORE Detector Reset
This bit is set if a BOD12 (VDDCORE) Reset has occurred.

Bit 0 – POR Power On Reset
This bit is set if a POR has occurred.

24. Watchdog Timer (WDT)

24.1 Overview

The Watchdog Timer (WDT) is a system function for monitoring correct program operation. It makes it possible to recover from error situations such as runaway or deadlocked code. The WDT is configured to a predefined time-out period, and is constantly running when enabled. If the WDT is not cleared within the time-out period, it will issue a system reset. An early-warning interrupt is available to indicate an upcoming watchdog time-out condition.

The window mode makes it possible to define a time slot (or window) inside the total time-out period during which the WDT must be cleared. If the WDT is cleared outside this window, either too early or too late, a system reset will be issued. Compared to the normal mode, this can also catch situations where a code error causes the WDT to be cleared frequently.

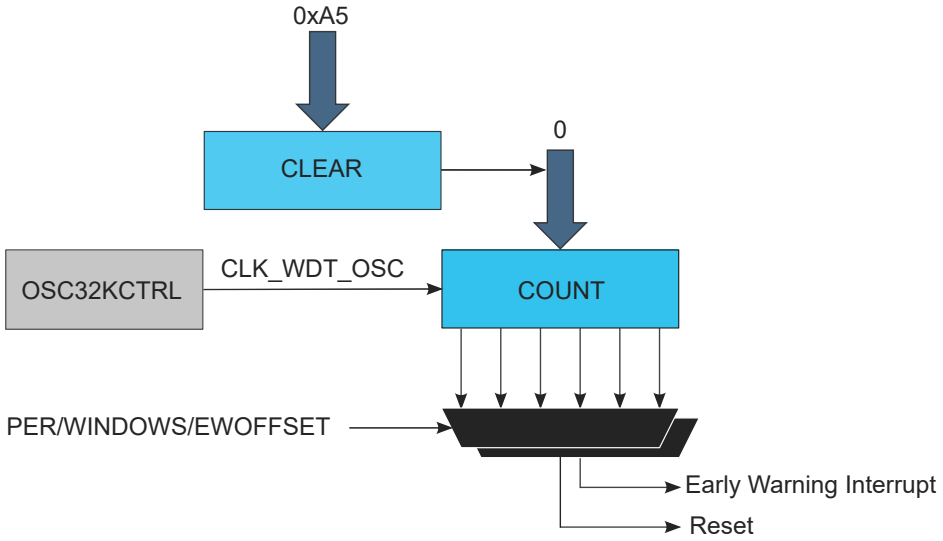
When enabled, the WDT will run in active mode and all sleep modes. It is asynchronous and runs from a CPU-independent clock source. The WDT will continue operation and issue a system reset or interrupt even if the main clocks fail.

24.2 Features

- Issues a system reset if the Watchdog Timer is not cleared before its time-out period
- Early Warning interrupt generation
- Asynchronous operation from dedicated oscillator
- Two types of operation
  - Normal
  - Window mode
- Selectable time-out periods
  - From 8 cycles to 16,384 cycles in Normal mode
  - From 16 cycles to 32,768 cycles in Window mode
- Always-On capability

24.3 Block Diagram

Figure 24-1. WDT Block Diagram



24.4 Peripheral Dependencies

Table 24-1. WDT Configuration Summary

Peripheral name	Base address	NVIC IRQ Index	MCLK AHB/APB Clocks	PAC Peripheral Identifier Index (PAC.WRCTRL)	Power Domain (PM.STDBYCFG)
WDT	0x40002000	1: EW	CLK_WDT_APB	8	PDSW

24.5 Functional Description

24.5.1 Principle of Operation

The Watchdog Timer (WDT) is a system for monitoring correct program operation, making it possible to recover from error situations such as runaway code, by issuing a Reset. When enabled, the WDT is a constantly running timer that is configured to a predefined time-out period. Before the end of the time-out period, the WDT should be set back, or else, a system Reset is issued.

The WDT has two modes of operation, Normal mode and Window mode. Both modes offer the option of Early Warning interrupt generation. The description for each of the basic modes is given below. The settings in the Control A register (CTRLA) and the Interrupt Enable register (handled by INTENCLR/INTENSET) determine the mode of operation:

Table 24-2. WDT Operating Modes

CTRLA.ENABLE	CTRLA.WEN	Interrupt Enable	Mode
0	x	x	Stopped
1	0	0	Normal mode
1	0	1	Normal mode with Early Warning interrupt
1	1	0	Window mode
1	1	1	Window mode with Early Warning interrupt

24.5.2 Basic Operation

24.5.2.1 Initialization

The following bits are enable-protected, meaning that they can only be written when the WDT is disabled (CTRLA.ENABLE=0):

- Control A register (CTRLA), except the Enable bit (CTRLA.ENABLE) and Always-On bit (CTRLA.ALWAYSON)
- Configuration register (CONFIG)
- Early Warning Interrupt Control register (EWCTRL)

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'.

The WDT can be configured only while the WDT is disabled. The WDT is configured by defining the required Time-Out Period bits in the Configuration register (CONFIG.PER). If Window mode operation is desired, the Window Enable bit in the Control A register must be set (CTRLA.WEN=1) and the Window Period bits in the Configuration register (CONFIG.WINDOW) must be defined.

Enable-protection is denoted by the "Enable-Protected" property in the register description.

24.5.2.2 Configurable Reset Values

After a Power-on Reset, some registers will be loaded with initial values from the NVM User Row.

This includes the following bits and bit groups:

- Enable bit in the Control A register, CTRLA.ENABLE
- Always-On bit in the Control A register, CTRLA.ALWAYSON
- Run In Standby Enable bit in the Control A register (CTRLA.RUNSTDBY)
- Watchdog Timer Windows Mode Enable bit in the Control A register, CTRLA.WEN
- Watchdog Timer Windows Mode Time-Out Period bits in the Configuration register, CONFIG.WINDOW
- Time-Out Period bits in the Configuration register, CONFIG.PER
- Early Warning Interrupt Time Offset bits in the Early Warning Interrupt Control register, EWCTRL.EWOFFSET

24.5.2.3 Enabling, Disabling, and Resetting

The WDT is enabled by writing a '1' to the Enable bit in the Control A register (CTRLA.ENABLE). The WDT is disabled by writing a '0' to CTRLA.ENABLE.

The WDT can be disabled only if the Always-On bit in the Control A register (CTRLA.ALWAYSON) is '0'.

24.5.2.4 Normal Mode

In Normal mode operation, the length of a time-out period is configured in CONFIG.PER. The WDT is enabled by writing a '1' to the Enable bit in the Control A register (CTRLA.ENABLE). Once enabled, the WDT will issue a system reset if a time-out occurs. This can be prevented by clearing the WDT at any time during the time-out period.

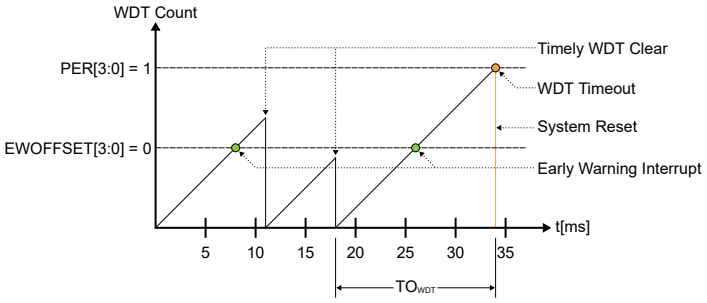
The WDT is cleared and a new WDT time-out period is started by writing 0xA5 to the Clear register (CLEAR). Writing any other value than 0xA5 to CLEAR will issue an immediate system reset.

There are 12 possible WDT time-out (TO_{WDT}) periods, selectable from 8ms to 16s.

By default, the early warning interrupt is disabled. If it is desired, the Early Warning Interrupt Enable bit in the Interrupt Enable register (INTENSET.EW) must be written to '1'. The Early Warning Interrupt is disabled again by writing a '1' to the Early Warning Interrupt bit in the Interrupt Enable Clear register (INTENCLR.EW).

If the Early Warning Interrupt is enabled, an interrupt is generated prior to a WDT time-out condition. In Normal mode, the Early Warning Offset bits in the Early Warning Interrupt Control register, EWCTRL.EWOFFSET, define the time when the early warning interrupt occurs. The Normal mode operation is illustrated in the figure Normal-Mode Operation.

Figure 24-2. Normal-Mode Operation



24.5.2.5 Window Mode

In Window mode operation, the WDT uses two different time specifications: the WDT can only be cleared by writing 0xA5 to the CLEAR register *after* the closed window time-out period (TO_{WDTW}), during the subsequent Normal time-out period (TO_{WDT}). If the WDT is cleared before the time window opens (before TO_{WDTW} is over), the WDT will issue a system reset.

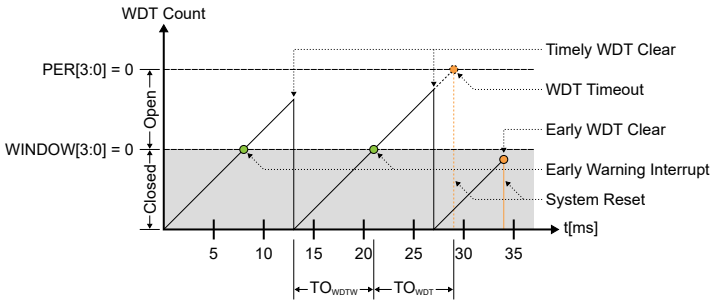
Both parameters TO_{WDTW} and TO_{WDT} are periods in a range from 8ms to 16s, so the total duration of the WDT time-out period is the sum of the two parameters.

The closed window period is defined by the Window Period bits in the Configuration register (CONFIG.WINDOW), and the open window period is defined by the Period bits in the Configuration register (CONFIG.PER).

By default, the Early Warning interrupt is disabled. If it is desired, the Early Warning Interrupt Enable bit in the Interrupt Enable register (INTENSET.EW) must be written to '1'. The Early Warning Interrupt is disabled again by writing a '1' to the Early Warning Interrupt bit in the Interrupt Enable Clear (INTENCLR.EW) register.

If the Early Warning interrupt is enabled in Window mode, the interrupt is generated at the start of the open window period, i.e. after TO_{WDTW}. The Window mode operation is illustrated in figure Window-Mode Operation.

Figure 24-3. Window-Mode Operation



24.5.3 Clocks

The WDT bus clock (CLK_WDT_APB) can be enabled and disabled (masked) in the Main Clock module (MCLK).

A 1024 Hz oscillator clock (CLK_WDT_OSC) is required to clock the WDT internal counter.

The CLK_WDT_OSC clock is sourced from the clock of the internal Ultra Low-Power Oscillator (OSCULP32K).

⚠ CAUTION Watchdog time-out period variations must be considered when implementing software that uses the WDT to ensure that the time-out periods used are valid for all devices. Refer to the OSCULP32K Electrical Specifications section of the Electrical Characteristics chapter.

The counter clock CLK_WDT_OSC is asynchronous to the bus clock (CLK_WDT_APB). Due to this asynchronicity, writing to certain registers will require synchronization between the clock domains. Refer to Synchronization for further details.

24.5.4 Interrupts

The WDT has the following interrupt source:

- Early Warning (EW): Indicates that the counter is approaching the time-out condition.
  - This interrupt is an asynchronous wake-up source.

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 WDT is reset. See the 24.6.6. INTFLAG register description 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 11.2. Nested Vector Interrupt Controller. 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.

24.5.5 Sleep Mode Operation

The Run-In-Standby bit in Control A (CTRLA.RUNSTDBY) control the behavior of the WDT during standby sleep mode. When the bit is zero, the watchdog is disabled during sleep, but maintains its current configuration. When CTRLA.RUNSTDBY is '1', the WDT continues to operate during sleep.

24.5.6 Debug Operation

When the CPU is halted in debug mode the WDT will halt normal operation.

24.5.7 Synchronization

Some registers (or bit fields within a register) require synchronization when read and/or written.

Synchronization is denoted by the "Read-Synchronized" (or "Read-Synchronized Bits") and/or "Write-Synchronized" (or "Write-Synchronized Bits") property in each individual register description.

For more details, refer to Register Synchronization.

24.5.8 Additional Features

24.5.8.1 Always-On Mode

The Always-On mode is enabled by setting the Always-On bit in the Control A register (CTRLA.ALWAYSON=1). When the Always-On mode is enabled, the WDT runs continuously, regardless of the state of CTRLA.ENABLE. Once written, the Always-On bit can only be cleared by a power-on reset. The Configuration (CONFIG) and Early Warning Control (EWCTRL) registers are read-only registers while the CTRLA.ALWAYSON bit is set. Thus, the time period configuration bits (CONFIG.PER, CONFIG.WINDOW, EWCTRL.EWOFFSET) of the WDT cannot be changed.

Enabling or disabling Window mode operation by writing the Window Enable bit (CTRLA.WEN) is allowed while in Always-On mode, but note that CONFIG.PER cannot be changed.

The Interrupt Clear and Interrupt Set registers are accessible in the Always-On mode. The Early Warning interrupt can still be enabled or disabled while in the Always-On mode, but note that EWCTRL.EWOFFSET cannot be changed.

Table WDT Operating Modes With Always-On shows the operation of the WDT for CTRLA.ALWAYSON=1.

Table 24-3. WDT Operating Modes With Always-On

WEN	Interrupt Enable	Mode
0	0	Always-on and normal mode
0	1	Always-on and normal mode with Early Warning interrupt
1	0	Always-on and window mode
1	1	Always-on and window mode with Early Warning interrupt

24.5.8.2 Early Warning

The Early Warning interrupt notifies that the WDT is approaching its time-out condition. The Early Warning interrupt behaves differently in Normal mode and in Window mode.

In Normal mode, the Early Warning interrupt generation is defined by the Early Warning Offset in the Early Warning Control register (EWCTRL.EWOFFSET). The Early Warning Offset bits define the number of CLK_WDT_OSC clocks before the interrupt is generated, relative to the start of the watchdog time-out period.

The user must take caution when programming the Early Warning Offset bits. If these bits define an Early Warning interrupt generation time greater than the watchdog time-out period, the watchdog time-out system reset is generated prior to the Early Warning interrupt. Consequently, the Early Warning interrupt will never be generated.

In window mode, the Early Warning interrupt is generated at the start of the open window period. In a typical application where the system is in sleep mode, the Early Warning interrupt can be used to wake up and clear the Watchdog Timer, after which the system can perform other tasks or return to sleep mode.

> If the WDT is operating in Normal mode with CONFIG.PER = 0x2 and EWCTRL.EWOFFSET = 0x1, the Early Warning interrupt is generated 16 CLK_WDT_OSC clock cycles after the start of the time-out period. The time-out system reset is generated 32 CLK_WDT_OSC clock cycles after the start of the watchdog time-out period.

24.6 Register Summary

Refer to the Registers Description section for more details on register properties and access permissions.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRLA	7:0	ALWAYSON	RUNSTDBY				WEN	ENABLE	
0x01	CONFIG	7:0	WINDOW[3:0]				PER[3:0]			
0x02	EWCTRL	7:0					EWOFFSET[3:0]			
0x03	Reserved									
0x04	INTENCLR	7:0								EW
0x05	INTENSET	7:0								EW
0x06	INTFLAG	7:0								EW
0x07	Reserved									
0x08	SYNCBUSY	31:24								
		23:16								
		15:8								
		7:0			CLEAR	ALWAYSON	RUNSTDBY	WEN	ENABLE	
0x0C	CLEAR	7:0	CLEAR[7:0]							

24.6.1 Control A

Name: CTRLA
Offset: 0x00
Reset: x initially determined from NVM User Row after reset
Property: PAC Write-Protection, Write-Synchronized Bits, Enable-Protected Bits

Bit	7	6	5	4	3	2	1	0
	ALWAYSON	RUNSTDBY				WEN	ENABLE	
Access	R/W	R/W				R/W	R/W	
Reset	x	x				x	x	

Bit 7 – ALWAYSON Always-On
This bit allows the WDT to run continuously. After being set, this bit cannot be written to '0', and the WDT will remain enabled until a power-on Reset is received. When this bit is '1', the Control A register (CTRLA), the Configuration register (CONFIG) and the Early Warning Control register (EWCTRL) will be read-only, and any writes to these registers are not allowed.
Writing a '0' to this bit has no effect.
This bit is loaded from NVM User Row at start-up.
Note: This bit is not enable-protected.

Value	Description
0	The WDT is enabled and disabled through the ENABLE bit.
1	The WDT is enabled and can only be disabled by a power-on reset (POR).

Bit 6 – RUNSTDBY Run in Standby
This bit controls the behavior of the watchdog during standby sleep mode.

- When CTRLA.ALWAYSON=0, this bit is enable-protected by CTRLA.ENABLE.
- When CTRLA.ALWAYSON=1, this bit is not enable-protected by CTRLA.ENABLE.

These bits are loaded from NVM User Row at startup.

Value	Description
0	The WDT is disabled during standby sleep.
1	The WDT is enabled continues to operate during standby sleep.

Bit 2 – WEN Watchdog Timer Window Mode Enable
This bit enables Window mode.

- When CTRLA.ALWAYSON=0, this bit is enable-protected by CTRLA.ENABLE.
- When CTRLA.ALWAYSON=1, this bit is not enable-protected by CTRLA.ENABLE.

This bit is loaded from NVM User Row at startup.

Value	Description
0	Window mode is disabled (normal operation).
1	Window mode is enabled.

Bit 1 – ENABLE Enable
This bit enables or disables the WDT. It can only be written if CTRLA.ALWAYSON=0.
This bit is loaded from NVM User Row at startup.
Note: This bit is write-synchronized: SYNCBUSY.ENABLE must be checked to ensure the CTRLA.ENABLE synchronization is complete.

Note: This bit is not enable-protected.

Value	Description
0	The WDT is disabled.
1	The WDT is enabled.

24.6.2 Configuration

Name: CONFIG
Offset: 0x01
Reset: x initially determined from NVM User Row after reset
Property: PAC Write-Protection, Enable-Protected

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

Bits 7:4 – WINDOW[3:0] Window Mode Time-Out Period
In Window mode, these bits determine the watchdog closed window period as a number of cycles of the 1024 Hz CLK_WDT_OSC clock.
These bits are loaded from NVM User Row at start-up.

Value	Name	Description
0x0	CYC8	8 clock cycles
0x1	CYC16	16 clock cycles
0x2	CYC32	32 clock cycles
0x3	CYC64	64 clock cycles
0x4	CYC128	128 clock cycles
0x5	CYC256	256 clock cycles
0x6	CYC512	512 clock cycles
0x7	CYC1024	1024 clock cycles
0x8	CYC2048	2048 clock cycles
0x9	CYC4096	4096 clock cycles
0xA	CYC8192	8192 clock cycles
0xB	CYC16384	16384 clock cycles
0xC-0xF	Reserved	Reserved

Bits 3:0 – PER[3:0] Time-Out Period
These bits determine the watchdog time-out period as a number of 1024 Hz CLK_WDTOSC clock cycles. In Window mode operation, these bits define the open window period.
These bits are loaded from NVM User Row at startup.

Value	Name	Description
0x0	CYC8	8 clock cycles
0x1	CYC16	16 clock cycles
0x2	CYC32	32 clock cycles
0x3	CYC64	64 clock cycles
0x4	CYC128	128 clock cycles
0x5	CYC256	256 clock cycles
0x6	CYC512	512 clock cycles
0x7	CYC1024	1024 clock cycles
0x8	CYC2048	2048 clock cycles
0x9	CYC4096	4096 clock cycles
0xA	CYC8192	8192 clock cycles
0xB	CYC16384	16384 clock cycles
0xC – 0xF	-	Reserved

24.6.3 Early Warning Control

Name: EWCTRL
Offset: 0x02
Reset: x initially determined from NVM User Row after reset
Property: PAC Write-Protection, Enable-Protected

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

Bits 3:0 – EWOFFSET[3:0] Early Warning Interrupt Time Offset
These bits determine the number of GCLK_WDT clock cycles between the start of the watchdog time-out period and the generation of the Early Warning interrupt. These bits are loaded from NVM User Row at start-up.

Value	Name	Description
0x0	CYC8	8 clock cycles
0x1	CYC16	16 clock cycles
0x2	CYC32	32 clock cycles
0x3	CYC64	64 clock cycles
0x4	CYC128	128 clock cycles
0x5	CYC256	256 clock cycles
0x6	CYC512	512 clock cycles
0x7	CYC1024	1024 clock cycles
0x8	CYC2048	2048 clock cycles
0x9	CYC4096	4096 clock cycles
0xA	CYC8192	8192 clock cycles
0xB - 0xF	Reserved	Reserved

24.6.4 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 (INTENSET) register.

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

Bit 0 – EW Early Warning Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Early Warning Interrupt Enable bit, which disables the Early Warning interrupt.

Value	Description
0	The Early Warning interrupt is disabled.
1	The Early Warning interrupt is enabled.

24.6.5 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 (INTENCLR) register.

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

Bit 0 – EW Early Warning Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit sets the Early Warning Interrupt Enable bit, which enables the Early Warning interrupt.

Value	Description
0	The Early Warning interrupt is disabled.
1	The Early Warning interrupt is enabled.

24.6.6 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x06
Reset: 0x00
Property: -

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

Bit 0 – EW Early Warning
This flag is cleared by writing a '1' to it.
This flag is set when an Early Warning interrupt occurs, as defined by the EWOFFSET bit group in EWCTRL.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Early Warning interrupt flag.

24.6.7 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
			CLEAR	ALWAYSON	RUNSTDBY	WEN	ENABLE	
Access			R	R	R	R	R	
Reset			0	0	0	0	0	

Bit 5 – CLEAR Clear Synchronization Busy

Value	Description
0	Write synchronization of the CLEAR register is complete.
1	Write synchronization of the CLEAR register is ongoing.

Bit 4 – ALWAYSON Always-On Synchronization Busy

Value	Description
0	Write synchronization of the CTRLA.ALWAYSON bit is complete.
1	Write synchronization of the CTRLA.ALWAYSON bit is ongoing.

Bit 3 – RUNSTDBY Run-In-Standby Synchronization Busy

Value	Description
0	Write synchronization of the CTRLA.RUNSTDBY bit is complete.
1	Write synchronization of the CTRLA.RUNSTDBY bit is ongoing.

Bit 2 – WEN Window Enable Synchronization Busy

Value	Description
0	Write synchronization of the CTRLA.WEN bit is complete.
1	Write synchronization of the CTRLA.WEN bit is ongoing.

Bit 1 – ENABLE Enable Synchronization Busy

Value	Description
0	Write synchronization of the CTRLA.ENABLE bit is complete.
1	Write synchronization of the CTRLA.ENABLE bit is ongoing.

24.6.8 Clear

Name: CLEAR
Offset: 0x0C
Reset: 0x00
Property: Write-Synchronized

Note: This register is write-synchronized: SYNCBUSY.CLEAR must be checked to ensure the CLEAR register synchronization is complete.

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

Bits 7:0 – CLEAR[7:0] Watchdog Clear
In Normal mode, writing 0xA5 to this register during the watchdog time-out period will clear the Watchdog Timer and the watchdog time-out period is restarted.
In Window mode, any writing attempt to this register before the time-out period started (i.e., during TO_{WDTW}) will issue an immediate system Reset. Writing 0xA5 during the time-out period TO_{WDT} will clear the Watchdog Timer and the complete time-out sequence (first TO_{WDTW} then TO_{WDT}) is restarted.
In both modes, writing any other value than 0xA5 will issue an immediate system Reset.
Note: This bit field is write-synchronized: SYNCBUSY.CLEAR must be checked to ensure the CLEAR.CLEAR synchronization is complete.

25. Real-Time Counter (RTC)

25.1 Overview

The Real-Time Counter (RTC) is a 32-bit counter with a 10-bit programmable prescaler that typically runs continuously to keep track of time. The RTC can wake up the device from sleep modes using the alarm/compare wake up, periodic wake up, or overflow wake up mechanisms, or from the wake inputs.

The RTC can generate periodic peripheral events from outputs of the prescaler, as well as alarm/compare interrupts and peripheral events, which can trigger at any counter value. Additionally, the timer can trigger an overflow interrupt and peripheral event, and can be reset on the occurrence of an alarm/compare match. This allows periodic interrupts and peripheral events at very long and accurate intervals.

The 10-bit programmable prescaler can scale down the clock source. By this, a wide range of resolutions and time-out periods can be configured. With a 32.768kHz clock source, the minimum counter tick interval is 30.5µs, and time-out periods can range up to 36 hours. For a counter tick interval of 1s, the maximum time-out period is more than 136 years.

25.2 Features

- 32-bit counter with 10-bit prescaler
- Multiple clock sources
- 32-bit or 16-bit counter mode
- One 32-bit or two 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
- 2 general purpose registers
- Tamper Detection
  - Up to 8 tamper inputs with programmable level detection (for on and off switches at the product enclosure level)
  - Up to 8 tamper inputs and 8 tamper outputs (for Active Protection against PCB Physical Tampering)
  - Timestamp on tampers event

25.3 Block Diagram

Figure 25-1. RTC Block Diagram (Mode 0 — 32-Bit Counter)

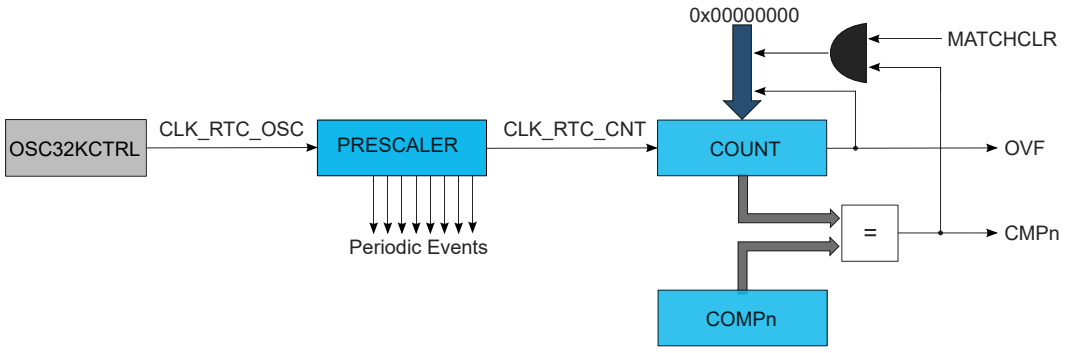


Figure 25-2. RTC Block Diagram (Mode 1 — 16-Bit Counter)

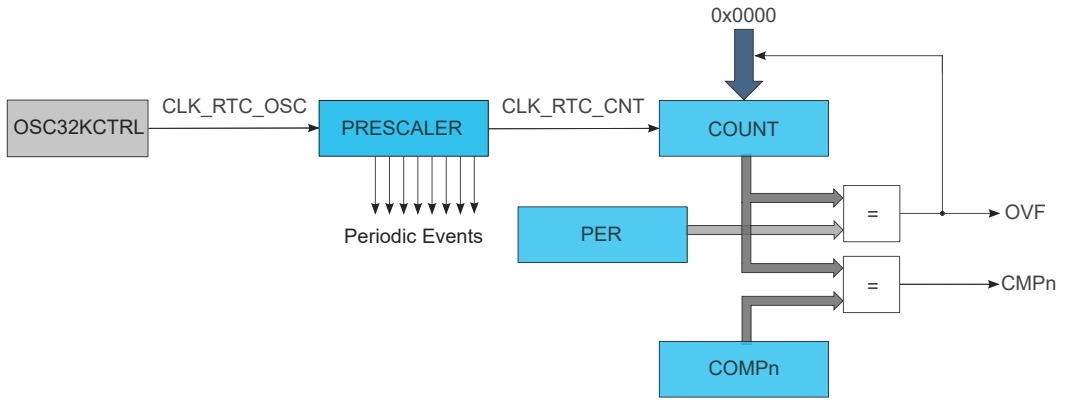


Figure 25-3. RTC Block Diagram (Mode 2 — Clock/Calendar)

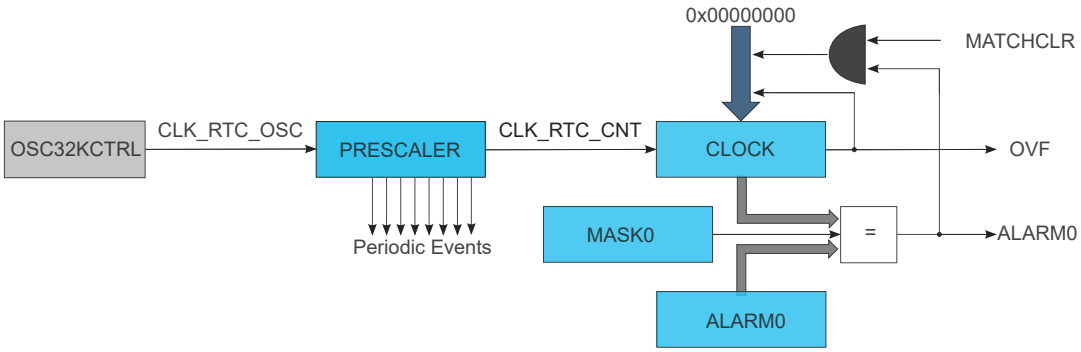
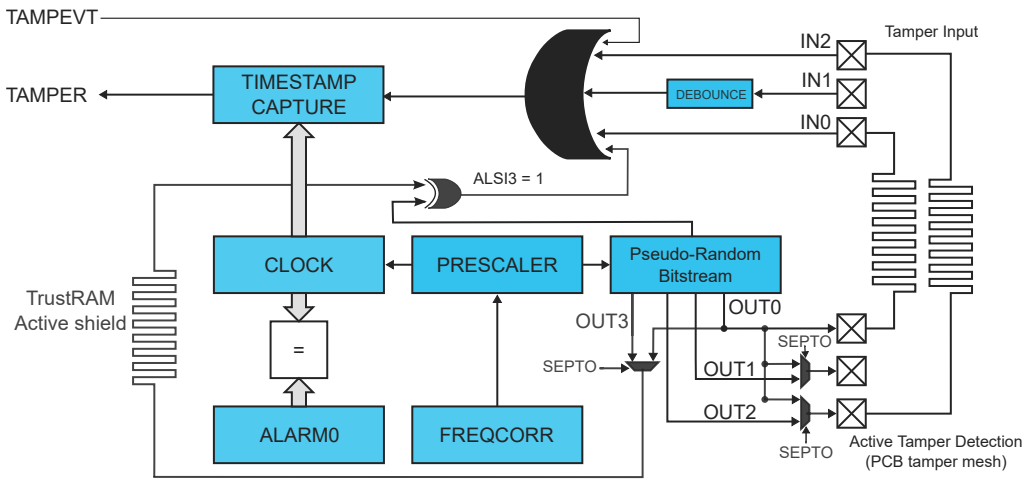


Figure 25-4. RTC Block Diagram (Tamper Detection Use Case with four tamper inputs/outputs)



25.4 Signal Description

Table 25-1. Signal Description

Signal	Description	Type
INn [n=0..7]	Tamper Detection Input	Digital input
OUTn [n=0..7]	Tamper Detection Output	Digital output

One signal can be mapped to one of several pins.

25.5 Peripheral Dependencies

Table 25-2. RTC Configuration Summary

Peripheral name	Base address	NVIC IRQ Index	MCLK AHB/APB Clocks	PAC Peripheral Identifier Index (PAC.WRCTRL)	DMA Trigger Source Index (DMAC.CHCTRLB)	Events (EVSYS)		Power Domain (PM.STDBYCFG)
						Users (EVSYS.USER)	Generators (EVSYS.CHANNEL)	
RTC	0x40002400	2: CMP0-1, TAMPER, OVF, PER0-7, ALARM0	CLK_RTC_APB	9	1: TIMESTAMP	1: TAMPEVT	4-11 : PER0-7 12 : ALARM0 12-13 : CMP0-1 14 : TAMPER 15 : OVF 16 : PERD	PDAO

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)
- GP Registers Reset On Tamper Enable bit in the Control A register (CTRLA.GPTRST)

The following registers are enable-protected:

- Control B register (CTRLB)
- Event Control register (EVCTRL)
- Tamper Control register (TAMPCTRL)
- Tamper Control B register (TAMPCTRLB)

Enable-protected bits and registers can be changed only when the RTC is disabled (CTRLA.ENABLE=0). If the RTC is enabled (CTRLA.ENABLE=1), these operations are necessary: first write CTRLA.ENABLE=0 and check whether the write synchronization has finished, then change the desired bit field value. Enable-protected bits in CTRLA register can be written at the same time as CTRLA.ENABLE is written to '1', but not at the same time as CTRLA.ENABLE is written to '0'.

Enable-protection is denoted by the "Enable-Protected" property in the register description.

The RTC prescaler divides the source clock for the RTC counter.

Note: In Clock/Calendar mode, the prescaler must be configured to provide a 1Hz clock to the counter for correct operation.

The frequency of the RTC clock (CLK_RTC_CNT) is given by the following formula:

$$f_{CLK\_RTC\_CNT} = \frac{f_{CLK\_RTC\_OSC}}{2^{PRESCALER}}$$

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. The block diagram of this mode is shown in Figure 25-1. When the RTC is enabled, the counter will increment on every 0-to-1 transition of CLK_RTC_CNT. The counter will increment until it reaches the top value of 0xFFFFFFFF, and then wrap to 0x00000000. This sets the Overflow Interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.OVF).

The RTC counter value can be read from or written to the Counter Value register (COUNT) in 32-bit format.

The counter value is continuously compared with the 32-bit Compare register (COMP0). When a compare match occurs, the Compare Interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.CMP0) 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 COMP0 register 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.CMP0 and INTFLAG.OVF will both be set simultaneously on a compare match with COMP0 register.

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 as shown in Figure 25-2. 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..1). When a compare match occurs, the Compare n Interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.CMPn, n=0..1) is set on the next 0-to-1 transition of CLK_RTC_CNT.

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, as shown in Figure 25-3. When the RTC is enabled, the counter will increment on every 0-to-1 transition of CLK_RTC_CNT. The selected clock source and RTC prescaler must be configured to provide a 1Hz clock to the counter for correct operation in this mode.

The time and date can be read from or written to the Clock Value register (CLOCK) in a 32-bit time/date format. Time is represented as:

- Seconds
- Minutes
- Hours

Hours can be represented in either 12- or 24-hour format, selected by the Clock Representation bit in the Control A register (CTRLA.CLKREP). This bit can be changed only while the RTC is disabled.

The date is represented in this form:

- Day as the numeric day of the month (starting at 1)
- Month as the numeric month of the year (1 = January, 2 = February, etc.)
- Year as a value from 0x00 to 0x3F. This value must be added to a user-defined reference year. The reference year must be a leap year (2016, 2020 etc). Example: the year value 0x2D, added to a reference year 2016, represents the year 2061.

The RTC will increment until it reaches the top value of 23:59:59 December 31 of year value 0x3F, and then wrap to 00:00:00 January 1 of year value 0x00. This will set the Overflow Interrupt flag in the Interrupt Flag Status and Clear registers (INTFLAG.OVF).

The clock value is continuously compared with the 32-bit Alarm register (ALARM0). When an alarm match occurs, the Alarm Interrupt flag in the Interrupt Flag Status and Clear registers (INTFLAG.ALARM0) is set on the next 0-to-1 transition of CLK_RTC_CNT. E.g. For a 1Hz clock counter, it means the Alarm Interrupt flag is set with a delay of 1s after the occurrence of alarm match.

A valid alarm match depends on the setting of the Alarm Mask Selection bits in the Alarm Mask register (MASK0.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 ALARM0 occurs. This allows the RTC to generate periodic interrupts or events with longer periods than it would be possible with the prescaler events only (see 25.6.10.1. Periodic Intervals).
Note: When CTRLA.MATCHCLR is 1, INTFLAG.ALARM0 and INTFLAG.OVF will both be set simultaneously on an alarm match with ALARM0 register.

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 Clocks

The RTC bus clock (CLK_RTC_APB) can be enabled and disabled in the Main Clock module MCLK, and the default state of CLK_RTC_APB can be found in 18.5.2.6. Peripheral Clock Masking section.

A 32.768 kHz or 1024 Hz oscillator clock (CLK_RTC_OSC) is required to clock the RTC. This clock must be configured and enabled in the 32.768 kHz oscillator controller (OSC32KCTRL) before using the RTC.

This oscillator clock is asynchronous to the bus clock (CLK_RTC_APB). Due to this asynchronicity, writing to certain registers will require synchronization between the clock domains. Refer to Synchronization for further details.

25.6.5 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 n (CMP0-1): Indicates a match between the counter value and the compare register.
- Alarm 0 (ALARM0): Indicates a match between the clock value and the alarm register.
- Period n (PER0-7): The corresponding bit in the prescaler has toggled. Refer to 25.6.10.1. Periodic Intervals for details.

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).

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. Refer to the Nested Vector Interrupt Controller for details. 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. Refer to the Nested Vector Interrupt Controller for details.

25.6.6 Events

The RTC can generate the following output events:

- Overflow (OVF): Generated when the counter has reached its top value and wrapped to zero
- Tamper (TAMPER): Generated on detection of valid signal on a tamper input pin or tamper event input
- Compare n (CMP0-1): Indicates a match between the counter value and the compare register (COUNT16 and COUNT32 modes)
- Alarm 0 (ALARM0): Indicates a match between the clock value and the alarm register (Clock/Calendar mode).
- Period n (PER0-7): The corresponding bit in the prescaler has toggled. Refer to 25.6.10.1. Periodic Intervals for details.
- Periodic Daily (PERD): Generated when the COUNT/CLOCK has incremented at a fixed period of time

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. Refer to the EVSYS - Event System for 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.

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.

25.6.7 Sleep Mode Operation

The RTC will continue to operate in any sleep mode where the source clock is active. The RTC *interrupts* can be used to wake up the device from a sleep mode. RTC *events* can trigger other operations in the system without exiting the sleep mode.

An interrupt request will be generated after the wake-up if the Interrupt Controller is configured accordingly. Otherwise the CPU will wake up directly, without triggering any interrupt. In this case, the CPU will continue 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 for more information.

25.6.8 Debug Operation

When the CPU is halted in debug mode the RTC will halt normal operation. The RTC can be forced to continue operation during debugging. Refer to 25.7.7. DBGCTRL for details.

25.6.9 Synchronization

Some registers (or bit fields within a register) require synchronization when read and/or written.

Synchronization is denoted by the "Read-Synchronized" (or "Read-Synchronized Bits") and/or "Write-Synchronized" (or "Write-Synchronized Bits") property in each individual register description.

For more details, refer to Register Synchronization.

25.6.10 Additional Features

25.6.10.1 Periodic Intervals

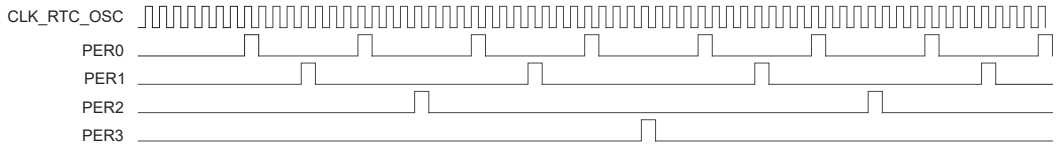
The RTC prescaler can generate interrupts and events at periodic intervals, allowing flexible system tick creation. Any of the upper eight bits of the prescaler (bits 2 to 9) can be the source of an interrupt/event. When one of the eight Periodic Event Output bits in the Event Control register (EVCTRL.PEREO[n=0..7]) 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 will generate an event every eight CLK_RTC_OSC cycles, PER1 every 16 cycles, etc. This is shown in the figure below.

Periodic events are independent of the prescaler setting used by the RTC counter, except if CTRLA.PRESCALER is zero. Then, no periodic events will be generated.

Figure 25-5. Example Periodic Events



25.6.10.2 Frequency Correction

The RTC Frequency Correction module employs periodic counter corrections to compensate for a too-slow or too-fast oscillator. Frequency correction requires that CTRLA.PRESCALER is greater than 1.

The digital correction circuit adds or subtracts cycles from the RTC prescaler to adjust the frequency in approximately 1ppm steps. Digital correction is achieved by adding or skipping a single count in the prescaler once every 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 0.95367ppm.

The Sign bit in the Frequency Correction register (FREQCORR.SIGN) determines the direction of the correction. A positive value will add counts and increase the period (reducing the frequency), and a negative value will reduce counts per period (speeding up the frequency).

Digital correction also affects the generation of the periodic events from the prescaler. When the correction is applied at the end of the correction cycle period, the interval between the previous periodic event and the next occurrence may also be shortened or lengthened depending on the correction value.

25.6.10.3 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 have a separate read buffer and write buffer. When the general purpose feature is enabled the even GP uses the read buffer while the odd GP uses the write buffer.

When the Compare / Alarm register is written, the write buffer hold temporarily the Compare / Alarm value until the synchronization is completed (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 Compare / Alarm register must be written with the correct value.

An example procedure to write the general purpose registers GP0 and GP1 is:

1. Wait for any ongoing write to the Compare register to complete (SYNCBUSY.COMP0 = 0). If the RTC is operating in Mode 1, wait for any ongoing write to COMP1 register to complete as well (SYNCBUSY.COMP1 = 0).
2. Write CTRLB.GP0EN = 1 if GP0 is needed.
3. Write GP0.
4. Wait for any ongoing write to GP0 to complete (SYNCBUSY.GP0 = 0). Note that GP1 will also show as busy when GP0 is busy.
5. Write GP1 if needed.

The following table provides the correspondence of General Purpose Registers and the Compare / Alarm read or write buffer in all RTC modes.

Table 25-3. General Purpose Registers Versus Compare/Alarm Registers: n in 0, 2, 4, 6...

Register	Mode 0	Mode 1	Mode 2	Write Before
GPn	COMP0 write buffer	(COMPn , COMPn+1) write buffer	ALARM0 write buffer	GPn+1
GPn+1	COMP0 read buffer	(COMPn , COMPn+1) read buffer	ALARM0 read buffer	-

25.6.10.4 Tamper Detection

The RTC provides eight tamper channels that can be used for tamper detection.

The action of each tamper channel is configured using the Input n Action bits in the Tamper Control register (TAMPCTRL.INnACT):

- **Off:** Detection for tamper channel n is disabled.
- **Wake:** A transition on INn input (tamper channel n) matching TAMPCTRL.TAMPLVLn will be detected and the tamper interrupt flag (INTFLAG.TAMPER) will be set. The RTC value will not be captured in the TIMESTAMP register.
- **Capture:** A transition on INn input (tamper channel n) matching TAMPCTRL.TAMPLVLn will be detected and the tamper interrupt flag (INTFLAG.TAMPER) will be set. The RTC value will be captured in the TIMESTAMP register.
- **Active Tamper Detection:** A mismatch of an internal RTC signal routed between INn and OUTn pins will be detected and the tamper interrupt flag (INTFLAG.TAMPER) will be set. The RTC value will be captured in the TIMESTAMP register.

In order to determine which tamper source caused a tamper event, the Tamper ID register (TAMPID) provides the detection status of each tamper channel. These bits remain active until cleared by software.

A single interrupt request (TAMPER) is available for all tamper channels.

The RTC also supports an input event (TAMPEVT) for generating a tamper condition within the Event System. The tamper input event is enabled by the Tamper Input Event Enable bit in the Event Control register (EVCTRL.TAMPEVEI).

Up to four polarity external inputs (INn) can be used for tamper detection. The polarity for each input is selected with the Tamper Level bits in the Tamper Control register (TAMPCTRL.TAMPLVLn).

Separate debouncers are embedded for each external input. The debouncer for each input is enabled/disabled with the Debounce Enable bits in the Tamper Control register (TAMPCTRL.DEBNCn). The debouncer configuration is fixed for all inputs as set by the Control B register (CTRLB). The debouncing period duration is configurable using the Debounce Frequency field in the Control B register (CTRLB.DEBF). The period is set for all debouncers (i.e., the duration cannot be adjusted separately for each debouncer).

When TAMPCTRL.DEBNCn = 0, INn is detected asynchronously. See the *Edge Detection with Debouncer Disabled* figure for an example.

When TAMPCTRL.DEBNCn = 1, the detection time depends on whether the debouncer operates synchronously or asynchronously, and whether majority detection is enabled or not. Refer to the table below for more details. Synchronous versus asynchronous stability debouncing is configured by the Debounce Asynchronous Enable bit in the Control B register (CTRLB.DEBASYNC):

- **Synchronous (CTRLB.DEBASYNC = 0):** INn is synchronized in two CLK_RTC periods and then must remain stable for four CLK_RTC_DEB periods before a valid detection occurs. See the *Edge Detection with Synchronous Stability Debouncing* figure for an example.
- **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. See the *Edge Detection with Asynchronous Stability Debouncing* figure for an example.

Majority debouncing is configured by the Debounce Majority Enable bit in the Control B register (CTRLB.DEBMAJ). INn must be valid for two out of three CLK_RTC_DEB periods. See the *Edge Detection with Majority Debouncing* figure for an example.

Table 25-4. 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 4 consecutive CLK_RTC_DEB periods.
1	0	1	Detect edge on INn with asynchronous stability debouncing. First detected edge is triggered immediately. All subsequent detected edges are ignored until INn is stable for 4 consecutive CLK_RTC_DEB periods.
1	1	X	Detect edge on INn with majority debouncing. Pin INn is sampled for 3 consecutive CLK_RTC_DEB periods. Signal level is determined by majority-rule (LLL, LLH, LHL, HLL = '0' and LHH, HLH, HHL, HHH = '1').

Figure 25-6. Edge Detection with Debouncer Disabled

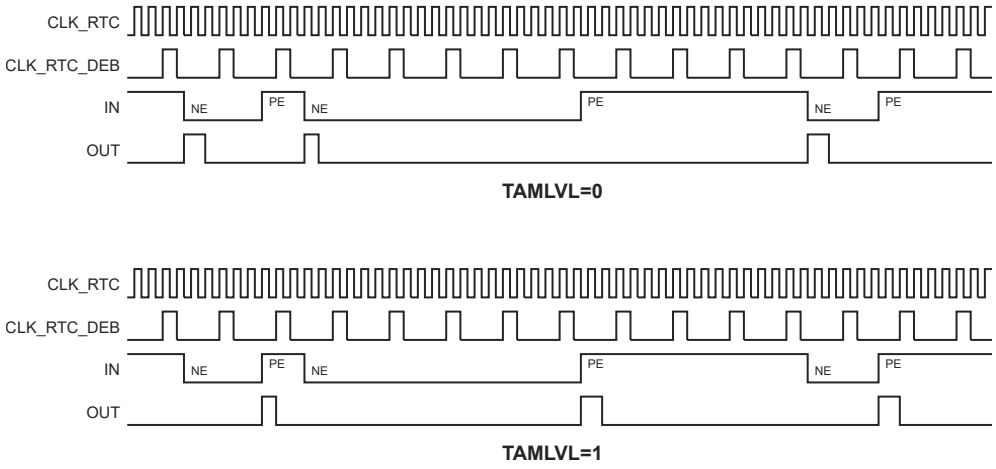


Figure 25-7. Edge Detection with Synchronous Stability Debouncing

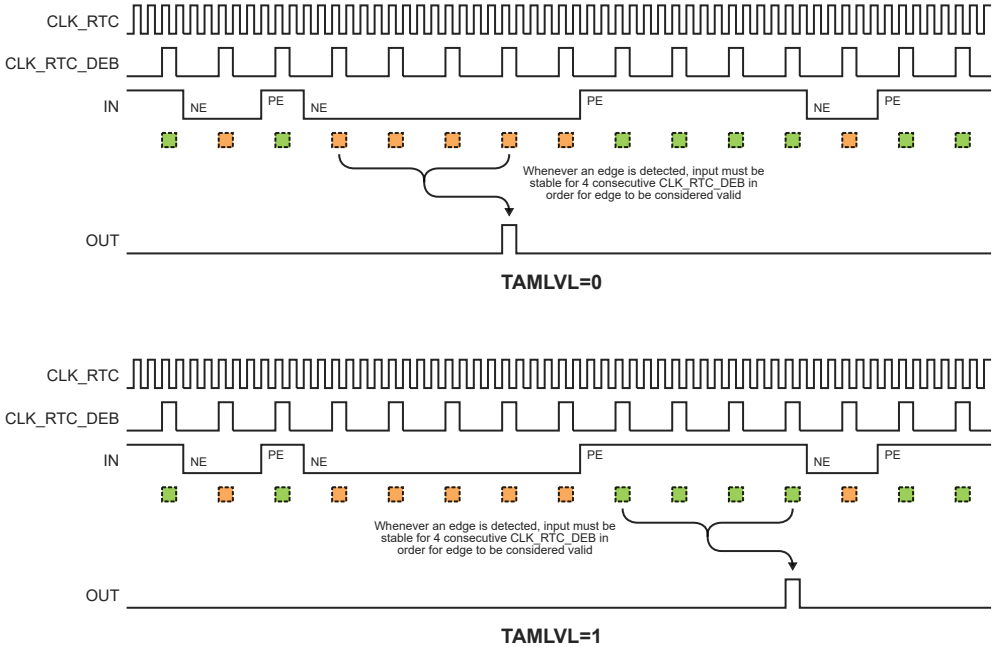


Figure 25-8. Edge Detection with Asynchronous Stability Debouncing

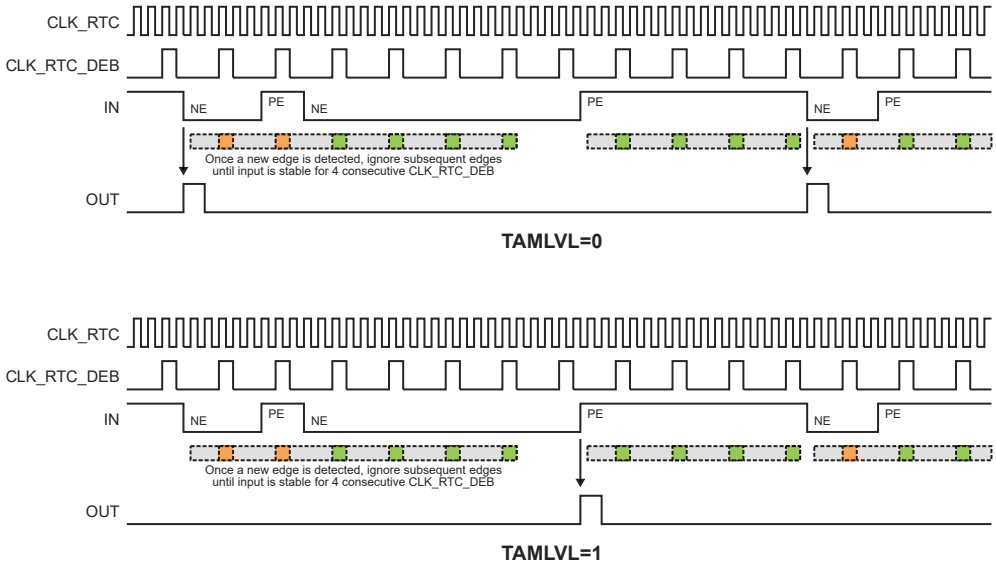
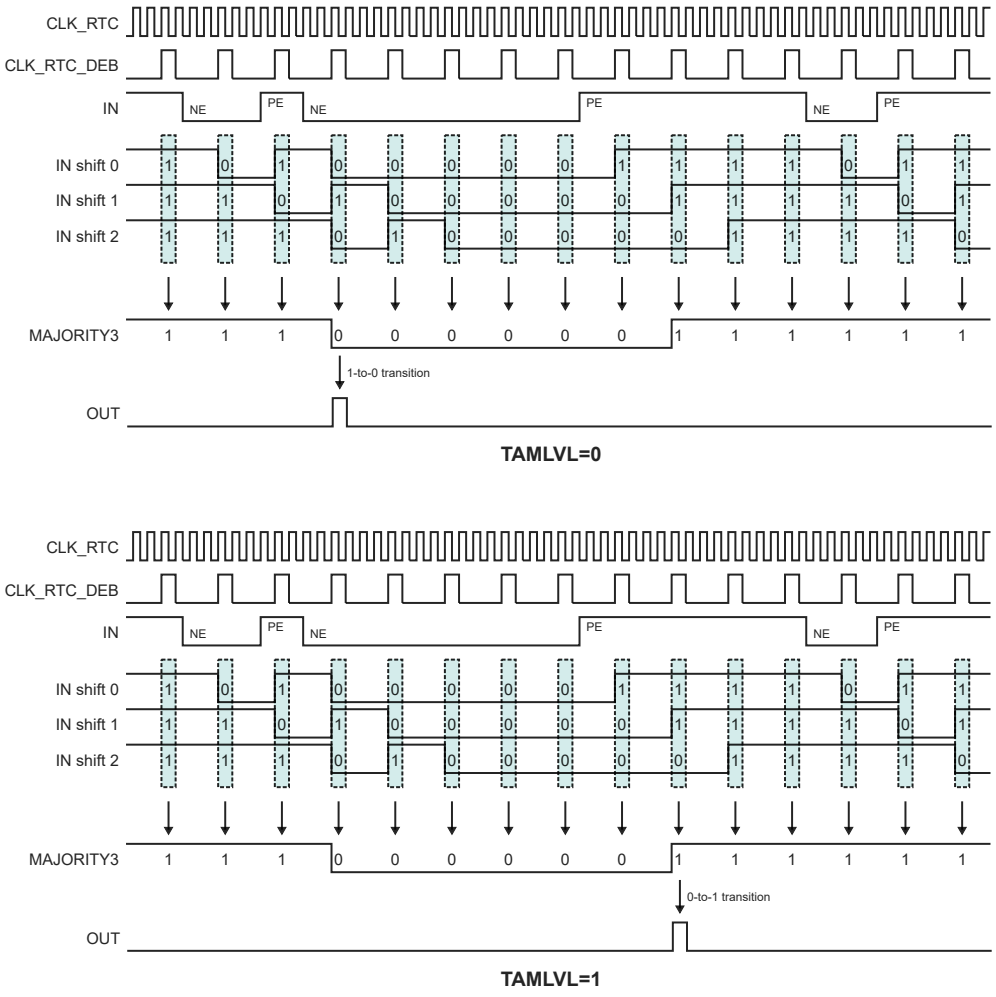


Figure 25-9. Edge Detection with Majority Debouncing



25.6.10.4.1 Timestamp

As part of tamper detection the RTC can capture the counter value (COUNT/CLOCK) into the TIMESTAMP register. Three CLK_RTC periods are required to detect the tampering condition and capture the value. The TIMESTAMP value can be read once the Tamper flag in the Interrupt Flag register (INTFLAG.TAMPER) is set. If the DMA Enable bit in the Control B register (CTRLB.DMAEN) is '1', a DMA request will be triggered by the timestamp. In order to determine which tamper source caused a capture, the Tamper ID register (TAMPID) provides the detection status of each tamper channel and the tamper input event. A DMA transfer can then read both TIMESTAMP and TAMPID in succession.

A new timestamp value cannot be captured as long as the previous value is not read. The Tamper flag is cleared by writing a '1' to INTFLAG.TAMPER. If several tamper conditions occur in a short window before the flag is cleared, only the first timestamp may be logged. However, the detection of each tamper will still be recorded in TAMPID.

The Tamper Input Event (TAMPEVT) will always perform a timestamp capture. To capture on the external inputs (INn), the corresponding Input Action field in the Tamper Control register (TAMPCTRL.INnACT) must be written to '1'. If an input is set for wake functionality it does not capture the timestamp; however the Tamper flag and TAMPID will still be updated.

Note: the TIMESTAMP value should be read once, and INTFLAG.TAMPER must be cleared. The next value should be read only after the INTFLAG.TAMPER is set again.

25.6.10.4.2 Active Tamper Detection

The RTC provides a mean of detecting broken traces on the PCB , also known as Active Tamper Detection. In this mode, a generated internal RTC signal can be directly routed over critical components on the board using 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 Tamper Detection mode and the generation of the RTC signal is enabled by setting the RTCOUT bit in the Control B register (CTRLB.RTCOUT).

Enabling Active Tamper Detection requires the following steps:

- Enable the RTC prescaler output by writing a one 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 Tamper Detection Frequency field in the Control B register (CTRLB.ACTF)

  $$\mathrm{GCLK\_RTC\_OUT} = \frac{\mathrm{CLK\_RTC}}{2^{\mathrm{CTRLB.ACTF}+1}}$$

- Enable the tamper input n (INn) in Active Tamper Detection by writing 3 to the corresponding Input Action field in the Tamper Control register (TAMPCTRL.INnACT).
- Select Active Tamper Detection Monitoring Source (TrustRAM or INn/OUTn tamper pins) using ALSIn bit of TAMPCTRLB register
- Enable Active Tamper Detection by setting the CTRLB.RTCOUT bit

25.7 Register Summary - Mode 0 - 32-Bit Counter

This Register Description section is valid if the RTC is in COUNT32 mode (CTRLA.MODE=0).

Refer to the Registers Description section for more details on register properties and access permissions.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRLA	15:8	COUNTSYNC	GPTRST			PRESCALER[3:0]			
		7:0	MATCHCLR				MODE[1:0]		ENABLE	SWRST
0x02	CTRLB	15:8	SEPTO	ACTF[2:0]				DEBF[2:0]		
		7:0	DMAEN	RTCOUT	DEBASYNC	DEBMAJ				GP0EN
0x04	EVCTRL	31:24								PERDEO
		23:16								TAMPEVEI
		15:8	OVFEO	TAMPEREO						CMPEO0
		7:0	PEREO7	PEREO6	PEREO5	PEREO4	PEREO3	PEREO2	PEREO1	PEREO0
0x08	INTENCLR	15:8	OVF	TAMPER						CMP0
		7:0	PER7	PER6	PER5	PER4	PER3	PER2	PER1	PER0
0x0A	INTENSET	15:8	OVF	TAMPER						CMP0
		7:0	PER7	PER6	PER5	PER4	PER3	PER2	PER1	PER0
0x0C	INTFLAG	15:8	OVF	TAMPER						CMP0
		7:0	PER7	PER6	PER5	PER4	PER3	PER2	PER1	PER0
0x0E	DBGCTRL	7:0								DBGRUN
0x0F	Reserved									
0x10	SYNCBUSY	31:24								
		23:16							GP1	GP0
		15:8	COUNTSYNC							
		7:0			COMP0		COUNT	FREQCORR	ENABLE	SWRST
0x14	FREQCORR	7:0	SIGN	VALUE[6:0]						
0x15 ... 0x17	Reserved									
0x18	COUNT	31:24	COUNT[31:24]							
		23:16	COUNT[23:16]							
		15:8	COUNT[15:8]							
		7:0	COUNT[7:0]							
0x1C ... 0x1F	Reserved									
0x20	COMP0	31:24	COMP[31:24]							
		23:16	COMP[23:16]							
		15:8	COMP[15:8]							
		7:0	COMP[7:0]							
0x24 ... 0x3F	Reserved									
0x40	GP0	31:24	GP[31:24]							
		23:16	GP[23:16]							
		15:8	GP[15:8]							
		7:0	GP[7:0]							
0x44	GP1	31:24	GP[31:24]							
		23:16	GP[23:16]							
		15:8	GP[15:8]							
		7:0	GP[7:0]							
0x48 ... 0x5F	Reserved									
0x60	TAMPCTRL	31:24	DEBNC7	DEBNC6	DEBNC5	DEBNC4	DEBNC3	DEBNC2	DEBNC1	DEBNC0
		23:16	TAMLVL7	TAMLVL6	TAMLVL5	TAMLVL4	TAMLVL3	TAMLVL2	TAMLVL1	TAMLVL0
		15:8	IN7ACT[1:0]		IN6ACT[1:0]		IN5ACT[1:0]		IN4ACT[1:0]	
		7:0	IN3ACT[1:0]		IN2ACT[1:0]		IN1ACT[1:0]		IN0ACT[1:0]	

.....continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x64	TIMESTAMP	31:24	COUNT[31:24]							
		23:16	COUNT[23:16]							
		15:8	COUNT[15:8]							
		7:0	COUNT[7:0]							
0x68	TAMPID	31:24	TAMPEVT							
		23:16								
		15:8								
		7:0	TAMPID7	TAMPID6	TAMPID5	TAMPID4	TAMPID3	TAMPID2	TAMPID1	TAMPID0
0x6C	TAMPCTRLB	31:24								
		23:16								
		15:8								
		7:0	ALSI7	ALSI6	ALSI5	ALSI4	ALSI3	ALSI2	ALSI1	ALSI0

25.7.1 Control A in COUNT32 mode (CTRLA.MODE=0)

Name: CTRLA
Offset: 0x00
Reset: 0x0000
Property: PAC Write-Protection, Enable-Protected Bits, Write-Synchronized Bits

This register must be written with 32-bit accesses only.

Bit	15	14	13	12	11	10	9	8
	COUNTSYNC	GPTRST			PRESCALER[3:0]			
Access	R/W	R/W			R/W	R/W	R/W	R/W
Reset	0	0			0	0	0	0

Bit	7	6	5	4	3	2	1	0
	MATCHCLR				MODE[1:0]		ENABLE	SWRST
Access	R/W				R/W	R/W	R/W	R/W
Reset	0				0	0	0	0

Bit 15 – COUNTSYNC COUNT Read Synchronization Enable
The COUNT register requires synchronization when reading. Disabling the synchronization will prevent reading valid values from the COUNT register.
Note: This bit is write-synchronized: SYNCBUSY.COUNTSYNC must be checked to ensure the CTRLA.COUNTSYNC synchronization is complete.

Note: 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.
Note: This bit is enable-protected. This bit is not synchronized.

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.
Note: This bit field is enable-protected. This bit field is not synchronized.

Value	Name	Description
0x0	OFF	CLK_RTC_CNT = GCLK_RTC/1
0x1	DIV1	CLK_RTC_CNT = GCLK_RTC/1
0x2	DIV2	CLK_RTC_CNT = GCLK_RTC/2
0x3	DIV4	CLK_RTC_CNT = GCLK_RTC/4
0x4	DIV8	CLK_RTC_CNT = GCLK_RTC/8
0x5	DIV16	CLK_RTC_CNT = GCLK_RTC/16
0x6	DIV32	CLK_RTC_CNT = GCLK_RTC/32
0x7	DIV64	CLK_RTC_CNT = GCLK_RTC/64
0x8	DIV128	CLK_RTC_CNT = GCLK_RTC/128
0x9	DIV256	CLK_RTC_CNT = GCLK_RTC/256
0xA	DIV512	CLK_RTC_CNT = GCLK_RTC/512
0xB	DIV1024	CLK_RTC_CNT = GCLK_RTC/1024
0xC-0xF	-	Reserved

Bit 7 – MATCHCLR Clear on Match
This bit defines if the counter is cleared or not on a match.
Note: This bit is enable-protected. This bit is not synchronized.

Value	Description
0	The counter is not cleared on a Compare match
1	The counter is cleared on a Compare match

Bits 3:2 – MODE[1:0] Operating Mode

This bit group defines the operating mode of the RTC.

Note: This bit is enable-protected. 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

Notes:

1. This bit is write-synchronized: SYNCBUSY.ENABLE must be checked to ensure the CTRLA.ENABLE synchronization is complete.
2. 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 RTC (except DBGCTRL) to their initial state, and the RTC will be disabled.

Writing a '1' to CTRLA.SWRST will always take precedence, meaning that all other writes in the same write-operation will be discarded.

Notes:

1. When the CTRLA.SWRST is written, the user must poll SYNCBUSY.SWRST bit to know when the reset operation is complete.
2. During a SWRST, access to registers/bits without SWRST are disallowed until SYNCBUSY.SWRST is cleared by hardware.
3. This bit is not enable protected.

Value	Description
0	There is not reset operation ongoing
1	The reset operation is ongoing

25.7.2 Control B in COUNT32 mode (CTRLA.MODE=0)

Name: CTRLB
Offset: 0x02
Reset: 0x0000
Property: PAC Write-Protection, Enable-Protected

Bit	15	14	13	12	11	10	9	8
	SEPTO	ACTF[2:0]				DEBF[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
	DMAEN	RTCOUT	DEBASYNC	DEBMAJ				GP0EN
Access	R/W	R/W	R/W	R/W				R/W
Reset	0	0	0	0				0

Bit 15 – SEPTO Separate Tamper Outputs

Value	Description
0	IN[n] is compared to OUT[0].
1	IN[n] is compared to OUT[n].

Bits 14:12 – ACTF[2:0] Active Tamper Detection Frequency
These bits define the prescaling factor for the RTC clock output (OUT) used during Active Tamper Detection 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.
1	Tamper DMA request is enabled.

Bit 6 – RTCOUT RTC Output Enable

Value	Description
0	The RTC Active Tamper Detection is disabled.
1	The RTC Active Tamper Detection 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	Majority vote is not enabled.
1	Majority vote (two values out of three) is 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.

25.7.3 Event Control in COUNT32 mode (CTRLA.MODE=0)

Name: EVCTRL
Offset: 0x04
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
								PERDEO
Access								R/W
Reset								0

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						CMPEO0
Access	R/W	R/W						R/W
Reset	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 24 – PERDEO Periodic Interval Daily Event Output Enable

Value	Description
0	Periodic Daily event is disabled and will not be generated.
1	Periodic Daily event is enabled and will be generated. The event occurs at the overflow of the RTC counter (i.e., when the RTC counter goes from 0xFFFF to 0x0000).

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.

Bit 8 – CMPEO0 Compare Event Output Enable

Value	Description
0	Compare event is disabled and will not be generated.
1	Compare 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.

25.7.4 Interrupt Enable Clear in COUNT32 mode (CTRLA.MODE=0)

Name: INTENCLR
Offset: 0x08
Reset: 0x0000
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	15	14	13	12	11	10	9	8
	OVF	TAMPER						CMP0
Access	R/W	R/W						R/W
Reset	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 Disable
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 Disable
Writing a '0' to this bit has no effect.
Writing a '1' to this but 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.

Bit 8 – CMP0 Compare Interrupt Disable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Compare Interrupt Enable bit, which disables the Compare interrupt.

Value	Description
0	The Compare interrupt is disabled.
1	The Compare interrupt is enabled.

Bits 0, 1, 2, 3, 4, 5, 6, 7 – PERn Periodic Interval n Interrupt Disable [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.

25.7.5 Interrupt Enable Set in COUNT32 mode (CTRLA.MODE=0)

Name: INTENSET
Offset: 0x0A
Reset: 0x0000
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	15	14	13	12	11	10	9	8
	OVF	TAMPER						CMP0
Access	R/W	R/W						R/W
Reset	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.

Bit 8 – CMP0 Compare Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Compare Interrupt Enable bit, which enables the Compare interrupt.

Value	Description
0	The Compare interrupt is disabled.
1	The Compare 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.

25.7.6 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						CMP0
Access	R/W	R/W						R/W
Reset	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 will be generated if INTENSET.OVF is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Overflow interrupt flag.

Bit 14 – TAMPER Tamper event
This flag is set after a tamper condition occurs, and an interrupt request will be generated if INTENSET.TAMPER is '1'. Writing a '0' to this bit has no effect. Writing a '1' to this bit clears the Tamper interrupt flag.

Bit 8 – CMP0 Compare
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 will be generated if INTENSET.COMP0 is one.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Compare interrupt flag.

Bits 0, 1, 2, 3, 4, 5, 6, 7 – PERn Periodic Interval n [n = 7..0]
This flag is cleared by writing a '1' to the flag.
This flag is set on the 0-to-1 transition of prescaler bit [n+2], and an interrupt request will be generated if INTENSET.PERn is one.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Periodic Interval n interrupt flag.

25.7.7 Debug Control

Name: DBGCTRL
Offset: 0x0E
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 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.

25.7.8 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
							GP1	GP0
Access							R	R
Reset							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
			COMP0		COUNT	FREQCORR	ENABLE	SWRST
Access			R		R	R	R	R
Reset			0		0	0	0	0

Bits 16, 17 – GPn General Purpose n Synchronization Busy Status [n = 1..0]

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.

Bit 5 – COMP0 Compare Synchronization Busy Status

Value	Description
0	Write synchronization for COMP0 register is complete.
1	Write synchronization for COMP0 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.

Value	Description
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.

25.7.9 Frequency Correction

Name: FREQCORR
Offset: 0x14
Reset: 0x00
Property: PAC Write-Protection, 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
0x0	The correction value is positive, i.e., frequency will be decreased.
0x1	The correction value is negative, i.e., frequency will be increased.

Bits 6:0 – VALUE[6:0] Correction Value

These bits define the amount of correction applied to the RTC prescaler.

Value	Description
0x0	Correction is disabled and the RTC frequency is unchanged.
0x1 – 0x7F	The RTC frequency is adjusted according to the value.

25.7.10 Counter Value in COUNT32 mode (CTRLA.MODE=0)

Name: COUNT
Offset: 0x18
Reset: 0x00000000
Property: PAC Write-Protection, Write-Synchronized

This register must be written with 32-bit accesses only.

Note: Prior to any read access, this register must be synchronized by the user by writing CTRLA.COUNTSYNC=1.

Note: 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.

25.7.11 Compare Value in COUNT32 mode (CTRLA.MODE=0)

Name: COMP0
Offset: 0x20
Reset: 0x00000000
Property: PAC Write-Protection, Write-Synchronized

Note: This register is write-synchronized: SYNCBUSY.COMP0 must be checked to ensure the COMP0 register synchronization is complete.

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 COMP is continuously compared with the 32-bit COUNT value. When a match occurs, the Compare interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.CMP0) is set on the next counter cycle, and the counter value is cleared if CTRLA.MATCHCLR is '1'.

25.7.12 General Purpose n

Name: GP
Offset: 0x40 + n*0x04 [n=0..1]
Reset: 0x00000000
Property: PAC Write-Protection, Write-Synchronized

Note: This register is write-synchronized: SYNCBUSY.GPn must be checked to ensure the GPn register synchronization is complete.

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 25.6.10.3. General Purpose Registers.

25.7.13 Tamper Control

Name: TAMPCTRL
Offset: 0x60
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
	DEBNC7	DEBNC6	DEBNC5	DEBNC4	DEBNC3	DEBNC2	DEBNC1	DEBNC0
Access								
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	TAMLVL7	TAMLVL6	TAMLVL5	TAMLVL4	TAMLVL3	TAMLVL2	TAMLVL1	TAMLVL0
Access								
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	IN7ACT[1:0]		IN6ACT[1:0]		IN5ACT[1:0]		IN4ACT[1:0]	
Access								
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	IN3ACT[1:0]		IN2ACT[1:0]		IN1ACT[1:0]		IN0ACT[1:0]	
Access								
Reset	0	0	0	0	0	0	0	0

Bits 24, 25, 26, 27, 28, 29, 30, 31 – DEBNCn Debounce Enable of Tamper Input INn [n=0..7]
Note: Debounce feature does not apply to the Active Tamper Detection 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, 20, 21, 22, 23 – TAMLVLn Tamper Level Select of Tamper Input INn [n=0..7]
Note: Tamper Level feature does not apply to the Active Tamper Detection mode (TAMPCTRL.INACT = ACTL).

Value	Description
0	A falling edge condition will be detected on Tamper input INn.
1	A rising edge condition will be detected on Tamper input INn.

Bits 0:1, 2:3, 4:5, 6:7, 8:9, 10:11, 12:13, 14:15 – INnACT Tamper Channel n Action [n=0..7]
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 OUTn pins . When a mismatch occurs, capture timestamp and set Tamper flag

25.7.14 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

25.7.15 Tamper ID

Name: TAMPID
Offset: 0x68
Reset: 0x00000000
Property: -

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
	TAMPID7	TAMPID6	TAMPID5	TAMPID4	TAMPID3	TAMPID2	TAMPID1	TAMPID0
Access	R/W	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 – 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, 4, 5, 6, 7 – TAMPIDn Tamper Channel n Detected [n=0..7]

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

25.7.16 Tamper Control B

Name: TAMPCTRLB
Offset: 0x6C
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
	ALSI7	ALSI6	ALSI5	ALSI4	ALSI3	ALSI2	ALSI1	ALSI0
Access	R/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 – ALSIn Active Tamper Detection Internal Select n [n=0..7]
Note: Only one ALSI bit must be set to enable Active Tamper Detection on the TrustRAM.

Value	Description
0	Active Tamper Detection is monitoring the RTC signal using INn and OUTn tamper pins
1	Active Tamper Detection is monitoring the RTC signal on the TrustRAM Active shield

25.8 Register Summary - Mode 1 - 16-Bit Counter

This Register Description section is valid if the RTC is in COUNT16 mode (CTRLA.MODE=1).

Refer to the Registers Description section for more details on register properties and access permissions.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRLA	15:8	COUNTSYNC	GPTRST			PRESCALER[3:0]			
		7:0					MODE[1:0]		ENABLE	SWRST
0x02	CTRLB	15:8	SEPTO	ACTF[2:0]				DEBF[2:0]		
		7:0	DMAEN	RTCOUT	DEBASYNC	DEBMAJ				GP0EN
0x04	EVCTRL	31:24								PERDEO
		23:16								TAMPEVEI
		15:8	OVFEO	TAMPEREO					CMPEO1	CMPEO0
		7:0	PEREO7	PEREO6	PEREO5	PEREO4	PEREO3	PEREO2	PEREO1	PEREO0
0x08	INTENCLR	15:8	OVF	TAMPER					CMP1	CMP0
		7:0	PER7	PER6	PER5	PER4	PER3	PER2	PER1	PER0
0x0A	INTENSET	15:8	OVF	TAMPER					CMP1	CMP0
		7:0	PER7	PER6	PER5	PER4	PER3	PER2	PER1	PER0
0x0C	INTFLAG	15:8	OVF	TAMPER					CMP1	CMP0
		7:0	PER7	PER6	PER5	PER4	PER3	PER2	PER1	PER0
0x0E	DBGCTRL	7:0								DBGRUN
0x0F	Reserved									
0x10	SYNCBUSY	31:24								
		23:16							GP1	GP0
		15:8	COUNTSYNC							
		7:0		COMP1	COMP0	PER	COUNT	FREQCORR	ENABLE	SWRST
0x14	FREQCORR	7:0	SIGN	VALUE[6:0]						
0x15 ... 0x17	Reserved									
0x18	COUNT	15:8	COUNT[15:8]							
		7:0	COUNT[7:0]							
0x1A ... 0x1B	Reserved									
0x1C	PER	15:8	PER[15:8]							
		7:0	PER[7:0]							
0x1E ... 0x1F	Reserved									
0x20	COMP0	15:8	COMP[15:8]							
		7:0	COMP[7:0]							
0x22	COMP1	15:8	COMP[15:8]							
		7:0	COMP[7:0]							
0x24 ... 0x3F	Reserved									
0x40	GP0	31:24	GP[31:24]							
		23:16	GP[23:16]							
		15:8	GP[15:8]							
		7:0	GP[7:0]							
0x44	GP1	31:24	GP[31:24]							
		23:16	GP[23:16]							
		15:8	GP[15:8]							
		7:0	GP[7:0]							
0x48 ... 0x5F	Reserved									

.....continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x60	TAMPCTRL	31:24	DEBNC7	DEBNC6	DEBNC5	DEBNC4	DEBNC3	DEBNC2	DEBNC1	DEBNC0
		23:16	TAMLVL7	TAMLVL6	TAMLVL5	TAMLVL4	TAMLVL3	TAMLVL2	TAMLVL1	TAMLVL0
		15:8	IN7ACT[1:0]		IN6ACT[1:0]		IN5ACT[1:0]		IN4ACT[1:0]	
		7:0	IN3ACT[1:0]		IN2ACT[1:0]		IN1ACT[1:0]		IN0ACT[1:0]	
0x64	TIMESTAMP	31:24								
		23:16								
		15:8	COUNT[15:8]							
		7:0	COUNT[7:0]							
0x68	TAMPID	31:24	TAMPEVT							
		23:16								
		15:8								
		7:0	TAMPID7	TAMPID6	TAMPID5	TAMPID4	TAMPID3	TAMPID2	TAMPID1	TAMPID0
0x6C	TAMPCTRLB	31:24								
		23:16								
		15:8								
		7:0	ALSI7	ALSI6	ALSI5	ALSI4	ALSI3	ALSI2	ALSI1	ALSI0

25.8.1 Control A in COUNT16 mode (CTRLA.MODE=1)

Name: CTRLA
Offset: 0x00
Reset: 0x0000
Property: PAC Write-Protection, Enable-Protected Bits, Write-Synchronized Bits

This register must be written with 16-bit accesses only.

Bit	15	14	13	12	11	10	9	8
	COUNTSYNC	GPTRST			PRESCALER[3:0]			
Access	R/W	R/W			R/W	R/W	R/W	R/W
Reset	0	0			0	0	0	0

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

Bit 15 – COUNTSYNC COUNT Read Synchronization Enable
The COUNT register requires synchronization when reading. Disabling the synchronization will prevent reading valid values from the COUNT register.
Note: This bit is write-synchronized: SYNCBUSY.COUNTSYNC must be checked to ensure the CTRLA.COUNTSYNC synchronization is complete.

Note: 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.
Note: This bit is enable-protected. 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.

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.
Note: This bit field is enable-protected. This bit field is not synchronized.

Value	Name	Description
0x0	OFF	CLK_RTC_CNT = GCLK_RTC/1
0x1	DIV1	CLK_RTC_CNT = GCLK_RTC/1
0x2	DIV2	CLK_RTC_CNT = GCLK_RTC/2
0x3	DIV4	CLK_RTC_CNT = GCLK_RTC/4
0x4	DIV8	CLK_RTC_CNT = GCLK_RTC/8
0x5	DIV16	CLK_RTC_CNT = GCLK_RTC/16
0x6	DIV32	CLK_RTC_CNT = GCLK_RTC/32
0x7	DIV64	CLK_RTC_CNT = GCLK_RTC/64
0x8	DIV128	CLK_RTC_CNT = GCLK_RTC/128
0x9	DIV256	CLK_RTC_CNT = GCLK_RTC/256
0xA	DIV512	CLK_RTC_CNT = GCLK_RTC/512
0xB	DIV1024	CLK_RTC_CNT = GCLK_RTC/1024
0xC-0xF	-	Reserved

Bits 3:2 – MODE[1:0] Operating Mode

This field defines the operating mode of the RTC.

Note: This bit is enable-protected. 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

Notes:

1. This bit is write-synchronized: SYNCBUSY.ENABLE must be checked to ensure the CTRLA.ENABLE synchronization is complete.
2. 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 RTC (except DBGCTRL) to their initial state, and the RTC will be disabled.

Writing a '1' to CTRLA.SWRST will always take precedence, meaning that all other writes in the same write-operation will be discarded.

Notes:

1. When the CTRLA.SWRST is written, the user must poll SYNCBUSY.SWRST bit to know when the reset operation is complete.
2. During a SWRST, access to registers/bits without SWRST are disallowed until SYNCBUSY.SWRST is cleared by hardware.
3. This bit is not enable protected.

Value	Description
0	There is not reset operation ongoing
1	The reset operation is ongoing

25.8.2 Control B in COUNT16 mode (CTRLA.MODE=1)

Name: CTRLB
Offset: 0x02
Reset: 0x0000
Property: PAC Write-Protection, Enable-Protected

Bit	15	14	13	12	11	10	9	8
	SEPTO	ACTF[2:0]				DEBF[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
	DMAEN	RTCOUT	DEBASYNC	DEBMAJ				GP0EN
Access	R/W	R/W	R/W	R/W				R/W
Reset	0	0	0	0				0

Bit 15 – SEPTO Separate Tamper Outputs

Value	Description
0	IN[n] is compared to OUT[0] (backward-compatible).
1	IN[n] is compared to OUT[n].

Bits 14:12 – ACTF[2:0] Active Tamper Detection Frequency
These bits define the prescaling factor for the RTC clock output (OUT) used during Active Tamper Detection 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.
1	Tamper DMA request is enabled.

Bit 6 – RTCOUT RTC Output Enable

Value	Description
0	The RTC Active Tamper Detection output is disabled.
1	The RTC Active Tamper Detection 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	Majority vote is not enabled.
1	Majority vote (two values out of three) is 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 COUNT16 mode (CTRLA.MODE=1)

Name: EVCTRL
Offset: 0x04
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
								PERDEO
Access								R/W
Reset								0

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					CMPEO1	CMPEO0
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 24 – PERDEO Periodic Interval Daily Event Output Enable

Value	Description
0	Periodic Daily event is disabled and will not be generated.
1	Periodic Daily event is enabled and will be generated. The event occurs at the overflow of the RTC counter (i.e., when the RTC counter goes from 0xFFFF to 0x0000).

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 8, 9 – CMPEOn 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 COUNT16 mode (CTRLA.MODE=1)

Name: INTENCLR
Offset: 0x08
Reset: 0x0000
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	15	14	13	12	11	10	9	8
	OVF	TAMPER					CMP1	CMP0
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

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 – CMPn Compare n Interrupt Enable [n = 1..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.

25.8.5 Interrupt Enable Set in COUNT16 mode (CTRLA.MODE=1)

Name: INTENSET
Offset: 0x0A
Reset: 0x0000
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	15	14	13	12	11	10	9	8
	OVF	TAMPER					CMP1	CMP0
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 is disabled.
1	The Tamper 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 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.

25.8.6 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					CMP1	CMP0
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
This flag is cleared by writing a '1' to the flag.
This flag is set on the next CLK_RTC_CNT cycle after an overflow condition occurs, and an interrupt request will be generated if INTENSET.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 will be generated if INTENSET.TAMPER is one.
Writing a '0' to this bit has no effect.
Writing a one to this bit clears the Tamper 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 a match with the compare condition, and an interrupt request will be generated if INTENSET.COMPn is one.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Compare n interrupt flag.

Bits 0, 1, 2, 3, 4, 5, 6, 7 – PERn Periodic Interval n [n = 7..0]
This flag is cleared by writing a '1' to the flag.
This flag is set on the 0-to-1 transition of prescaler bit [n+2], and an interrupt request will be generated if INTENSET.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.

25.8.7 Debug Control

Name: DBGCTRL
Offset: 0x0E
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 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 COUNT16 mode (CTRLA.MODE=1)

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
							GP1	GP0
Access							R	R
Reset							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	PER	COUNT	FREQCORR	ENABLE	SWRST
Access		R/W	R/W	R	R	R	R	R
Reset		0	0	0	0	0	0	0

Bits 16, 17 – GPn General Purpose n Synchronization Busy Status [n = 1..0]

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 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

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: PAC Write-Protection, 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
0x0	The correction value is positive, i.e., frequency will be decreased.
0x1	The correction value is negative, i.e., frequency will be increased.

Bits 6:0 – VALUE[6:0] Correction Value

These bits define the amount of correction applied to the RTC prescaler.

Value	Description
0x0	Correction is disabled and the RTC frequency is unchanged.
0x1 – 0x7F	The RTC frequency is adjusted according to the value.

25.8.10 Counter Value in COUNT16 mode (CTRLA.MODE=1)

Name: COUNT
Offset: 0x18
Reset: 0x0000
Property: PAC Write-Protection, Write-Synchronized

This register must be written with 16-bit accesses only.

Notes:

1. Prior to any read access, this register must be synchronized by the user by writing CTRLA.COUNTSYNC=1.
2. 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).

25.8.11 Counter Period in COUNT16 mode (CTRLA.MODE=1)

Name: PER
Offset: 0x1C
Reset: 0x0000
Property: PAC Write-Protection, 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).

25.8.12 Compare n Value in COUNT16 mode (CTRLA.MODE=1)

Name: COMP
Offset: 0x20 + n*0x02 [n=0..1]
Reset: 0x0000
Property: PAC Write-Protection, Write-Synchronized

Note: This register is write-synchronized: SYNCBUSY.COMPn must be checked to ensure the COMPn register synchronization is complete.

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.

25.8.13 General Purpose n

Name: GP
Offset: 0x40 + n*0x04 [n=0..1]
Reset: 0x00000000
Property: PAC Write-Protection, Write-Synchronized

Note: This register is write-synchronized: SYNCBUSY.GPn must be checked to ensure the GPn register synchronization is complete.

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 25.6.10.3. General Purpose Registers.

25.8.14 Tamper Control

Name: TAMPCTRL
Offset: 0x60
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
	DEBNC7	DEBNC6	DEBNC5	DEBNC4	DEBNC3	DEBNC2	DEBNC1	DEBNC0
Access								
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	TAMLVL7	TAMLVL6	TAMLVL5	TAMLVL4	TAMLVL3	TAMLVL2	TAMLVL1	TAMLVL0
Access								
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	IN7ACT[1:0]		IN6ACT[1:0]		IN5ACT[1:0]		IN4ACT[1:0]	
Access								
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	IN3ACT[1:0]		IN2ACT[1:0]		IN1ACT[1:0]		IN0ACT[1:0]	
Access								
Reset	0	0	0	0	0	0	0	0

Bits 24, 25, 26, 27, 28, 29, 30, 31 – DEBNCn Debounce Enable of Tamper Input INn [n=0..7]
Note: Debounce feature does not apply to the Active Tamper Detection 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, 20, 21, 22, 23 – TAMLVLn Tamper Level Select of Tamper Input INn [n=0..7]
Note: Tamper Level feature does not apply to the Active Tamper Detection mode (TAMPCTRL.INACT = ACTL).

Value	Description
0	A falling edge condition will be detected on Tamper input INn.
1	A rising edge condition will be detected on Tamper input INn.

Bits 0:1, 2:3, 4:5, 6:7, 8:9, 10:11, 12:13, 14:15 – INnACT Tamper Channel n Action [n=0..7]
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 OUTn pins . When a mismatch occurs, capture timestamp and set Tamper flag

25.8.15 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.

25.8.16 Tamper ID

Name: TAMPID
Offset: 0x68
Reset: 0x00000000
Property: -

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
	TAMPID7	TAMPID6	TAMPID5	TAMPID4	TAMPID3	TAMPID2	TAMPID1	TAMPID0
Access	R/W	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 – 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, 4, 5, 6, 7 – TAMPIDn Tamper Channel n Detected [n=0..7]

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 Tamper Control B

Name: TAMPCTRLB
Offset: 0x6C
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
	ALSI7	ALSI6	ALSI5	ALSI4	ALSI3	ALSI2	ALSI1	ALSI0
Access	R/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 – ALSIn Active Tamper Detection Internal Select n [n=0..7]
Note: Only one ALSI bit must be set to enable Active Tamper Detection on the TrustRAM.

Value	Description
0	Active Tamper Detection is monitoring the RTC signal using INn and OUTn tamper pins
1	Active Tamper Detection is monitoring the RTC signal on the TrustRAM Active shield

25.9 Register Summary - Mode 2 - Clock/Calendar

This Register Description section is valid if the RTC is in Clock/Calendar mode (CTRLA.MODE=2).

Refer to the Registers Description section for more details on register properties and access permissions.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRLA	15:8	CLOCKSYNC	GPTRST			PRESCALER[3:0]			
		7:0	MATCHCLR	CLKREP			MODE[1:0]		ENABLE	SWRST
0x02	CTRLB	15:8	SEPTO	ACTF[2:0]				DEBF[2:0]		
		7:0	DMAEN	RTCOUT	DEBASYNC	DEBMAJ				GP0EN
0x04	EVCTRL	31:24								PERDEO
		23:16								TAMPEVEI
		15:8	OVFEO	TAMPEREO						ALARMEO0
		7:0	PEREO7	PEREO6	PEREO5	PEREO4	PEREO3	PEREO2	PEREO1	PEREO0
0x08	INTENCLR	15:8	OVF	TAMPER						ALARM0
		7:0	PER7	PER6	PER5	PER4	PER3	PER2	PER1	PER0
0x0A	INTENSET	15:8	OVF	TAMPER						ALARM0
		7:0	PER7	PER6	PER5	PER4	PER3	PER2	PER1	PER0
0x0C	INTFLAG	15:8	OVF	TAMPER						ALARM0
		7:0	PER7	PER6	PER5	PER4	PER3	PER2	PER1	PER0
0x0E	DBGCTRL	7:0								DBGRUN
0x0F	Reserved									
0x10	SYNCBUSY	31:24								
		23:16							GP1	GP0
		15:8	CLOCKSYNC				MASK0			
		7:0			ALARM0		CLOCK	FREQCORR	ENABLE	SWRST
0x14	FREQCORR	7:0	SIGN	VALUE[6:0]						
0x15 ... 0x17	Reserved									
0x18	CLOCK	31:24	YEAR[5:0]						MONTH[3:2]	
		23:16	MONTH[1:0]		DAY[4:0]					HOUR[4]
		15:8	HOUR[3:0]				MINUTE[5:2]			
		7:0	MINUTE[1:0]		SECOND[5:0]					
0x1C ... 0x1F	Reserved									
0x20	ALARM0	31:24	YEAR[5:0]						MONTH[3:2]	
		23:16	MONTH[1:0]		DAY[4:0]					HOUR[4]
		15:8	HOUR[3:0]				MINUTE[5:2]			
		7:0	MINUTE[1:0]		SECOND[5:0]					
0x24	MASK0	7:0						SEL[2:0]		
0x25 ... 0x3F	Reserved									
0x40	GP0	31:24	GP[31:24]							
		23:16	GP[23:16]							
		15:8	GP[15:8]							
		7:0	GP[7:0]							
0x44	GP1	31:24	GP[31:24]							
		23:16	GP[23:16]							
		15:8	GP[15:8]							
		7:0	GP[7:0]							
0x48 ... 0x5F	Reserved									

.....continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x60	TAMPCTRL	31:24	DEBNC7	DEBNC6	DEBNC5	DEBNC4	DEBNC3	DEBNC2	DEBNC1	DEBNC0
		23:16	TAMLVL7	TAMLVL6	TAMLVL5	TAMLVL4	TAMLVL3	TAMLVL2	TAMLVL1	TAMLVL0
		15:8	IN7ACT[1:0]		IN6ACT[1:0]		IN5ACT[1:0]		IN4ACT[1:0]	
		7:0	IN3ACT[1:0]		IN2ACT[1:0]		IN1ACT[1:0]		IN0ACT[1:0]	
0x64	TIMESTAMP	31:24	YEAR[5:0]						MONTH[3:2]	
		23:16	MONTH[1:0]		DAY[4:0]					HOUR[4]
		15:8	HOUR[3:0]				MINUTE[5:2]			
		7:0	MINUTE[1:0]		SECOND[5:0]					
0x68	TAMPID	31:24	TAMPEVT							
		23:16								
		15:8								
		7:0	TAMPID7	TAMPID6	TAMPID5	TAMPID4	TAMPID3	TAMPID2	TAMPID1	TAMPID0
0x6C	TAMPCTRLB	31:24								
		23:16								
		15:8								
		7:0	ALSI7	ALSI6	ALSI5	ALSI4	ALSI3	ALSI2	ALSI1	ALSI0

25.9.1 Control A in Clock/Calendar mode (CTRLA.MODE=2)

Name: CTRLA
Offset: 0x00
Reset: 0x0000
Property: PAC Write-Protection, Enable-Protected Bits, Write-Synchronized Bits

This register must be written with 32-bit accesses only.

Bit	15	14	13	12	11	10	9	8
	CLOCKSYNC	GPTRST			PRESCALER[3:0]			
Access	R/W	R/W			R/W	R/W	R/W	R/W
Reset	0	0			0	0	0	0

Bit	7	6	5	4	3	2	1	0
	MATCHCLR	CLKREP			MODE[1:0]		ENABLE	SWRST
Access	R/W	R/W			R/W	R/W	R/W	R/W
Reset	0	0			0	0	0	0

Bit 15 – CLOCKSYNC CLOCK Read Synchronization Enable
The CLOCK register requires synchronization when reading. Disabling the synchronization will prevent reading valid values from the CLOCK register.
Note: This bit is not enable-protected

Note: This bit is write-synchronized: SYNCBUSY.CLOCKSYNC must be checked to ensure the CTRLA.CLOCKSYNC synchronization is complete.

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.
Note: This bit is enable-protected. This bit is not synchronized.

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.
Note: This bit field is enable-protected. This bit field is not synchronized.

Value	Name	Description
0x0	OFF	CLK_RTC_CNT = GCLK_RTC/1
0x1	DIV1	CLK_RTC_CNT = GCLK_RTC/1
0x2	DIV2	CLK_RTC_CNT = GCLK_RTC/2
0x3	DIV4	CLK_RTC_CNT = GCLK_RTC/4
0x4	DIV8	CLK_RTC_CNT = GCLK_RTC/8
0x5	DIV16	CLK_RTC_CNT = GCLK_RTC/16
0x6	DIV32	CLK_RTC_CNT = GCLK_RTC/32
0x7	DIV64	CLK_RTC_CNT = GCLK_RTC/64
0x8	DIV128	CLK_RTC_CNT = GCLK_RTC/128
0x9	DIV256	CLK_RTC_CNT = GCLK_RTC/256
0xA	DIV512	CLK_RTC_CNT = GCLK_RTC/512
0xB	DIV1024	CLK_RTC_CNT = GCLK_RTC/1024
0xC-0xF	-	Reserved

Bit 7 – MATCHCLR Clear on Match
This bit is valid only in Mode 0 (COUNT32) and Mode 2 (CLOCK). This bit can be written only when the peripheral is disabled.

Note: This bit is enable-protected. 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) and Alarm Value (ALARM0) registers. This bit can be written only when the peripheral is disabled.

Note: This bit is enable-protected. This bit is not synchronized.

Value	Description
0	24 Hour
1	12 Hour (AM/PM)

Bits 3:2 – MODE[1:0] Operating Mode

This bit group defines the operating mode of the RTC.

Note: This bit is enable-protected. 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

Notes:

1. This bit is write-synchronized: SYNCBUSY.ENABLE must be checked to ensure the CTRLA.ENABLE synchronization is complete.
2. 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 RTC, except DBGCTRL, to their initial state, and the RTC will be disabled.

Writing a '1' to CTRLA.SWRST will always take precedence, meaning that all other writes in the same write-operation will be discarded.

Notes:

1. When the CTRLA.SWRST is written, the user must poll SYNCBUSY.SWRST bit to know when the reset operation is complete.
2. During a SWRST, access to registers/bits without SWRST are disallowed until SYNCBUSY.SWRST is cleared by hardware.
3. This bit is not enable protected.

Value	Description
0	There is not reset operation ongoing
1	The reset operation is ongoing

25.9.2 Control B in Clock/Calendar mode (CTRLA.MODE=2)

Name: CTRLB
Offset: 0x2
Reset: 0x0000
Property: PAC Write-Protection, Enable-Protected

Bit	15	14	13	12	11	10	9	8
	SEPTO	ACTF[2:0]				DEBF[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
	DMAEN	RTCOUT	DEBASYNC	DEBMAJ				GP0EN
Access	R/W	R/W	R/W	R/W				R/W
Reset	0	0	0	0				0

Bit 15 – SEPTO Separate Tamper Outputs

Value	Description
0	IN[n] is compared tp OUT[0] (backward-compatible).
1	IN[n] is compared tp OUT[n].

Bits 14:12 – ACTF[2:0] Active Tamper Detection Frequency

These bits define the prescaling factor for the RTC clock output (OUT) used during Active Tamper Detection 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.
1	Tamper DMA request is enabled.

Bit 6 – RTCOUT RTC Out Enable

Value	Description
0	The RTC Active Tamper Detection is disabled.
1	The RTC Active Tamper Detection 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	Majority vote is not enabled.
1	Majority vote (two values out of three) is enabled.

Bit 0 – GP0EN General Purpose 0 Enable

Value	Description
0	COMP0 compare function enabled. GP0 disabled.
1	COMP0 compare function disabled. GP0 enabled.

25.9.3 Event Control in Clock/Calendar mode (CTRLA.MODE=2)

Name: EVCTRL
Offset: 0x04
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
								PERDEO
Access								R/W
Reset								0

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						ALARMEO0
Access	R/W	R/W						R/W
Reset	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 24 – PERDEO Periodic Interval Daily Event Output Enable

Value	Description
0	Periodic Daily event is disabled and will not be generated.
1	Periodic Daily event is enabled and will be generated. The event occurs at the last second of each day depending on the CTRLA.CLKREP bit: • If CLKREP = 0, the event will occur at 23:59:59 • If CLKREP = 1, the event will occur at 11:59:59, PM = 1

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.

Bit 8 – ALARMEO0 Alarm Event Output Enable

Value	Description
0	Alarm event is disabled and will not be generated.
1	Alarm 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.

25.9.4 Interrupt Enable Clear in Clock/Calendar mode (CTRLA.MODE=2)

Name: INTENCLR
Offset: 0x08
Reset: 0x0000
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	15	14	13	12	11	10	9	8
	OVF	TAMPER						ALARM0
Access	R/W	R/W						R/W
Reset	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 it disabled.
1	The Tamper interrupt is enabled.

Bit 8 – ALARM0 Alarm Interrupt Enable
Writing a '0' to this bit has no effect. Writing a '1' to this bit will clear the Alarm Interrupt Enable bit, which disables the Alarm interrupt.

Value	Description
0	The Alarm interrupt is disabled.
1	The Alarm 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.

25.9.5 Interrupt Enable Set in Clock/Calendar mode (CTRLA.MODE=2)

Name: INTENSET
Offset: 0x0A
Reset: 0x0000
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	15	14	13	12	11	10	9	8
	OVF	TAMPER						ALARM0
Access	R/W	R/W						R/W
Reset	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.

Bit 8 – ALARM0 Alarm Interrupt Enable
Writing a '0' to this bit has no effect. Writing a '1' to this bit will set the Alarm Interrupt Enable bit, which enables the Alarm interrupt.

Value	Description
0	The Alarm interrupt is disabled.
1	The Alarm 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.

25.9.6 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						ALARM0
Access	R/W	R/W						R/W
Reset	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 will be generated if INTENSET.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 will be generated if INTENSET.TAMPER is '1'. Writing a '0' to this bit has no effect. Writing a '1' to this bit clears the Tamper interrupt flag.

Bit 8 – ALARM0 Alarm
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 will be generated if INTENSET.ALARM0 is one.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Alarm interrupt flag.

Bits 0, 1, 2, 3, 4, 5, 6, 7 – PERn Periodic Interval n [n = 7..0]
This flag is cleared by writing a '1' to the flag.
This flag is set on the 0-to-1 transition of prescaler bit [n+2], and an interrupt request will be generated if INTENSET.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.

25.9.7 Debug Control

Name: DBGCTRL
Offset: 0x0E
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 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.

25.9.8 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
							GP1	GP0
Access							R	R
Reset							0	0

Bit	15	14	13	12	11	10	9	8
	CLOCKSYNC				MASK0			
Access	R				R			
Reset	0				0			

Bit	7	6	5	4	3	2	1	0
			ALARM0		CLOCK	FREQCORR	ENABLE	SWRST
Access			R		R	R	R	R
Reset			0		0	0	0	0

Bits 16, 17 – GPn General Purpose n Synchronization Busy Status [n = 1..0]

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.

Bit 11 – MASK0 Mask Synchronization Busy Status

Value	Description
0	Write synchronization for MASK0 register is complete.
1	Write synchronization for MASK0 register is ongoing.

Bit 5 – ALARM0 Alarm Synchronization Busy Status

Value	Description
0	Write synchronization for ALARM0 register is complete.
1	Write synchronization for ALARM0 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.

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

Value	Description
0	Write synchronization for CTRLA.SWRST bit is complete.
1	Write synchronization for CTRLA.SWRST bit is ongoing.

25.9.9 Frequency Correction

Name: FREQCORR
Offset: 0x14
Reset: 0x00
Property: PAC Write-Protection, 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
0x0	The correction value is positive, i.e., frequency will be decreased.
0x1	The correction value is negative, i.e., frequency will be increased.

Bits 6:0 – VALUE[6:0] Correction Value

These bits define the amount of correction applied to the RTC prescaler.

Value	Description
0x0	Correction is disabled and the RTC frequency is unchanged.
0x1 – 0x7F	The RTC frequency is adjusted according to the value.

25.9.10 Clock Value in Clock/Calendar mode (CTRLA.MODE=2)

Name: CLOCK
Offset: 0x18
Reset: 0x00000000
Property: PAC Write-Protection, Write-Synchronized

This register must be written with 32-bit accesses only.

Notes:

1. Prior to any read access, this register must be synchronized by the user by writing CTRLA.CLOCKSYNC=1.
2. This register is write-synchronized: SYNCBUSY.CLOCK must be checked to ensure the CLOCK register synchronization is complete.

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

25.9.11 Alarm Value in Clock/Calendar mode (CTRLA.MODE=2)

Name: ALARM0
Offset: 0x20
Reset: 0x00000000
Property: PAC Write-Protection, Write-Synchronized

The 32-bit value of ALARM0 is continuously compared with the 32-bit CLOCK value, based on the masking set by MASK0.SEL. When a match occurs, the Alarm interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.ALARM0) is set on the next counter cycle, and the counter is cleared if CTRLA.MATCHCLR is '1'.
Note: This register is write-synchronized: SYNCBUSY.ALARM0 must be checked to ensure the ALARM0 register synchronization is complete.

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 MASK0.SEL is 6

Bits 25:22 – MONTH[3:0] Month
The alarm month. Months are matched only if MASK0.SEL is greater than 4.

Bits 21:17 – DAY[4:0] Day
The alarm day. Days are matched only if MASK0.SEL is greater than 3.

Bits 16:12 – HOUR[4:0] Hour
The alarm hour. Hours are matched only if MASK0.SEL is greater than 2.
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
The alarm minute. Minutes are matched only if MASK0.SEL is greater than 1.

Bits 5:0 – SECOND[5:0] Second
The alarm second. Seconds are matched only if MASK0.SEL is greater than 0.

25.9.12 Alarm Mask in Clock/Calendar mode (CTRLA.MODE=2)

Name: MASK0
Offset: 0x24
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized

Note: This register is write-synchronized: SYNCBUSY.MASK0 must be checked to ensure the MASK0 register synchronization is complete.

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 ALARM0 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

25.9.13 General Purpose n

Name: GP
Offset: 0x40 + n*0x04 [n=0..1]
Reset: 0x00000000
Property: PAC Write-Protection, Write-Synchronized

Note: This register is write-synchronized: SYNCBUSY.GP must be checked to ensure the GP register synchronization is complete.

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 25.6.10.3. General Purpose Registers.

25.9.14 Tamper Control

Name: TAMPCTRL
Offset: 0x60
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
	DEBNC7	DEBNC6	DEBNC5	DEBNC4	DEBNC3	DEBNC2	DEBNC1	DEBNC0
Access								
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	TAMLVL7	TAMLVL6	TAMLVL5	TAMLVL4	TAMLVL3	TAMLVL2	TAMLVL1	TAMLVL0
Access								
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	IN7ACT[1:0]		IN6ACT[1:0]		IN5ACT[1:0]		IN4ACT[1:0]	
Access								
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	IN3ACT[1:0]		IN2ACT[1:0]		IN1ACT[1:0]		IN0ACT[1:0]	
Access								
Reset	0	0	0	0	0	0	0	0

Bits 24, 25, 26, 27, 28, 29, 30, 31 – DEBNCn Debounce Enable of Tamper Input INn [n=0..7]
Note: Debounce feature does not apply to the Active Tamper Detection 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, 20, 21, 22, 23 – TAMLVLn Tamper Level Select of Tamper Input INn [n=0..7]
Note: Tamper Level feature does not apply to the Active Tamper Detection mode (TAMPCTRL.INACT = ACTL).

Value	Description
0	A falling edge condition will be detected on Tamper input INn.
1	A rising edge condition will be detected on Tamper input INn.

Bits 0:1, 2:3, 4:5, 6:7, 8:9, 10:11, 12:13, 14:15 – INnACT Tamper Channel n Action [n=0..7]
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 OUTn pins . When a mismatch occurs, capture timestamp and set Tamper flag

25.9.15 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.

25.9.16 Tamper ID

Name: TAMPID
Offset: 0x68
Reset: 0x00000000
Property: -

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
	TAMPID7	TAMPID6	TAMPID5	TAMPID4	TAMPID3	TAMPID2	TAMPID1	TAMPID0
Access	R/W	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 – 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, 4, 5, 6, 7 – TAMPIDn Tamper Channel n Detected [n=0..7]

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

25.9.17 Tamper Control B

Name: TAMPCTRLB
Offset: 0x6C
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
	ALSI7	ALSI6	ALSI5	ALSI4	ALSI3	ALSI2	ALSI1	ALSI0
Access	R/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 – ALSIn Active Tamper Detection Internal Select n [n=0..7]
Note: Only one ALSI bit must be set to enable Active Tamper Detection on the TrustRAM.

Value	Description
0	Active Tamper Detection is monitoring the RTC signal using INn and OUTn tamper pins
1	Active Tamper Detection is monitoring the RTC signal on the TrustRAM Active shield

26. Frequency Meter (FREQM)

26.1 Overview

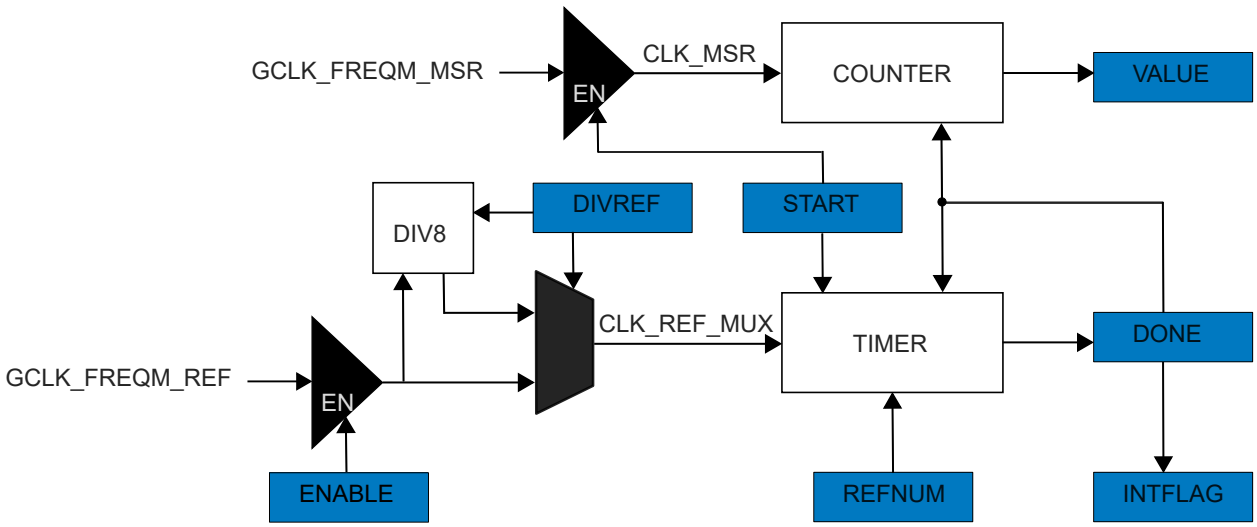
The Frequency Meter (FREQM) can be used to accurately measure the frequency of a clock by comparing it to a known reference clock.

26.2 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

26.3 Block Diagram

Figure 26-1. FREQM Block Diagram



26.4 Peripheral Dependencies

Table 26-1. FREQM Configuration Summary

Peripheral name	Base address	NVIC IRQ Index	MCLK AHB/APB Clocks	GCLK Peripheral Channel Index (GCLK.PCHCTRL)	PAC Peripheral Identifier Index (PAC.WRCTRL)	Power Domain (PM.STDBYCFG)
FREQM	0x40002C00	12: DONE	CLK_FREQM_APB	5: GCLK_FREQM_MSR 6: GCLK_FREQM_REF	11	PDSW

26.5 Functional Description

26.5.1 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/$f_{CLK\_REF\_MUX}$ 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

$f_{CLK\_MSR} = \left(\frac{VALUE + 1 + Error}{REFNUM + 1}\right) f_{CLK\_REF\_MUX}$,where Error represents the error introduced by the synchronization mechanism. The error can be maximum two measured clock cycles.

26.5.2 Basic Operation

26.5.2.1 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 for which the measurement is to be done in the Configuration A register (CFGA.REFNUM). This must be a non-zero number.

The following register is enable-protected, that is, 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.

26.5.2.2 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 will be reset to their initial state, and the FREQM will be disabled.

Then ENABLE and SWRST bits are write-synchronized.

26.5.2.3 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:

$f_{CLK\_MSR} = \left(\frac{VALUE + 1 + Error}{REFNUM + 1}\right) f_{CLK\_REF\_MUX}$

Notes:

1. In order to make sure the measurement result (VALUE.VALUE[23:0]) is valid, the overflow status (STATUS.OVF) should be checked.
2. Due to asynchronous operations, the VALUE Error measurement can be up to two samples.

In case 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.

26.5.3 I/O Lines

The GCLK I/O lines (GCLK_IO[7:0]) can be used as measurement or reference clock sources. This requires the I/O pins to be configured. For additional information, refer to 31. I/O Pin Controller (PORT).

26.5.4 Clocks

The clock for the FREQM bus interface (CLK_FREQM_APB) is enabled and disabled by the Main Clock Controller, the default state of CLK_FREQM_APB can be found in Peripheral Clock Masking.

Two generic clocks are used by the FREQM: Reference Clock (GCLK_FREQM_REF) and Measurement Clock (GCLK_FREQM_MSR).

GCLK_FREQM_REF is required to clock the internal reference timer, which acts as the frequency reference.

GCLK_FREQM_MSR is required to clock a ripple counter for frequency measurement. These clocks must be configured and enabled in the generic clock controller before using the FREQM.

26.5.5 Interrupts

The FREQM has one interrupt source:

- DONE: A frequency measurement is done.

The interrupt flag in the Interrupt Flag Status and Clear (26.6.6. 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 (26.6.5. INTENSET) register, and disabled by writing a '1' to the corresponding bit in the Interrupt Enable Clear (26.6.4. 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 FREQM is reset. See 26.6.6. INTFLAG for details on how to clear interrupt flags.

This interrupt is a synchronous wake-up source.

Note: Interrupts must be globally enabled for interrupt requests to be generated.

26.5.6 Sleep Mode Operation

The FREQM will continue to operate in idle sleep mode where the selected source clock is running. The FREQM's interrupts can be used to wake up the device from idle sleep mode.

For lowest chip power consumption in sleep modes, FREQM should be disabled before entering a sleep mode.

26.5.7 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.

26.5.8 Synchronization

Some registers (or bit fields within a register) require synchronization when read and/or written.

Synchronization is denoted by the "Read-Synchronized" (or "Read-Synchronized Bits") and/or "Write-Synchronized" (or "Write-Synchronized Bits") property in each individual register description.

For more details, refer to Register Synchronization.

26.6 Register Summary

Refer to the Registers Description section for more details on register properties and access permissions.

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	15:8	DIVREF							
		7:0	REFNUM[7:0]							
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	31:24								
		23:16								
		15:8								
		7:0							ENABLE	SWRST
0x10	VALUE	31:24								
		23:16	VALUE[23:16]							
		15:8	VALUE[15:8]							
		7:0	VALUE[7:0]							

26.6.1 Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized Bits

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

Bit 1 – ENABLE Enable

Note: This bit is write-synchronized: SYNCBUSY.ENABLE must be checked to ensure the CTRLA.ENABLE synchronization 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 FREQM to their initial state, and the FREQM will be disabled.
Writing a '1' to this bit will always take precedence, meaning that all other writes in the same write-operation will be discarded.

Notes:

1. When the CTRLA.SWRST is written, the user must poll SYNCBUSY.SWRST bit to know when the reset operation is complete.
2. During a SWRST, access to registers/bits without SWRST are disallowed until SYNCBUSY.SWRST is cleared by hardware.

Value	Description
0	There is no ongoing Reset operation.
1	The Reset operation is ongoing.

26.6.2 Control B

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.

26.6.3 Configuration A

Name: CFGA
Offset: 0x02
Reset: 0x0000
Property: PAC Write-Protection, Enable-protected

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

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

Bit 15 – DIVREF Divide Reference Clock
Divides the reference clock by 8

Value	Description
0	The reference clock is divided by 1.
1	The reference clock is divided by 8.

Bits 7:0 – REFNUM[7:0] Number of Reference Clock Cycles
Selects the duration of a measurement in number of CLK_FREQM_REF cycles. This must be a non-zero value, i.e. 0x01 (one cycle) to 0xFF (255 cycles).

26.6.4 Interrupt Enable Clear

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 will clear 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.

26.6.5 Interrupt Enable Set

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 will set 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.

26.6.6 Interrupt Flag Status and Clear

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 Measurement Done
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 will clear the DONE interrupt flag.

26.6.7 Status

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 will clear the OVF status.

Bit 0 – BUSY FREQM Status

Value	Description
0	No ongoing frequency measurement.
1	Frequency measurement is ongoing.

26.6.8 Synchronization Busy

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.

26.6.9 Value

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.

27. Direct Memory Access Controller (DMAC)

27.1 Overview

The Direct Memory Access Controller (DMAC) contains both a Direct Memory Access engine and a Cyclic Redundancy Check (CRC) engine. The DMAC can transfer data between memories and peripherals, and thus off-load these tasks from the CPU. It enables high data transfer rates with minimum CPU intervention, and frees up CPU time. With access to all peripherals, the DMAC can handle automatic transfer of data between communication modules.

The DMA part of the DMAC has several DMA channels which all can receive different types of transfer triggers to generate transfer requests from the DMA channels to the arbiter, see also the Block Diagram. The arbiter will grant one DMA channel at a time to act as the active channel. When an active channel has been granted, the fetch engine of the DMAC will fetch a transfer descriptor from the SRAM and store it in the internal memory of the active channel, which will execute the data transmission.

An ongoing data transfer of an active channel can be interrupted by a higher prioritized DMA channel. The DMAC will write back the updated transfer descriptor from the internal memory of the active channel to SRAM, and grant the higher prioritized channel to start transfer as the new active channel. Once a DMA channel is done with its transfer, interrupts and events can be generated optionally.

The DMAC has the following four bus interfaces:

- The *data transfer bus* is used for performing the actual DMA transfer.
- The *AHB/APB Bridge bus* is used when writing and reading the I/O registers of the DMAC.
- The *descriptor fetch bus* is used by the fetch engine to fetch transfer descriptors before data transfer can be started or continued.
- The *write-back bus* is used to write the transfer descriptor back to SRAM.

All buses are AHB host interfaces but the AHB/APB Bridge bus, which is an APB client interface.

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.

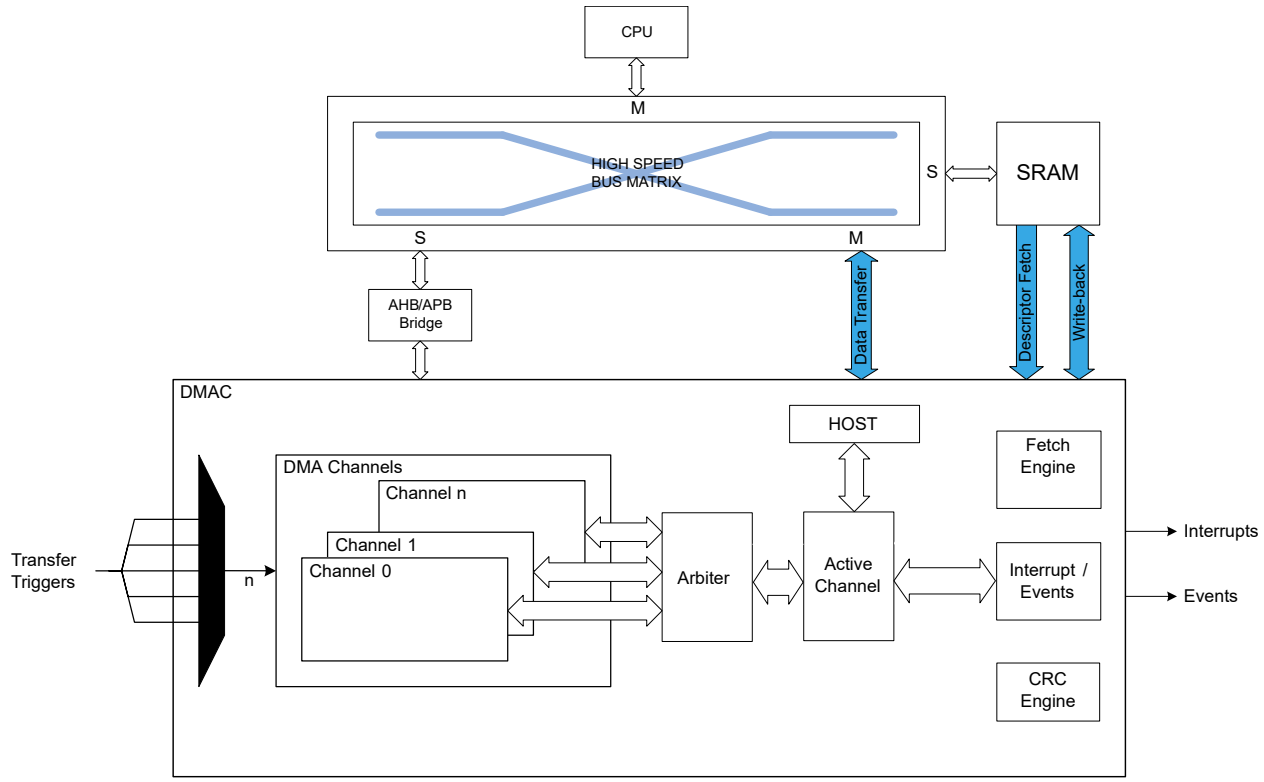
27.2 Features

- Data transfer from:
  - Peripheral to peripheral
  - Peripheral to memory
  - Memory to peripheral
  - Memory to memory
- Transfer trigger sources
  - Software
  - Events from Event System
  - Dedicated requests from peripherals
- SRAM based transfer descriptors
  - Single transfer using one descriptor
  - Multi-buffer or circular buffer modes by linking multiple descriptors
- Up to 16 channels
  - Enable 16 independent transfers
  - Automatic descriptor fetch for each channel
  - Suspend/resume operation support for each channel
- Flexible arbitration scheme
  - 4 configurable priority levels for each channel

  - Fixed or round-robin priority scheme within each priority level
- From 1 to 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
- 8 event inputs
  - One event input for each of the 8 least significant DMA channels
  - Can be selected to trigger normal transfers, periodic transfers or conditional transfers
  - Can be selected to suspend or resume channel operation
- 8 event outputs
  - One output event for each of the 8 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)

27.3 Block Diagram

Figure 27-1. DMAC Block Diagram



27.4 Peripheral Dependencies

Table 27-1. DMAC Configuration Summary

Peripheral name	Base address	NVIC IRQ Index	MCLK AHB/APB Clocks	PAC Peripheral Identifier Index (PAC.WRCTRL)	Events (EVSYS)		Power Domain (PM.STDBYCFG)
					Users (EVSYS.USER)	Generators (EVSYS.CHANNEL)	
DMAC	0x41006000	15: SUSP0, TERR0, TCMPL0 16: SUSP1, TERR1, TCMPL1 17: SUSP2, TERR2, TCMPL2 18: SUSP3, TERR3, TCMPL3 19: SUSP4-15, TERR4-15, TCMPL4-15	CLK_DMAC_AHB	35	7-14: CH0-7	33-40: CH0-7	PDSW

27.5 Functional Description

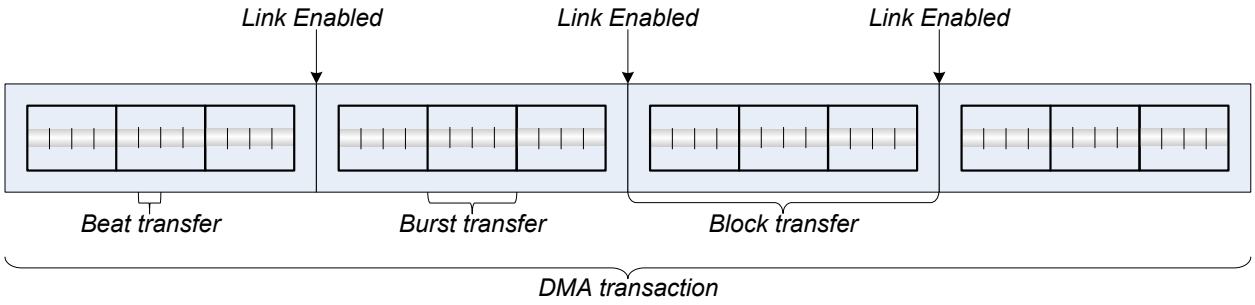
27.5.1 Principle of Operation

The DMAC consists of a DMA module and a CRC module.

27.5.1.1 DMA

The DMAC can transfer data between memories and peripherals without interaction from the CPU. The data transferred by the DMAC are called transactions, and these transactions can be split into smaller data transfers. The following figure shows the relationship between the different transfer sizes:

Figure 27-2. DMA Transfer Sizes



- Beat transfer: The size of one data transfer bus access, and the size is selected by writing the Beat Size bit group in the Block Transfer Control register (BTCTRL.BEATSIZE)
- Block transfer: The amount of data one transfer descriptor can transfer, and the amount can range from 1 to 64k beats. A block transfer can be interrupted.
- Transaction: The DMAC can link several transfer descriptors by having the first descriptor pointing to the second and so forth, as shown in the figure above. A DMA transaction is the complete transfer of all blocks within a linked list.

A transfer descriptor describes how a block transfer should be carried out by the DMAC, and it must remain in SRAM. For further details on the transfer descriptor refer to 27.5.2.3. Transfer Descriptors.

The figure above shows several block transfers linked together, which are called linked descriptors. For further information about linked descriptors, refer to 27.5.3.1. Linked Descriptors.

A DMA transfer is initiated by an incoming transfer trigger on one of the DMA channels. This trigger can be configured to be either a software trigger, an event trigger, or one of the dedicated peripheral triggers. The transfer

trigger will result in a DMA transfer request from the specific channel to the arbiter. If there are several DMA channels with pending transfer requests, the arbiter chooses which channel is granted access to become the active channel. The DMA channel granted access as the active channel will carry out the transaction as configured in the transfer descriptor. A current transaction can be interrupted by a higher prioritized channel, but will resume the block transfer when the according DMA channel is granted access as the active channel again.

For each beat transfer, an optional output event can be generated. For each block transfer, optional interrupts and an optional output event can be generated. When a transaction is completed, dependent of the configuration, the DMA channel will either be suspended or disabled.

27.5.1.2 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 27.5.3.7. CRC Operation for details.

27.5.2 Basic Operation

27.5.2.1 Initialization

The following DMAC registers are enable-protected, meaning that they can only be written when the DMAC is disabled (CTRL.DMAENABLE=0):

- Descriptor Base Memory Address register (BASEADDR)
- Write-Back Memory Base Address register (WRBADDR)

The following DMAC bit is enable-protected, meaning that it can only be written when both the DMAC and CRC are disabled (CTRL.DMAENABLE=0 and CTRL.CRCENABLE=0):

- Software Reset bit in Control register (CTRL.SWRST)

The following DMA channel register is enable-protected, meaning that it can only be written when the corresponding DMA channel is disabled (CHCTRLA.ENABLE=0):

- Channel Control B (CHCTRLB) register, except the Command bit (CHCTRLB.CMD) and the Channel Arbitration Level bit (CHCTRLB.LVL)

The following DMA channel bit is enable-protected, meaning that it can only be written when the corresponding DMA channel is disabled:

- Channel Software Reset bit in Channel Control A register (CHCTRLA.SWRST)

The following CRC registers are enable-protected, meaning that they can only be written when the CRC is disabled (CTRL.CRCENABLE=0):

- CRC Control register (CRCCTRL)
- CRC Checksum register (CRCCHKSUM)

Enable-protection is denoted by the "Enable-Protected" property in the register description.

Before the DMAC is enabled it must be configured, as outlined by the following steps:

- 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 should 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)

Before a DMA channel is enabled, the DMA channel and the corresponding first transfer descriptor must be configured, as outlined by the following steps:

- DMA channel configurations
  - The channel number of the DMA channel to configure must be written to the Channel ID (CHID) register
  - Trigger action must be selected by writing the Trigger Action bit group in the Channel Control B register (CHCTRLB.TRIGACT)

  - Trigger source must be selected by writing the Trigger Source bit group in the Channel Control B register (CHCTRLB.TRIGSRC)
- 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 register (BTCTRL.BEATSIZE)
  - The transfer descriptor must be made valid by writing a one to the Valid bit in the Block Transfer Control register (BTCTRL.VALID)
  - 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

If CRC calculation is needed, the CRC engine must be configured before it is enabled, as outlined by the following steps:

- The CRC input source must selected by writing the CRC Input Source bit group in the CRC Control register (CRCCTRL.CRCSRC)
- The type of CRC calculation must be selected by writing the CRC Polynomial Type bit group in the CRC Control register (CRCCTRL.CRCPOLY)
- 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 register (CRCCTRL.CRCBEATSIZE)

27.5.2.2 Enabling, Disabling, and Resetting

The DMAC is enabled by writing the DMA Enable bit in the Control register (CTRL.DMAENABLE) to '1'. The DMAC is disabled by writing a '0' to CTRL.DMAENABLE.

A DMA channel is enabled by writing the Enable bit in the Channel Control A register (CHCTRLA.ENABLE) to '1', after writing the corresponding channel id to the Channel ID bit group in the Channel ID register (CHID.ID). A DMA channel is disabled by writing a '0' to CHCTRLA.ENABLE.

The CRC is enabled by writing a '1' to the CRC Enable bit in the Control register (CTRL.CRCENABLE). The CRC is disabled by writing a '0' to CTRL.CRCENABLE.

The DMAC is reset by writing a '1' to the Software Reset bit in the Control register (CTRL.SWRST) while the DMAC and CRC are disabled. All registers in the DMAC except DBGCTRL will be reset to their initial state.

A DMA channel is reset by writing a '1' to the Software Reset bit in the Channel Control A register (CHCTRLA.SWRST), after writing the corresponding channel id to the Channel ID bit group in the Channel ID register (CHID.ID). The channel registers will be reset to their initial state. The corresponding DMA channel must be disabled in order for the reset to take effect.

27.5.2.3 Transfer Descriptors

Together with the channel configurations the transfer descriptors decides how a block transfer should be executed. Before a DMA channel is enabled (CHCTRLA.ENABLE is written to one), and receives a transfer trigger, its first transfer descriptor has to be initialized and valid (BTCTRL.VALID). The first transfer descriptor describes the first block transfer of a transaction.

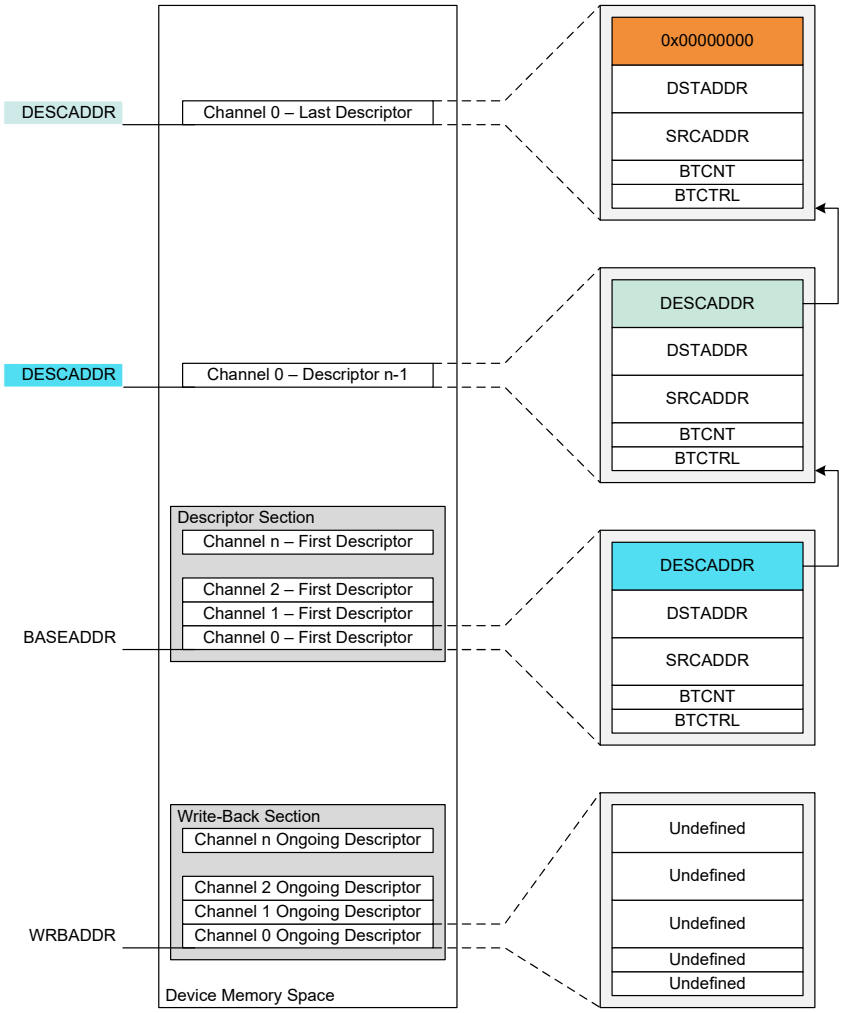
All transfer descriptors must reside in SRAM. The addresses stored in the Descriptor Memory Section Base Address (BASEADDR) and Write-Back Memory Section Base Address (WRBADDR) registers tell the DMAC where to find the descriptor memory section and the write-back memory section.

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 figure below), all first transfer descriptors must be stored in a contiguous memory section, where the transfer descriptors must be ordered according to their channel number. For further details on linked descriptors, refer to 27.5.3.1. Linked Descriptors.

The write-back memory section is the section 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 will be 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. For further details on linked descriptors, refer to 27.5.3.1. Linked Descriptors.

Figure 27-3. Memory Sections



The size of the descriptor and write-back memory sections is 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 always 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 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.

27.5.2.4 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 active channel is the DMA channel being granted access to perform its next transfer. When the arbiter has granted a DMA channel access to the DMAC, the corresponding bit PENDCH.PENDCHx will be cleared. See also the following figure.

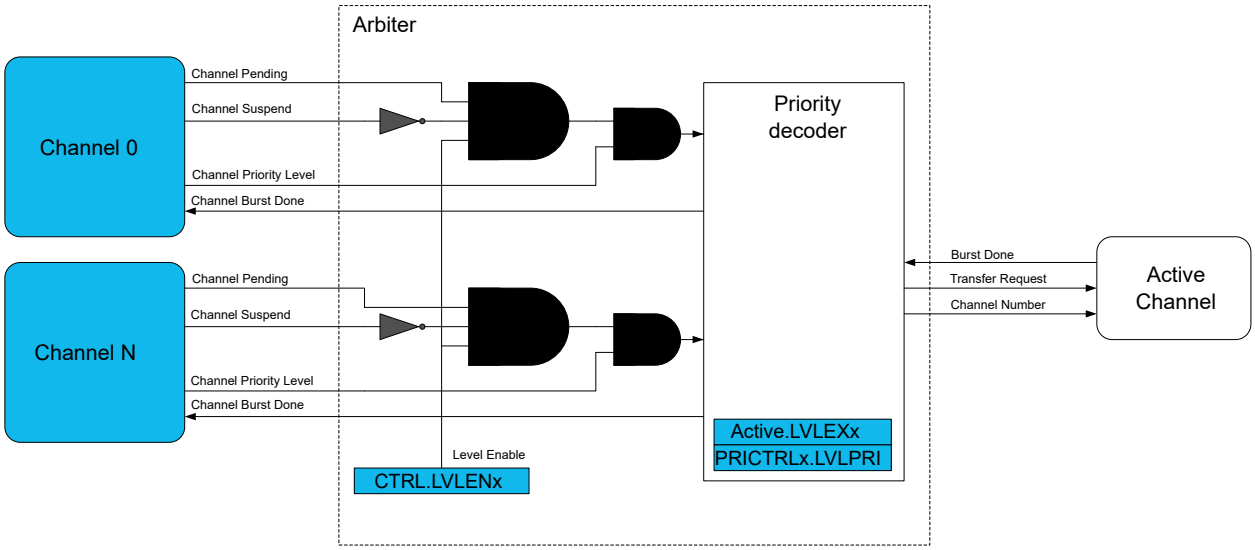
If the upcoming transfer is the first for the transfer request, the corresponding Busy Channel x bit in the Busy Channels register will be set (BUSYCH.BUSYCHx=1), and it will remain '1' for the subsequent granted transfers.

When the channel has performed its granted 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 fed into the queue of channels with pending transfers, the corresponding BUSYCH.BUSYCHx will remain '1'. 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. 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 27-4. 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).

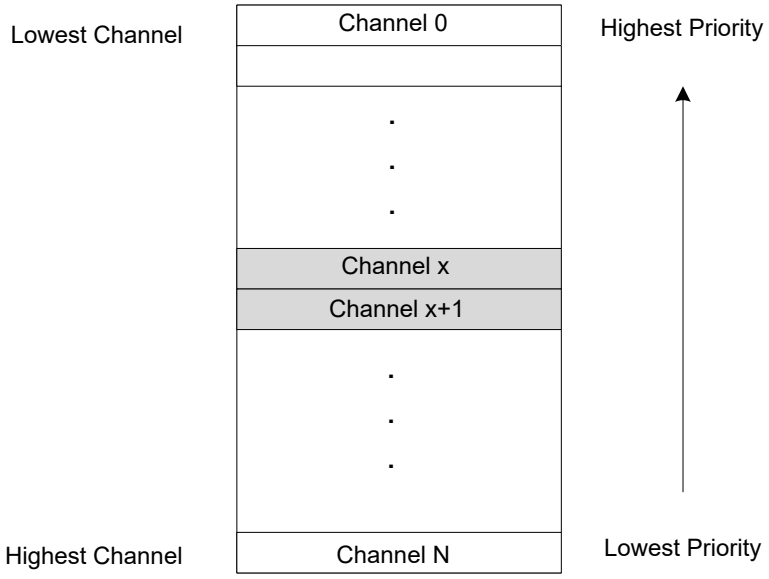
Each DMA channel supports a 4-level priority scheme. The priority level for a channel is configured by writing to the Channel Arbitration Level bit group in the Channel Control B register (CHCTRLB.LVL). As long as all priority levels are enabled, a channel with a higher priority level number will have priority over a channel with a lower priority level number. Each priority level x is enabled by setting the corresponding Priority Level x Enable bit in the Control register (CTRL.LVLENx=1).

Within each priority level the DMAC's arbiter can be configured to prioritize statically or dynamically:

Static Arbitration within a priority level is selected by writing a '0' to the Level x Round-Robin Scheduling Enable bit in the Priority Control register (PRICTRL.RRLVLENx).

When static arbitration is selected, the arbiter will prioritize a low channel number over a high channel number as shown in the figure below. When using the static arbitration 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.

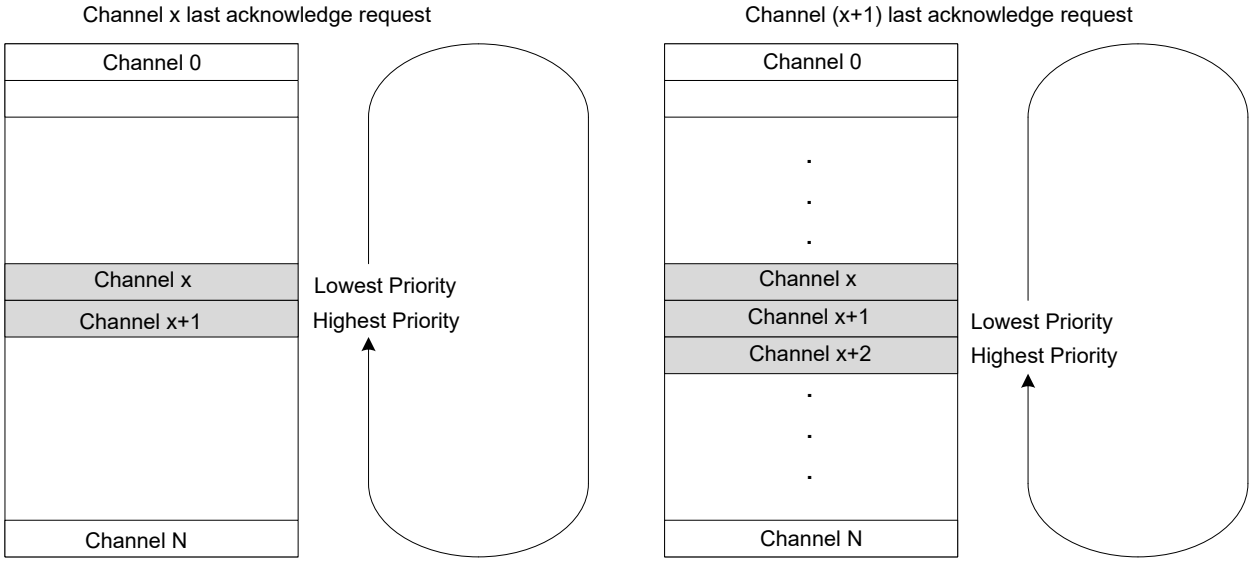
Figure 27-5. Static Priority Scheduling



Dynamic Arbitration within a priority level is selected by writing a '1' to PRICTRL.RRLVLENx.

The dynamic arbitration scheme in the DMAC is round-robin. 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 Figure 27-6. 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 register (PRICTRL.LVLPRIx) for the corresponding priority level.

Figure 27-6. Dynamic (Round-Robin) Priority Scheduling



27.5.2.5 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 (refer to DMA Block Diagram section) 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, refer to the section on Addressing.

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.

27.5.2.6 Transfer Triggers and Actions

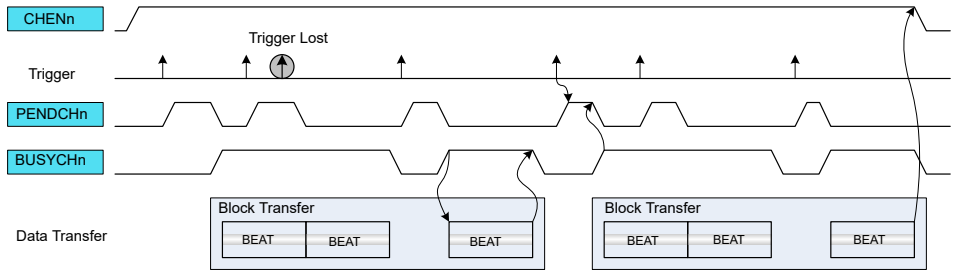
A DMA transfer through a DMA channel can be started only when a DMA transfer request is detected, and the DMA channel has been granted access to the DMA. A transfer request can be triggered from software, from a peripheral, or from an event. There are dedicated Trigger Source selections for each DMA Channel Control B (CHCTRLB.TRIGSRC).

The trigger actions are available in the Trigger Action bit group in the Channel Control B register (CHCTRLB.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. If 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 beat transfer (CHCTRLB.TRIGACT=0x2) or transaction transfer (CHCTRLB.TRIGACT=0x3) instead of a block transfer (CHCTRLB.TRIGACT=0x0).

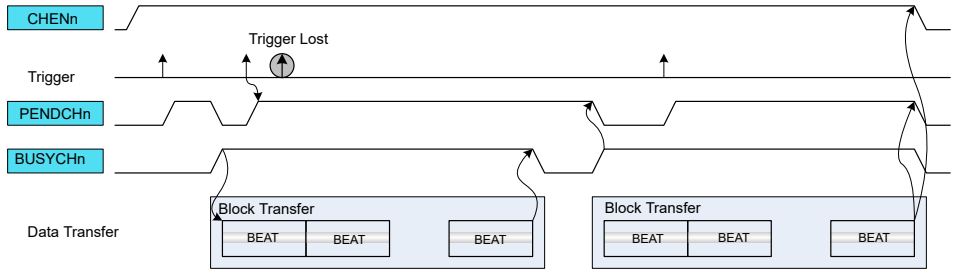
Figure 27-7 shows an example where triggers are used with two linked block descriptors.

Figure 27-7. Trigger Action and Transfers

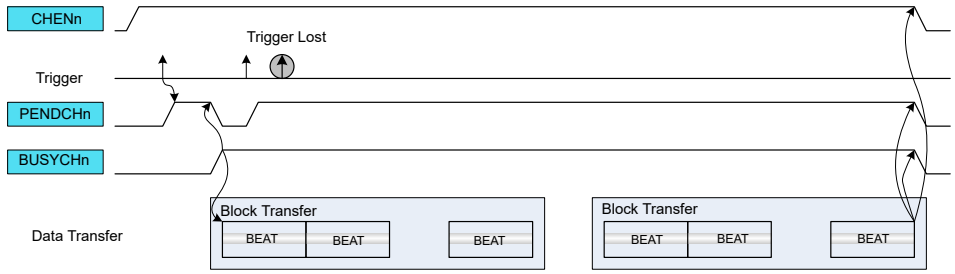
Beat Trigger Action



Block Trigger Action



Transaction Trigger Action



If the trigger source generates a transfer request for a channel during an ongoing transfer, the new transfer request will be kept pending (CHSTATUS.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 Status register (CHSTATUS.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.

27.5.2.7 Addressing

Each block transfer needs to have both a source address and a destination address defined. The source address is set by writing the Transfer Source Address (SRCADDR) register, the destination address is set by writing the Transfer Destination Address (SRCADDR) register.

The addressing of this DMAC module can be static or incremental, for either source or destination of a block transfer, or both.

Incrementation for the source address of a block transfer is enabled by writing the Source Address Incrementation Enable bit in the Block Transfer Control register (BTCTRL.SRCINC=1). The step size of the incrementation is configurable and can be chosen by writing the Step Selection bit in the Block Transfer Control register (BTCTRL.STEPSEL=1) and writing the desired step size in the Address Increment Step Size bit group in the Block Transfer Control register (BTCTRL.STEPSIZE). If BTCTRL.STEPSEL = 0, the step size for the source incrementation will be the size of one beat.

When source address incrementation is configured (BTCTRL.SRCINC = 1), SRCADDR is calculated as follows:

If BTCTRL.STEPSEL = 1:

$$\text{SRCADDR} = \text{SRCADDR}_{START} + BTCNT \cdot \left(BEATSIZE + 1\right) \cdot 2^{\text{STEPSIZE}}$$

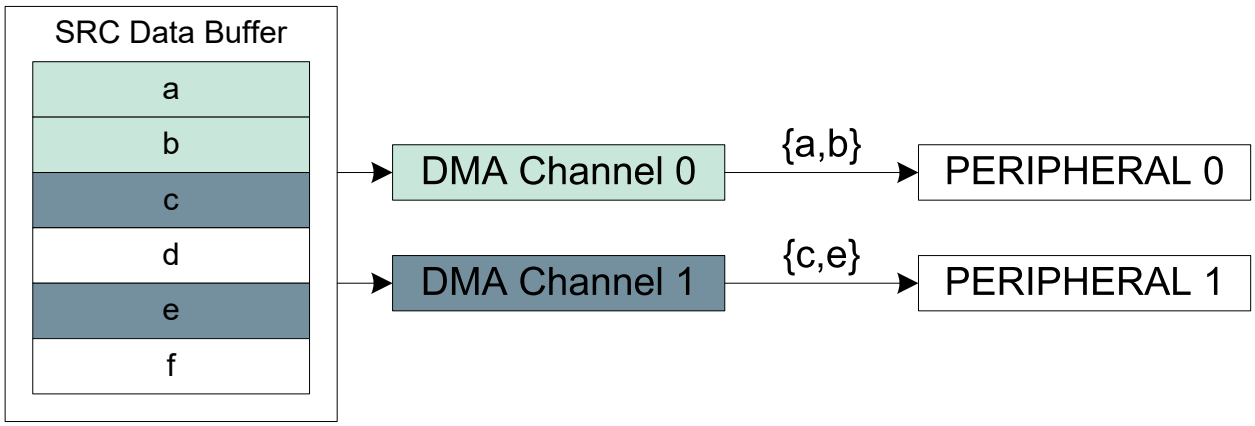
If BTCTRL.STEPSEL = 0:

$$\text{SRCADDR} = \text{SRCADDR}_{START} + BTCNT \cdot \left(BEATSIZE + 1\right)$$

- 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 27-8. 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.

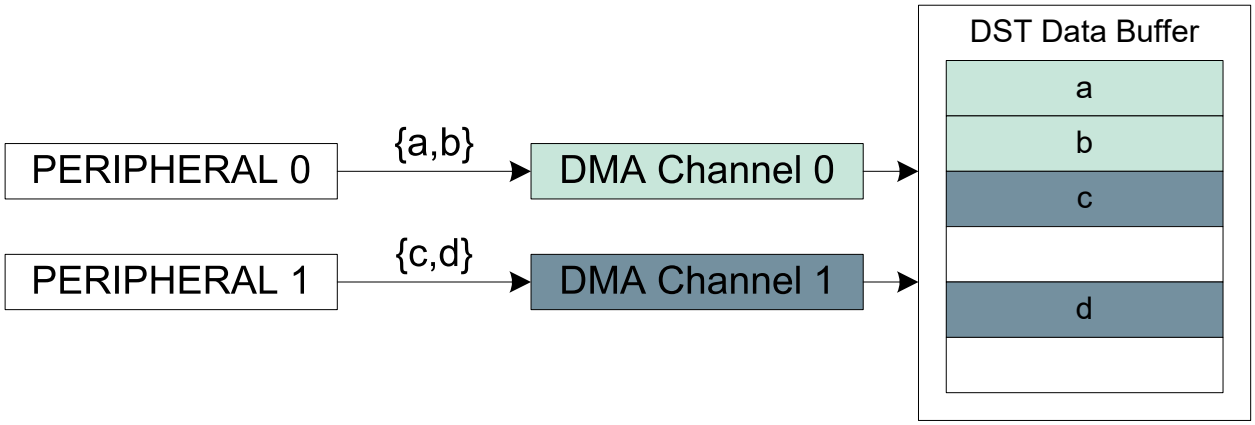
When the destination address incrementation is configured (BTCTRL.DSTINC = 1), DSTADDR must be set and calculated as follows:

$DSTADDR = DSTADDR_{START} + BTCNT \bullet \left(BEATSIZE + 1\right) \bullet 2^{STEPSIZE}$	where BTCTRL.STEPSEL is zero
$DSTADDR = DSTADDR_{START} + BTCNT \bullet \left(BEATSIZE + 1\right)$	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 followiong 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 27-9. Destination Address Increment



27.5.2.8 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.

27.5.3 Additional Features

27.5.3.1 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.

Figure 27-3 illustrates how linked descriptors work. 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 (27.7.5. DESCADDR) register of the first transfer descriptor. Fetching the next transfer descriptor (27.7.5. DESCADDR) is continued until the last transfer descriptor. When the block transfer for the last transfer descriptor is executed and 27.7.5. DESCADDR=0x00000000, the transaction is terminated. For further details on how the next descriptor is fetched from SRAM, refer to section 27.5.2.5. Data Transmission.

27.5.3.1.1 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 27.7.5. DESCADDR value of descriptor A to point to descriptor C instead of descriptor B.
   c. Set the 27.7.5. 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.

27.5.3.1.2 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 (27.7.5. DESCADDR)
   – Set the destination address (27.7.4. DSTADDR)
   – Set the source address (27.7.3. SRCADDR)
   – Configure the block transfer control (27.7.1. 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 27.7.5. DESCADDR from the Write-Back memory.
   – If the DMA has not already fetched the descriptor which requires changes (i.e., DESCADDR is wrong):
      • Update the 27.7.5. 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 which requires changes:
      • Set the Channel Suspend software command and wait for the Suspend interrupt
      • Update the next descriptor address (27.7.5. DESCADDR) in the write-back memory
      • Clear the interrupt sources and set the Resume software command
      • Update the 27.7.5. 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.

27.5.3.1.3 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.

27.5.3.2 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.

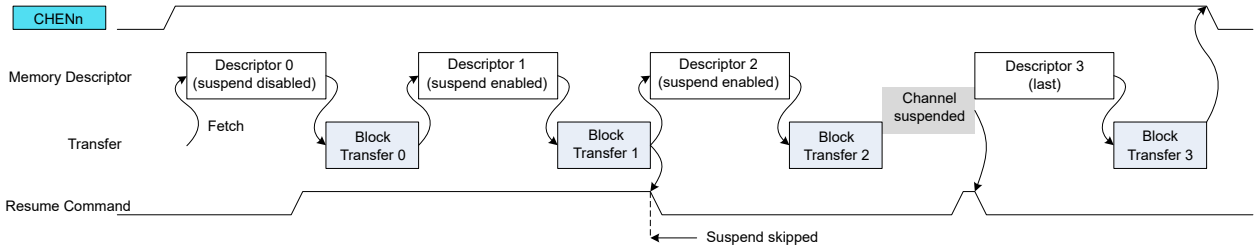
Note: Only enabled DMA channels can be suspended. If a channel is disabled when it is attempted to be suspended, the internal suspend command will be ignored.

For more details on transfer descriptors, refer to section 27.5.2.3. Transfer Descriptors.

27.5.3.3 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 27-10. Channel Suspend/Resume Operation



27.5.3.4 Event Input Actions

The event input actions are available only on the 8 least significant DMA channels. For details on channels with event input support, refer to the in the Event system documentation.

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 Control B register (CHCTRLB.EVIE) must be written to '1'. Refer also to 27.5.5. Events.

Table 27-2. Event Input Action

Action	CHCTRLB.EVACT	CHCTRLB.TRGSRC
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	

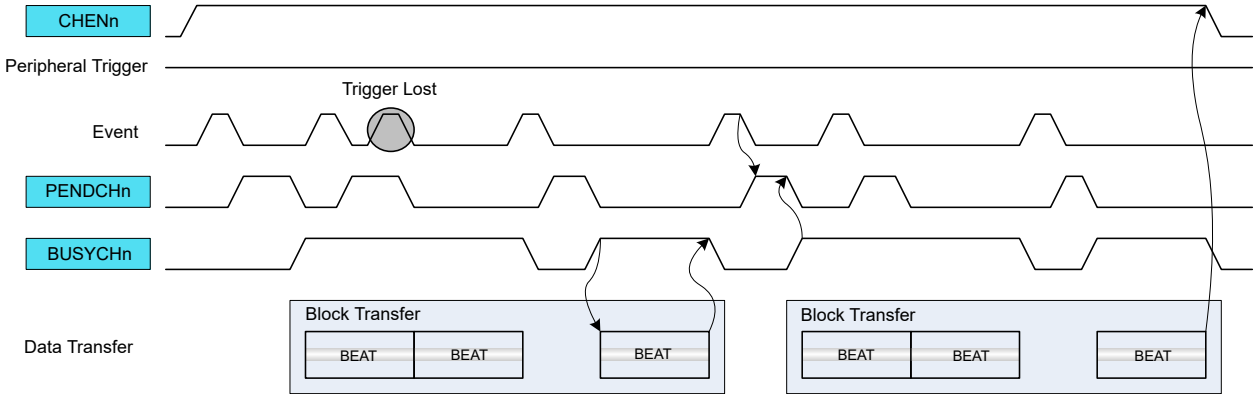
Normal Transfer

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 (27.6.23. CHSTATUS.PEND) and the corresponding Channel n bit in the Pending Channels register (27.6.13. PENDCH.PENDCHn) are set. If the event is received while the channel is pending, the event trigger is lost.

The figure below shows an example where beat transfers are enabled by internal events.

Figure 27-11. Beat Event Trigger Action



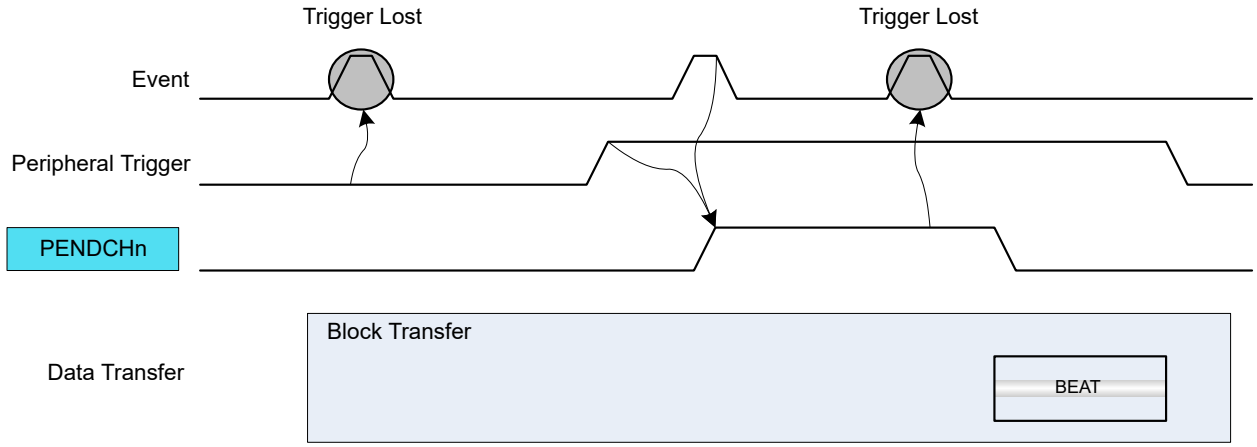
Conditional Transfer on Strobe

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, e.g. 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 (i.e. the channel is not pending) and when an active event is received. If the peripheral trigger is active, the DMA will wait for an event before the peripheral trigger is internally registered. When both event and peripheral transfer trigger are active, both 27.6.23. CHSTATUS.PEND and 27.6.13. PENDCH.PENDCHn are set. A software trigger will now trigger a transfer.

The figure below shows an example where the peripheral beat transfer is started by a conditional strobe event action.

Figure 27-12. Periodic Event with Beat Peripheral Triggers



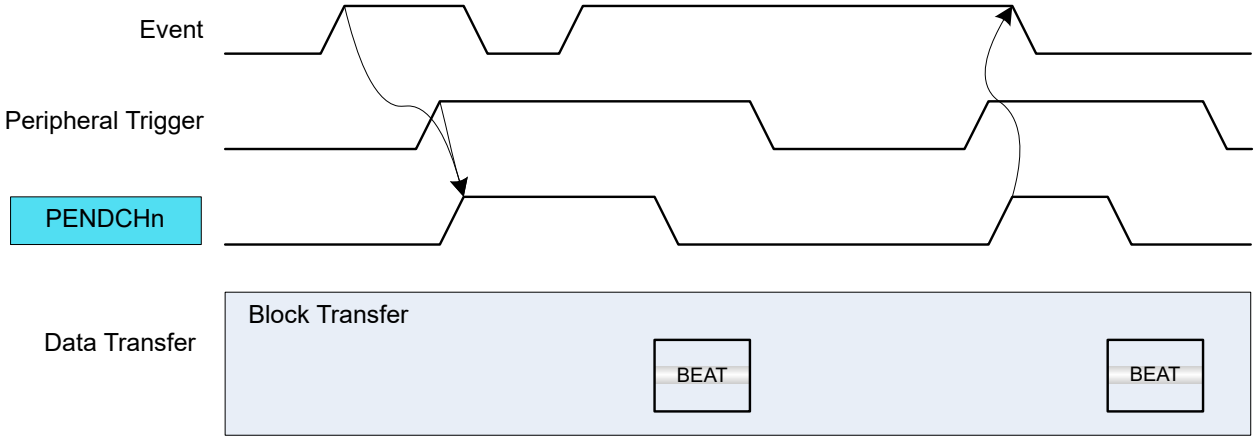
Conditional Transfer

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 (27.6.23. CHSTATUS.PEND), the respective Pending Channel n Bit in the Pending Channels register is set (27.6.13. PENDCH.PENDCHn), and the event is acknowledged. A software trigger will now trigger a transfer.

The figure below shows an example where conditional event is enabled with peripheral beat trigger requests.

Figure 27-13. Conditional Event with Beat Peripheral Triggers

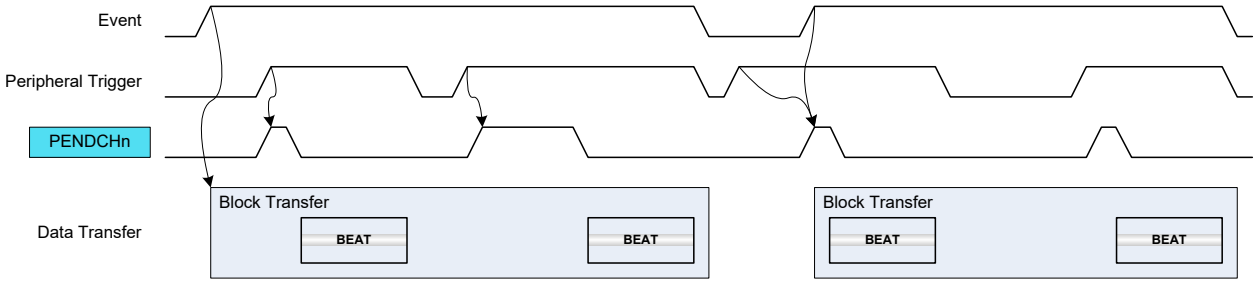


Conditional Block Transfer
The event input is used to trigger a conditional block transfer on peripherals.

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 figure below shows an example where conditional event block transfer is started with peripheral beat trigger requests.

Figure 27-14. Conditional Block Transfer with Beat 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 further details on Channel Suspend, refer to 27.5.3.2. Channel Suspend.

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 (27.6.22. CHINTFLAG.SUSP) is cleared. For further details refer to 27.5.3.2. Channel Suspend.

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.

27.5.3.5 Event Output Selection

Event output selection is available only for the 8 least significant DMA channels. The pulse width of an event output from a channel is one AHB clock cycle.

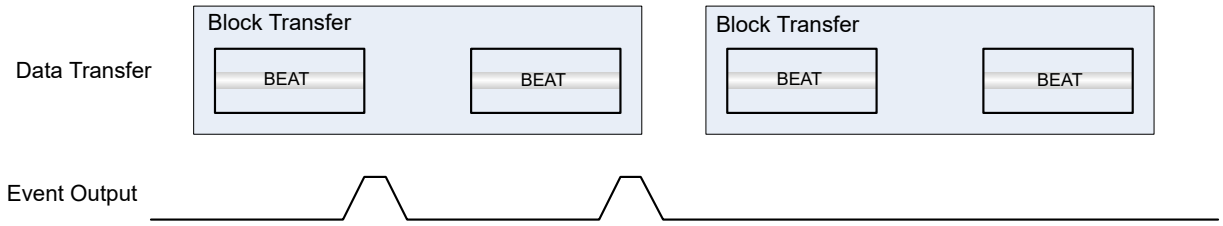
The output of channel events is enabled by writing a '1' to the Channel Event Output Enable bit in the Control B register (CHCTRLB.EVOE). The event output cause is selected by writing to the Event Output Selection bits in

the Block Transfer Control register (BTCTRL.EVOSEL). It is possible to generate events after each block transfer (BTCTRL.EVOSEL=0x1) or beat transfer (BTCTRL.EVOSEL=0x3). To enable an event being generated when a transaction is complete, the block event selection must be set in the last transfer descriptor only.

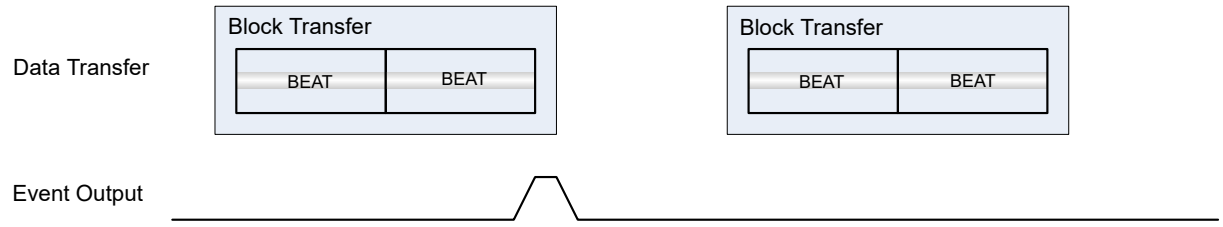
The figure Figure 27-15 shows an example where the event output generation is enabled in the first block transfer, and disabled in the second block.

Figure 27-15. Event Output Generation

Beat Event Output



Block Event Output



27.5.3.6 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.

27.5.3.7 CRC Operation

A cyclic redundancy check (CRC) is an error detection technique used to find errors in data. It is commonly used to determine whether the data during a transmission, or data present in data and program memories has been corrupted or not. A CRC takes a data stream or a block of data as input and generates a 16- or 32-bit output that can be appended to the data and used as a checksum.

When the data is received, the device or application repeats the calculation: If the new CRC result does not match the one calculated earlier, the block contains a data error. The application will then detect this and may take a corrective action, such as requesting the data to be sent again or simply not using the incorrect data.

The CRC engine in DMAC supports two commonly used CRC polynomials: CRC-16 (CRC-CCITT) and CRC-32 (IEEE 802.3). Typically, applying CRC-n (CRC-16 or CRC-32) to a data block of arbitrary length will detect any single alteration that is ≤n bits in length, and will detect the fraction 1-2-n of all longer error bursts.

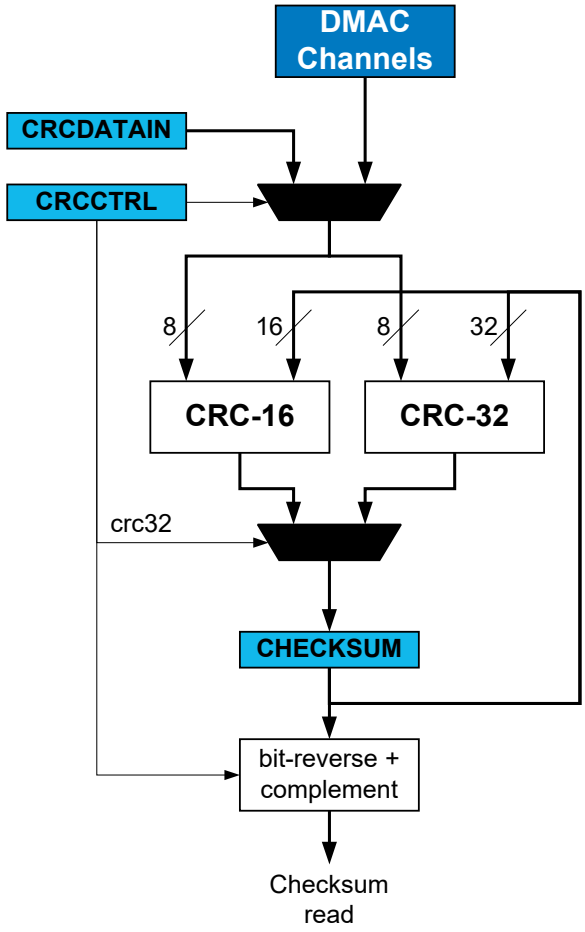
- CRC-16:

  - 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 (27.6.4. CRCCHKSUM). When CRC-32 polynomial is used, the final checksum read is bit reversed and complemented, as shown in Figure 27-16.

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 27.6.3. CRCDATAIN register and the CRC engine will operate on the input data in a byte by byte manner.

Figure 27-16. 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 (27.6.3. 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 27.6.3. 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 27.6.3. 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.

27.5.4 Interrupts

The DMAC channels have the following interrupt sources:

- Transfer Complete (TCMPL): Indicates that a block transfer is completed on the corresponding channel. Refer to 27.5.2.5. Data Transmission for details.
- Transfer Error (TERR): Indicates that a bus error has occurred during a burst transfer, or that an invalid descriptor has been fetched. Refer to 27.5.2.8. Error Handling for details.
- Channel Suspend (SUSP): Indicates that the corresponding channel has been suspended. Refer to 27.5.3.2. Channel Suspend and 27.5.2.5. Data Transmission for details.

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).

An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until the interrupt flag is cleared, the interrupt is disabled, the DMAC is reset or the corresponding DMA channel is reset. See CHINTFLAG for details on how to clear interrupt flags. All interrupt requests are ORed together on system level to generate one combined interrupt request to the NVIC.

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.

Important: INTPEND is dedicated to channels which are multiplexed on a single interrupt line (Channels 4 to 15).

27.5.5 Events

The DMAC can generate the following output events:

- Channel (CH0-3): Generated when a block transfer for a given channel has been completed, or when a beat transfer within a block transfer for a given channel has been completed. Refer to Event Output Selection for details.

Setting the Channel Control B Event Output Enable bit (CHCTRLB.EVOE=1) enables the corresponding output event configured in the Event Output Selection bit group in the Block Transfer Control register (BTCTRL.EVOSEL). Clearing CHCTRLB.EVOE=0 disables the corresponding output event.

The DMAC can take the following actions on an input event (CH0-3):

- 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 Control B Event Input Enable bit (CHCTRLB.EVIE=1) enables the corresponding action on input event. Clearing this bit disables the corresponding action on input event. Note that several actions can be enabled for incoming events. If several events are connected to the peripheral, any enabled action will be taken for any of the incoming events. For further details on event input actions, refer to Event Input Actions.
Note: Event input and outputs are not available for every channel. Refer to 27.2. Features for more information.

27.5.6 Sleep Mode Operation

Each DMA channel can be configured to operate in any sleep mode. To be able to run in standby, the RUNSTDBY bit in Channel Control A register (CHCTRLA.RUNSTDBY) must be written to '1'. The DMAC can wake up the device using interrupts from any sleep mode or perform actions through the Event System.

For channels with CHCTRLA.RUNSTDBY = 0, it is up to software to stop DMA transfers on these channels and wait for completion before going to standby mode using the following sequence:

1. Suspend the DMAC channels for which CHCTRLA.RUNSTDBY = 0.
2. Check the SYNCBUSY bits of registers accessed by the DMAC channels being suspended.
3. Go to sleep.
4. When the device wakes up, resume the suspended channels.

27.5.7 Debug Operation

When the CPU is halted in debug mode the DMAC will halt normal operation. The DMAC can be forced to continue operation during debugging. Refer to 27.6.6. DBGCTRL for details.

27.6 Register Summary

Refer to the Registers Description section for more details on register properties and access permissions.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRL	15:8					LVLEN3	LVLEN2	LVLEN1	LVLEN0
		7:0						CRCENABLE	DMAENABLE	SWRST
0x02	CRCCTRL	15:8			CRCSRC[5:0]					
		7:0					CRCPOLY[1:0]		CRCBEATSIZE[1:0]	
0x04	CRCDATAIN	31:24	CRCDATAIN[31:24]							
		23:16	CRCDATAIN[23:16]							
		15:8	CRCDATAIN[15:8]							
		7:0	CRCDATAIN[7:0]							
0x08	CRCCHKSUM	31:24	CRCCHKSUM[31:24]							
		23:16	CRCCHKSUM[23:16]							
		15:8	CRCCHKSUM[15:8]							
		7:0	CRCCHKSUM[7:0]							
0x0C	CRCSTATUS	7:0							CRCZERO	CRCBUSY
0x0D	DBGCTRL	7:0								DBGRUN
0x0E	QOSCTRL	7:0			DQOS[1:0]		FQOS[1:0]		WRBQOS[1:0]	
0x0F	Reserved									
0x10	SWTRIGCTRL	31:24								
		23:16								
		15:8	SWTRIG15	SWTRIG14	SWTRIG13	SWTRIG12	SWTRIG11	SWTRIG10	SWTRIG9	SWTRIG8
		7:0	SWTRIG7	SWTRIG6	SWTRIG5	SWTRIG4	SWTRIG3	SWTRIG2	SWTRIG1	SWTRIG0
0x14	PRICTRL0	31:24	RRLVLEN3				LVLPRI3[3:0]			
		23:16	RRLVLEN2				LVLPRI2[3:0]			
		15:8	RRLVLEN1				LVLPRI1[3:0]			
		7:0	RRLVLEN0				LVLPRI0[3:0]			
0x18 ... 0x1F	Reserved									
0x20	INTPEND	15:8	PEND	BUSY	FERR			SUSP	TCMPL	TERR
		7:0					ID[3:0]			
0x22 ... 0x23	Reserved									
0x24	INTSTATUS	31:24								
		23:16								
		15:8	CHINT15	CHINT14	CHINT13	CHINT12	CHINT11	CHINT10	CHINT9	CHINT8
		7:0	CHINT7	CHINT6	CHINT5	CHINT4	CHINT3	CHINT2	CHINT1	CHINT0
0x28	BUSYCH	31:24								
		23:16								
		15:8	BUSYCH15	BUSYCH14	BUSYCH13	BUSYCH12	BUSYCH11	BUSYCH10	BUSYCH9	BUSYCH8
		7:0	BUSYCH7	BUSYCH6	BUSYCH5	BUSYCH4	BUSYCH3	BUSYCH2	BUSYCH1	BUSYCH0
0x2C	PENDCH	31:24								
		23:16								
		15:8	PENDCH15	PENDCH14	PENDCH13	PENDCH12	PENDCH11	PENDCH10	PENDCH9	PENDCH8
		7:0	PENDCH7	PENDCH6	PENDCH5	PENDCH4	PENDCH3	PENDCH2	PENDCH1	PENDCH0
0x30	ACTIVE	31:24	BTCNT[15:8]							
		23:16	BTCNT[7:0]							
		15:8	ABUSY			ID[4:0]				
		7:0					LVLEX3	LVLEX2	LVLEX1	LVLEX0
0x34	BASEADDR	31:24	BASEADDR[31:24]							
		23:16	BASEADDR[23:16]							
		15:8	BASEADDR[15:8]							
		7:0	BASEADDR[7:0]							

.....continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x38	WRBADDR	31:24	WRBADDR[31:24]							
		23:16	WRBADDR[23:16]							
		15:8	WRBADDR[15:8]							
		7:0	WRBADDR[7:0]							
0x3C ... 0x3E	Reserved									
0x3F	CHID	7:0					ID[3:0]			
0x40	CHCTRLA	7:0		RUNSTDBY					ENABLE	SWRST
0x41 ... 0x43	Reserved									
0x44	CHCTRLB	31:24							CMD[1:0]	
		23:16	TRIGACT[1:0]							
		15:8			TRIGSRC[5:0]					
		7:0		LVL[1:0]		EVOE	EVIE	EVACT[2:0]		
0x48 ... 0x4B	Reserved									
0x4C	CHINTENCLR	7:0						SUSP	TCMPL	TERR
0x4D	CHINTENSET	7:0						SUSP	TCMPL	TERR
0x4E	CHINTFLAG	7:0						SUSP	TCMPL	TERR
0x4F	CHSTATUS	7:0						FERR	BUSY	PEND

27.6.1 Control

Name: CTRL
Offset: 0x00
Reset: 0x00X0
Property: PAC Write-Protection, Enable-Protected Bits

Bit	15	14	13	12	11	10	9	8
					LVLEN3	LVLEN2	LVLEN1	LVLEN0
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
						CRCENABLE	DMAENABLE	SWRST
Access						R/W	R/W	R/W
Reset						0	0	0

Bits 8, 9, 10, 11 – LVLENx Priority Level x Enable [x=0..3]
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, refer to the 27.5.2.4. Arbitration section.
Note: This bit field is 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 2 – CRCENABLE CRC Enable
Writing a '0' to this bit will disable the CRC calculation when the CRC Status Busy flag is cleared (CRCSTATUS. CRCBUSY). The bit is zero when the CRC is disabled.
Writing a '1' to this bit will enable the CRC calculation.
Note: This bit is not enable-protected.

Value	Description
0	The CRC calculation is disabled
1	The CRC calculation is enabled

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.
Note: 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 both the DMAC and the CRC module are disabled (DMAENABLE and CRCENABLE are '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

27.6.2 CRC Control

Name: CRCCTRL
Offset: 0x02
Reset: 0x0000
Property: PAC Write-Protection, Enable-Protected

Bit	15	14	13	12	11	10	9	8
			CRCSRC[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
					CRCPOLY[1:0]		CRCBEATSIZE[1:0]	
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bits 13:8 – CRCSRC[5:0] CRC Input Source
These bits select the input source for generating the CRC, as shown in the table below. 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	CHN0	DMA channel 0
0x21	CHN1	DMA channel 1
0x22	CHN2	DMA channel 2
0x23	CHN3	DMA channel 3
0x24	CHN4	DMA channel 4
0x25	CHN5	DMA channel 5
0x26	CHN6	DMA channel 6
0x27	CHN7	DMA channel 7
0x28	CHN8	DMA Channel 8
0x29	CHN9	DMA Channel 9
0x2A	CHN10	DMA Channel 10
0x2B	CHN11	DMA Channel 11
0x2C	CHN12	DMA Channel 12
0x2D	CHN13	DMA Channel 13
0x2E	CHN14	DMA Channel 14
0x2F	CHN15	DMA Channel 15

Bits 3:2 – CRCPOLY[1:0] CRC Polynomial Type
These bits define the size of the data transfer for each bus access when the CRC is used with I/O interface, as shown in the table below.

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

Value	Name	Description
0x1	HWORD	16-bit bus transfer
0x2	WORD	32-bit bus transfer
0x3	-	Reserved

27.6.3 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.

27.6.4 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.

27.6.5 CRC Status

Name: CRCSTATUS
Offset: 0x0C
Reset: 0x00
Property: PAC Write-Protection

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

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.
When running CRC-32 and appending the checksum at the end of the packet (as little endian), the final checksum should be 0x2144df1c, and not zero. However, if the checksum is complemented before it is appended (as little endian) to the data, the final result in the checksum register will be zero. See the description of CRCCHKSUM to read out different versions of the checksum.

Bit 0 – CRCBUSY CRC Module Busy

This flag is cleared by writing a one to it when used with I/O interface. When used with a DMA channel, the bit is set when the corresponding DMA channel is enabled, and cleared when the corresponding DMA channel is disabled. This register bit cannot be cleared by the application when the CRC is used with a DMA channel.
This bit is set when a source configuration is selected and as long as the source is using the CRC module.

27.6.6 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.

27.6.7 Quality of Service Control

Name: QOSCTRL
Offset: 0x0E
Reset: 0x2A
Property: PAC Write-Protection

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

Bits 5:4 – DQOS[1:0] Data Transfer Quality of Service
These bits define the memory priority access during the data transfer operation.

Value	Name	Description
0x0	DISABLE	Background (no sensitive operation)
0x1	LOW	Sensitive Bandwidth
0x2	MEDIUM	Sensitive Latency
0x3	HIGH	Critical Latency

Bits 3:2 – FQOS[1:0] Fetch Quality of Service
These bits define the memory priority access during the fetch operation.

Value	Name	Description
0x0	DISABLE	Background (no sensitive operation)
0x1	LOW	Sensitive Bandwidth
0x2	MEDIUM	Sensitive Latency
0x3	HIGH	Critical Latency

Bits 1:0 – WRBQOS[1:0] Write-Back Quality of Service
These bits define the memory priority access during the write-back operation.

Value	Name	Description
0x0	DISABLE	Background (no sensitive operation)
0x1	LOW	Sensitive Bandwidth
0x2	MEDIUM	Sensitive Latency
0x3	HIGH	Critical Latency

27.6.8 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
	SWTRIG15	SWTRIG14	SWTRIG13	SWTRIG12	SWTRIG11	SWTRIG10	SWTRIG9	SWTRIG8
Access	R/W	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
	SWTRIG7	SWTRIG6	SWTRIG5	SWTRIG4	SWTRIG3	SWTRIG2	SWTRIG1	SWTRIG0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 – SWTRIGn Channel n Software Trigger [n = 15..0]
This bit is cleared when the Channel Pending bit in the Channel Status register (27.6.23. CHSTATUS.PEND) for the corresponding channel is either set, or by writing a '1' to it.
This bit is set if 27.6.23. CHSTATUS.PEND is already '1' when writing a '1' to that bit.
Writing a '0' to this bit will clear the bit.
Writing a '1' to this bit will generate a DMA software trigger on channel x, if 27.6.23. CHSTATUS.PEND=0 for channel x. CHSTATUS.PEND will be set and SWTRIGn will remain cleared.

27.6.9 Priority Control

Name: PRICTRL0
Offset: 0x14
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
	RRLVLEN3				LVLPRI3[3:0]			
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
	RRLVLEN2				LVLPRI2[3:0]			
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
	RRLVLEN1				LVLPRI1[3:0]			
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
	RRLVLEN0				LVLPRI0[3:0]			
Access	R/W				R/W	R/W	R/W	R/W
Reset	0				0	0	0	0

Bit 31 – RRLVLEN3 Level 3 Round-Robin Arbitration Enable
This bit controls which arbitration scheme is selected for DMA channels with priority level 3. For details on arbitration schemes, refer to 27.5.2.4. Arbitration.

Value	Description
0	Static arbitration scheme for channels with level 3 priority.
1	Round-robin arbitration scheme for channels with level 3 priority.

Bits 27:24 – LVLPRI3[3:0] Level 3 Channel Priority Number
When round-robin arbitration is enabled (RRLVLEN3=1) for priority level 3, this register holds the channel number of the last DMA channel being granted access as the active channel with priority level 3.
When static arbitration is enabled (RRLVLEN3=0) for priority level 3, 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 (RRLVLEN3 written to '0').

Bit 23 – RRLVLEN2 Level 2 Round-Robin Arbitration Enable
This bit controls which arbitration scheme is selected for DMA channels with priority level 2. For details on arbitration schemes, refer to 27.5.2.4. Arbitration.

Value	Description
0	Static arbitration scheme for channels with level 2 priority.
1	Round-robin arbitration scheme for channels with level 2 priority.

Bits 19:16 – LVLPRI2[3:0] Level 2 Channel Priority Number
When round-robin arbitration is enabled (RRLVLEN2=1) for priority level 2, this register holds the channel number of the last DMA channel being granted access as the active channel with priority level 2.
When static arbitration is enabled (RRLVLEN2=0) for priority level 2, 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 (RRLVLEN2 written to '0').

Bit 15 – RRLVLEN1 Level 1 Round-Robin Scheduling Enable

For details on arbitration schemes, refer to 27.5.2.4. Arbitration.

Value	Description
0	Static arbitration scheme for channels with level 1 priority.
1	Round-robin arbitration scheme for channels with level 1 priority.

Bits 11:8 – LVLPRI1[3:0] Level 1 Channel Priority Number

When round-robin arbitration is enabled (RRLVLEN1=1) for priority level 1, this register holds the channel number of the last DMA channel being granted access as the active channel with priority level 1.
When static arbitration is enabled (RRLVLEN1=0) for priority level 1, 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 (RRLVLEN1 written to '0').

Bit 7 – RRLVLEN0 Level 0 Round-Robin Scheduling Enable

For details on arbitration schemes, refer to 27.5.2.4. Arbitration.

Value	Description
0	Static arbitration scheme for channels with level 0 priority.
1	Round-robin arbitration scheme for channels with level 0 priority.

Bits 3:0 – LVLPRI0[3:0] Level 0 Channel Priority Number

When round-robin arbitration is enabled (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 (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 (RRLVLEN0 written to '0').

27.6.10 Interrupt Pending

Name: INTPEND
Offset: 0x20
Reset: 0x0000
Property: -

This register allows the user to identify the lowest DMA channel with pending interrupt.

Important: INTPEND is dedicated to channels which are multiplexed on a single interrupt line (Channels 4 to 15).

Bit	15	14	13	12	11	10	9	8
	PEND	BUSY	FERR			SUSP	TCMPL	TERR
Access	R	R	R			R/W	R/W	R/W
Reset	0	0	0			0	0	0

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

Bit 15 – PEND Pending
This bit will read '1' when the channel selected by Channel ID field (ID) is pending.

Bit 14 – BUSY Busy
This bit will read '1' when the channel selected by Channel ID field (ID) is busy.

Bit 13 – FERR Fetch Error
This bit will read '1' when the channel selected by Channel ID field (ID) fetched an invalid descriptor.

Bit 10 – SUSP Channel Suspend
This bit will read '1' when the channel selected by Channel ID field (ID) has pending Suspend interrupt.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Channel ID (ID) Suspend interrupt flag.

Bit 9 – TCMPL Transfer Complete
This bit will read '1' when the channel selected by Channel ID field (ID) has pending Transfer Complete interrupt.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Channel ID (ID) Transfer Complete interrupt flag.

Bit 8 – TERR Transfer Error
This bit is read one when the channel selected by Channel ID field (ID) has pending Transfer Error interrupt.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Channel ID (ID) Transfer Error interrupt flag.

Bits 3:0 – ID[3:0] Channel ID
These bits store the lowest channel number with pending interrupts. The number is valid if Suspend (SUSP), Transfer Complete (TCMPL) or Transfer Error (TERR) bits are set. The Channel ID field is refreshed when a new channel (with channel number less than the current one) with pending interrupts is detected, or when the application clears the corresponding channel interrupt sources. When no pending channels interrupts are available, these bits will always return zero value when read.
When the bits are written, indirect access to the corresponding Channel Interrupt Flag register is enabled.

27.6.11 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
	CHINT15	CHINT14	CHINT13	CHINT12	CHINT11	CHINT10	CHINT9	CHINT8
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
	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, 12, 13, 14, 15 – CHINT 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.

27.6.12 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
	BUSYCH15	BUSYCH14	BUSYCH13	BUSYCH12	BUSYCH11	BUSYCH10	BUSYCH9	BUSYCH8
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
	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, 12, 13, 14, 15 – BUSYCH 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.

27.6.13 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
	PENDCH15	PENDCH14	PENDCH13	PENDCH12	PENDCH11	PENDCH10	PENDCH9	PENDCH8
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
	PENDCH7	PENDCH6	PENDCH5	PENDCH4	PENDCH3	PENDCH2	PENDCH1	PENDCH0
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, 14, 15 – PENDCH Pending Channel n [n=15..0]
This bit is cleared when 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.

27.6.14 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
					LVLEX3	LVLEX2	LVLEX1	LVLEX0
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 – LVLEXx 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.

27.6.15 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.

27.6.16 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.

27.6.17 Channel ID

Name: CHID
Offset: 0x3F
Reset: 0x00
Property: -

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

Bits 3:0 – ID[3:0] Channel ID
These bits define the channel number that will be affected by the channel registers (CH*). Before reading or writing a channel register, the channel ID bit group must be written first.

27.6.18 Channel Control A

Name: CHCTRLA
Offset: 0x40
Reset: 0x00
Property: PAC Write-Protection, Enable-Protected

This register affects the DMA channel that is selected in the Channel ID register (CHID.ID).

Bit	7	6	5	4	3	2	1	0
		RUNSTDBY					ENABLE	SWRST
Access	R	R/W	R	R	R	R	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 6 – RUNSTDBY Channel run in standby
This bit is used to keep the DMAC channel running in standby mode.
This bit is not enable-protected.

Value	Description
0	The DMAC channel is halted in standby.
1	The DMAC channel continues to run in standby.

Bit 1 – ENABLE Channel Enable
Writing a '0' to this bit during an ongoing transfer, the bit will not be cleared until the internal data transfer buffer is empty and the DMA transfer is aborted. The internal data transfer buffer will be empty once the ongoing burst transfer is completed.
Writing a '1' to this bit will enable the DMA channel.
This bit is not enable-protected.

Value	Description
0	DMA channel is disabled.
1	DMA channel is enabled.

Bit 0 – SWRST Channel Software Reset
Writing a '0' to this bit has no effect.
Writing a '1' to this bit resets the channel registers to their initial state. The bit can be set when the channel is disabled (ENABLE=0). Writing a '1' to this bit will be ignored as long as ENABLE=1. This bit is automatically cleared when the reset is completed.

Value	Description
0	There is no reset operation ongoing.
1	The reset operation is ongoing.

27.6.19 Channel Control B

Name: CHCTRLB
Offset: 0x44
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

This register affects the DMA channel that is selected in the Channel ID register (CHID.ID).

Bit	31	30	29	28	27	26	25	24
							CMD[1:0]	
Access							R/W	R/W
Reset							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[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
		LVL[1:0]		EVOE	EVIE	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

Bits 25:24 – CMD[1:0] Software Command
These bits define the software commands. Refer to 27.5.3.2. Channel Suspend and 27.5.3.3. Channel Resume and Next Suspend Skip.
These bits are not enable-protected.

Value	Name	Description
0x0	NOACT	No action
0x1	SUSPEND	Channel suspend operation
0x2	RESUME	Channel resume operation
0x3	-	Reserved

Bits 23:22 – TRIGACT[1:0] Trigger Action
These bits define the trigger action used for a transfer.

Value	Name	Description
0x0	BLOCK	One trigger required for each block transfer
0x1	-	Reserved
0x2	BEAT	One trigger required for each beat transfer
0x3	TRANSACTION	One trigger required for each transaction

Bits 13:8 – TRIGSRC[5:0] Trigger Source
These bits define the peripheral trigger which is source of the transfer. For details on trigger selection and trigger modes, refer to Transfer Triggers and Actions and CHCTRLB.TRIGACT.

Value	Name	Description
0x00	DISABLE	Only software/event triggers
0x01	RTC_TIMESTAMP	RTC Timestamp Trigger
0x02	DSU_DCC0	ID for DCC0 register
0x03	DSU_DCC1	ID for DCC1 register

Value	Name	Description
0x04	SERCOM0_RX	SERCOM0 RX Trigger
0x05	SERCOM0_TX	SERCOM0 TX Trigger
0x06	SERCOM1_RX	SERCOM1 RX Trigger
0x07	SERCOM1_TX	SERCOM1 TX Trigger
0x08	SERCOM2_RX	SERCOM2 RX Trigger
0x09	SERCOM2_TX	SERCOM2 TX Trigger
0x0A	SERCOM3_RX	SERCOM3 RX Trigger
0x0B	SERCOM3_TX	SERCOM3 TX Trigger
0x0C	SERCOM4_RX	SERCOM4 RX Trigger
0x0D	SERCOM4_TX	SERCOM4 TX Trigger
0x0E	SERCOM5_RX	SERCOM5 RX Trigger
0x0F	SERCOM5_TX	SERCOM5 TX Trigger
0x10	TC0_OVF	TC0 Overflow Trigger
0x11	TC0_MC0	TC0 Match/Compare 0 Trigger
0x12	TC0_MC1	TC0 Match/Compare 1 Trigger
0x13	TC1_OVF	TC1 Overflow Trigger
0x14	TC1_MC0	TC1 Match/Compare 0 Trigger
0x15	TC1_MC1	TC1 Match/Compare 1 Trigger
0x16	TC2_OVF	TC2 Overflow Trigger
0x17	TC2_MC0	TC2 Match/Compare 0 Trigger
0x18	TC2_MC1	TC2 Match/Compare 1 Trigger
0x19	TCC0_OVF	TCC0 Overflow Trigger
0x1A	TCC0_MC0	TCC0 Match/Compare 0 Trigger
0x1B	TCC0_MC1	TCC0 Match/Compare 1 Trigger
0x1C	TCC0_MC2	TCC0 Match/Compare 2 Trigger
0x1D	TCC0_MC3	TCC0 Match/Compare 3 Trigger
0x1E	TCC1_OVF	TCC1 Overflow Trigger
0x1F	TCC1_MC0	TCC1 Match/Compare 0 Trigger
0x20	TCC1_MC1	TCC1 Match/Compare 1 Trigger
0x21	TCC2_OVF	TCC2 Overflow Trigger
0x22	TCC2_MC0	TCC2 Match/Compare 0 Trigger
0x23	TCC2_MC1	TCC2 Match/Compare 1 Trigger
0x24	TCC3_OVF	TCC3 Overflow Trigger
0x25	TCC3_MC0	TCC3 Match/Compare 0 Trigger
0x26	TCC3_MC1	TCC3 Match/Compare 1 Trigger
0x27	TCC3_MC2	TCC3 Match/Compare 2 Trigger
0x28	TCC3_MC3	TCC3 Match/Compare 3 Trigger
0x29	ADC_RESRDY	ADC Result Ready Trigger
0x2A	DAC_EMPTY0	DAC Empty 0 Trigger
0x2B	DAC_EMPTY1	DAC Empty 1 Trigger
0x2C	PTC_EOC	PTC End of Conversion Trigger
0x2D	PTC_SEQ	PTC Sequence Trigger
0x2E	PTC_WCOMP	PTC Window Compare Trigger
0x2F	I2S_RX0	I2S RX0 Trigger
0x30	I2S_RX1	I2S RX1 Trigger
0x31	I2S_TX0	I2S TX0 Trigger
0x32	I2S_TX1	I2S TX1 Trigger

Bits 6:5 – LVL[1:0] Channel Arbitration Level

These bits define the arbitration level used for the DMA channel, where a high level has priority over a low level. For further details on arbitration schemes, refer to 27.5.2.4. Arbitration.

These bits are not enable-protected.

Value	Name	Description
0x0	LVL0	Channel Priority Level 0

.....continued		
Value	**Name**	**Description**
0x1	LVL1	Channel Priority Level 1
0x2	LVL2	Channel Priority Level 2
0x3	LVL3	Channel Priority Level 3

Bit 4 – EVOE Channel Event Output Enable

This bit indicates if the Channel event generation is enabled. The event will be generated for every condition defined in the descriptor Event Output Selection (27.7.1. BTCTRL.EVOSEL).

This bit is available only for the 8 least significant DMA channels. Refer to table: *User Multiplexer Selection* and *Event Generator Selection* of the Event System for details.

Value	Description
0	Channel event generation is disabled.
1	Channel event generation is enabled.

Bit 3 – EVIE Channel Event Input Enable

This bit is available only for the 8 least significant DMA channels. Refer to table: *User Multiplexer Selection* and *Event Generator Selection* of the Event System for details.

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 2:0 – EVACT[2:0] Event Input Action

These bits define the event input action, as shown below. The action is executed only if the corresponding EVIE bit in the CHCTRLB register of the channel is set.

These bits are available only for the 8 least significant DMA channels. Refer to table: *User Multiplexer Selection* and *Event Generator Selection* of the Event System for details.

Value	Name	Description
0x0	NOACT	No action
0x1	TRIG	Normal Transfer and Conditional Transfer on Strobe 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	-	Reserved

27.6.20 Channel Interrupt Enable Clear

Name: CHINTENCLR
Offset: 0x4C
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.
This register affects the DMA channel that is selected in the Channel ID register (CHID.ID).

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.

27.6.21 Channel Interrupt Enable Set

Name: CHINTENSET
Offset: 0x4D
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.
This register affects the DMA channel that is selected in the Channel ID register (CHID.ID).

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.

27.6.22 Channel Interrupt Flag Status and Clear

Name: CHINTFLAG
Offset: 0x4E
Reset: 0x00
Property: -

This register affects the DMA channel that is selected in the Channel ID register (CHID.ID).

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, refer to CHCTRLB.CMD.
For details on available event input actions, refer to CHCTRLB.EVACT.
For details on available block actions, refer to BTCTRL.BLOCKACT.

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.

27.6.23 Channel Status

Name: CHSTATUS
Offset: 0x4F
Reset: 0x00
Property: -

This register affects the DMA channel that is selected in the Channel ID register (CHID.ID).

Bit	7	6	5	4	3	2	1	0
						FERR	BUSY	PEND
Access						R	R	R
Reset						0	0	0

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, refer to CHCTRLB.TRIGACT.
This bit is set when a transfer is pending on the DMA channel, as soon as the transfer request is received.

27.7 Register Summary - SRAM

Refer to the Registers Description section for more details on register properties and access permissions.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	BTCTRL	15:8	STEPSIZE[2:0]			STEPSEL	DSTINC	SRCINC	BEATSIZE[1:0]	
		7:0				BLOCKACT[1:0]		EVOSEL[1:0]		VALID
0x02	BTCNT	15:8	BTCNT[15:8]							
		7:0	BTCNT[7:0]							
0x04	SRCADDR	31:24	SRCADDR[31:24]							
		23:16	SRCADDR[23:16]							
		15:8	SRCADDR[15:8]							
		7:0	SRCADDR[7:0]							
0x08	DSTADDR	31:24	DSTADDR[31:24]							
		23:16	DSTADDR[23:16]							
		15:8	DSTADDR[15:8]							
		7:0	DSTADDR[7:0]							
0x0C	DESCADDR	31:24	DESCADDR[31:24]							
		23:16	DESCADDR[23:16]							
		15:8	DESCADDR[15:8]							
		7:0	DESCADDR[7:0]							

27.7.1 Block Transfer Control

Name: BTCTRL
Offset: 0x00
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	-	-	-	-	-	-	-	-
Reset								

Bit	7	6	5	4	3	2	1	0
				BLOCKACT[1:0]		EVOSEL[1:0]		VALID
Access				-	-	-	-	-
Reset								

Bits 15:13 – STEPSIZE[2:0] Address Increment Step Size
These bits select the address increment step size. The setting apply to source or destination address, depending on STEPSEL setting.

Value	Name	Description
0x0	X1	Next ADDR = ADDR + (Beat size in byte) * 1
0x1	X2	Next ADDR = ADDR + (Beat size in byte) * 2
0x2	X4	Next ADDR = ADDR + (Beat size in byte) * 4
0x3	X8	Next ADDR = ADDR + (Beat size in byte) * 8
0x4	X16	Next ADDR = ADDR + (Beat size in byte) * 16
0x5	X32	Next ADDR = ADDR + (Beat size in byte) * 32
0x6	X64	Next ADDR = ADDR + (Beat size in byte) * 64
0x7	X128	Next ADDR = ADDR + (Beat size in byte) * 128

Bit 12 – STEPSEL Step Selection
This bit selects if source or destination addresses are using the step size settings.

Value	Name	Description
0x0	DST	Step size settings apply to the destination address
0x1	SRC	Step size settings apply to the source address

Bit 11 – DSTINC Destination Address Increment Enable
Writing a '0' to this bit will disable the destination address incrementation. The address will be kept fixed during the data transfer.
Writing a '1' to this bit will enable the destination address incrementation. By default, the destination address is incremented by 1. If the STEPSEL bit is cleared, flexible step-size settings are available in the STEPSIZE register.

Value	Description
0	The Destination Address Increment is disabled.
1	The Destination Address Increment is enabled.

Bit 10 – SRCINC Source Address Increment Enable
Writing a '0' to this bit will disable the source address incrementation. The address will be kept fixed during the data transfer.
Writing a '1' to this bit will enable the source address incrementation. By default, the source address is incremented by 1. If the STEPSEL bit is set, flexible step-size settings are available in the STEPSIZE register.

Value	Description
0	The Source Address Increment is disabled.
1	The Source Address Increment is enabled.

Bits 9:8 – BEATSIZE[1:0] Beat Size

These bits define the size of one beat. A beat is the size of one data transfer bus access, and the setting apply to both read and write accesses.

Value	Name	Description
0x0	BYTE	8-bit bus transfer
0x1	HWORD	16-bit bus transfer
0x2	WORD	32-bit bus transfer
0x3	Reserved	-

Bits 4:3 – BLOCKACT[1:0] Block Action

These bits define what actions the DMAC should take after a block transfer has completed.

Value	Name	Description
0x0	NOACT	Channel will be disabled if it is the last block transfer in the transaction
0x1	INT	Channel will be disabled if it is the last block transfer in the transaction and block interrupt
0x2	SUSPEND	Channel suspend operation is completed
0x3	BOTH	Both channel suspend operation and block interrupt

Bits 2:1 – EVOSEL[1:0] Event Output Selection

These bits define the event output selection.

Value	Name	Description
0x0	DISABLE	Event generation disabled
0x1	BLOCK	Event strobe when block transfer complete
0x2	Reserved	-
0x3	BEAT	Event strobe when beat transfer complete

Bit 0 – VALID Descriptor Valid

Writing a '0' to this bit in the Descriptor or Write-Back memory will suspend the DMA channel operation when fetching the corresponding descriptor.

The bit is automatically cleared in the Write-Back memory section when channel is aborted, when an error is detected during the block transfer, or when the block transfer is completed.

Value	Description
0	The descriptor is not valid.
1	The descriptor is valid.

27.7.2 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	-	-	-	-	-	-	-	-
Reset								

Bit	7	6	5	4	3	2	1	0
	BTCNT[7:0]							
Access	-	-	-	-	-	-	-	-
Reset								

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.

27.7.3 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								

Bit	23	22	21	20	19	18	17	16
	SRCADDR[23:16]							
Access	-	-	-	-	-	-	-	-
Reset								

Bit	15	14	13	12	11	10	9	8
	SRCADDR[15:8]							
Access	-	-	-	-	-	-	-	-
Reset								

Bit	7	6	5	4	3	2	1	0
	SRCADDR[7:0]							
Access	-	-	-	-	-	-	-	-
Reset								

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:

$$\mathrm{SRCADDR} = \mathrm{SRCADDR}_{START} + BTCNT \cdot (BEATSIZE + 1)$$

- $\mathrm{SRCADDR}_{START}$ 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

27.7.4 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								

Bit	23	22	21	20	19	18	17	16
	DSTADDR[23:16]							
Access	-	-	-	-	-	-	-	-
Reset								

Bit	15	14	13	12	11	10	9	8
	DSTADDR[15:8]							
Access	-	-	-	-	-	-	-	-
Reset								

Bit	7	6	5	4	3	2	1	0
	DSTADDR[7:0]							
Access	-	-	-	-	-	-	-	-
Reset								

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

27.7.5 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 64-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.

28. External Interrupt Controller (EIC)

28.1 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, or both edges, or on high or low levels. Each external pin has a configurable filter to remove spikes. Each external pin can also be configured to be asynchronous in order to wake up the device from sleep modes where all clocks have been disabled. External pins can also generate an event. Each external pin can be defined as secured or non-secured, where secured pins can only be handled by secure accesses.

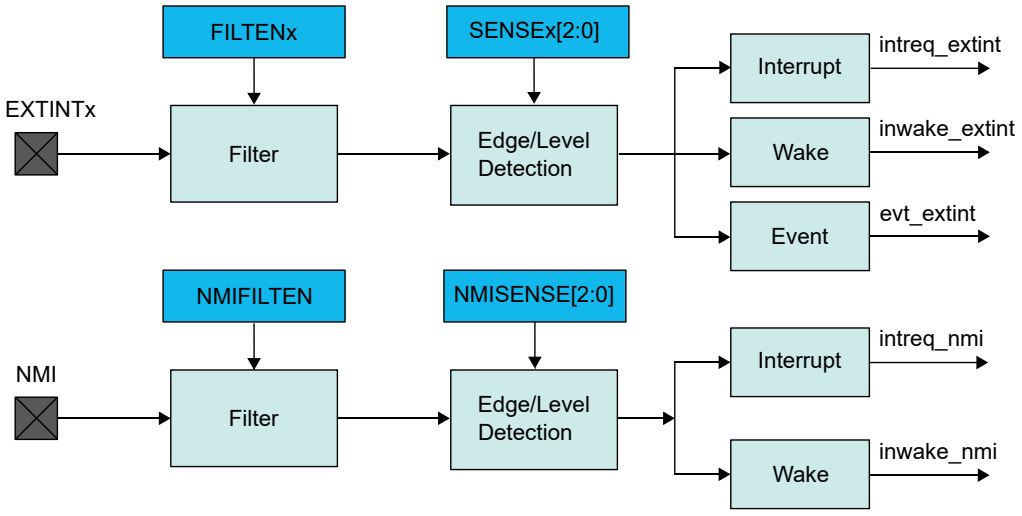
A separate non-maskable interrupt (NMI) is also supported. It has properties similar to the other external interrupts, but is connected to the NMI request of the CPU, enabling it to interrupt any other interrupt mode.

28.2 Features

- Up to 16 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
- Selectable secured or non-secured attribution for each individual external pin **(PIC32CM LS00/LS60)**

28.3 Block Diagram

Figure 28-1. EIC Block Diagram



28.4 Signal Description

Signal Name	Type	Description
EXTINT[15..0]	Digital Input	External interrupt pin
NMI	Digital Input	Non-maskable interrupt pin

One signal may be available on several pins.

28.5 Peripheral Dependencies

Table 28-1. EIC Configuration Summary

Peripheral name	Base address	NVIC IRQ Index	MCLK AHB/APB Clocks	GCLK Peripheral Channel Index (GCLK.PCHCTRL)	PAC Peripheral Identifier Index (PAC.WRCTRL)	Events (EVSYS)		Power Domain (PM.STDBYCFG)
						Users (EVSYS.USER)	Generators (EVSYS.CHANNEL)	
EIC	0x40002800	3: EXTINT0 4: EXTINT1 5: EXTINT2 6: EXTINT3 7: EXTINT4 8: EXTINT5 9: EXTINT6 10: EXTINT7 11: EXTINT8-15, NSCHK NMI	CLK_EIC_APB	4: GCLK_EIC	10	—	17-32: EXTINT0-15	PDAO

28.6 Functional Description

28.6.1 Principle of Operation

The EIC detects edge or level condition to generate interrupts to the CPU interrupt controller or events to the Event System. Each external interrupt pin (EXTINT) can be filtered using majority vote filtering, clocked by GCLK_EIC or by CLK_ULP32K.

28.6.2 Basic Operation

28.6.2.1 Initialization

The EIC must be initialized in the following order:

1. Enable CLK_EIC_APB
2. If required, configure the NMI by writing the Non-Maskable Interrupt Control register (28.7.2. NMICTRL)
3. Enable GCLK_EIC or CLK_ULP32K when one of the following configuration 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 32.768 kHz is required for filtering.

CLK_ULP32K is recommended when power consumption is the priority. For CLK_ULP32K write a '1' to the Clock Selection bit in the Control A register (CTRLA.CKSEL).

4. Configure the EIC input sense and filtering by writing the Configuration n register (CONFIG).
5. Optionally, enable the asynchronous mode.
6. Optionally, enable the debouncer mode.
7. Enable the EIC by writing a '1' to CTRLA.ENABLE.

The following bits are enable-protected, meaning that it can only be written when the EIC is disabled (28.7.1. CTRLA.ENABLE=0):

- Clock Selection bit in Control A register (28.7.1. CTRLA.CKSEL)

The following registers are enable-protected:

- Event Control register (28.7.5. EVCTRL)
- Configuration n register (CONFIG).
- External Interrupt Asynchronous Mode register (28.7.9. ASYNCH)
- Debouncer Enable register (28.7.12. DEBOUNCEN)
- Debounce Prescaler register (28.7.13. DPRESCALER)

Enable-protected bits in the 28.7.1. CTRLA register can be written at the same time when setting 28.7.1. CTRLA.ENABLE to '1', but not at the same time as 28.7.1. CTRLA.ENABLE is being cleared.

Enable-protection is denoted by the "Enable-Protected" property in the register description.

28.6.2.2 Enabling, Disabling, and Resetting

The EIC is enabled by writing a '1' to the Enable bit in the Control A register (28.7.1. CTRLA.ENABLE). The EIC is disabled by writing 28.7.1. CTRLA.ENABLE to '0'.

The EIC is reset by setting the Software Reset bit in the Control register (28.7.1. CTRLA.SWRST). All registers in the EIC will be reset to their initial state, and the EIC will be disabled.

Refer to the 28.7.1. CTRLA register description for details.

28.6.3 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 (CONFIGn.SENSEx). The corresponding interrupt flag (INTFLAG.EXTINT[x]) in the Interrupt Flag Status and Clear register (28.7.8. 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 interrupt has been cleared, INTFLAG.EXTINT[x] will be set immediately if the EXTINTx pin still matches the interrupt condition.

Each external pin can be filtered by a majority vote filtering, clocked by GCLK_EIC or CLK_ULP32K. Filtering is enabled if bit Filter Enable x in the Configuration n register (CONFIGn.FILTENx) is written to '1'. The majority vote filter samples the external pin three times with GCLK_EIC or CLK_ULP32K and outputs the value when two or more samples are equal.

Table 28-2. Majority Vote Filter

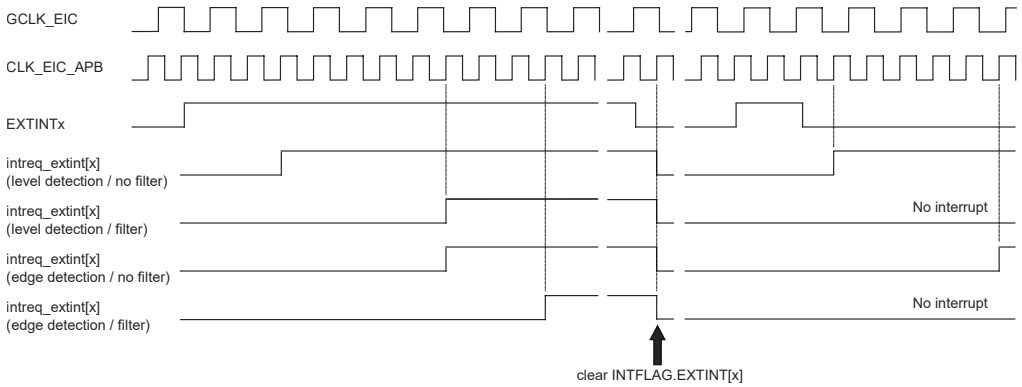
Samples [0, 1, 2]	Filter Output
[0,0,0]	0
[0,0,1]	0
[0,1,0]	0
[0,1,1]	1
[1,0,0]	0

.....continued	
Samples [0, 1, 2]	**Filter Output**
[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 CLK_ULP32K, 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 CLK_ULP32K to operate. The selection between these two clocks is done by writing the Clock Selection bits in the Control A register (28.7.1. CTRLA.CKSEL). GCLK_EIC must be enabled in the GCLK module. 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 28-2. Interrupt Detection Latency by modes (Rising Edge)



The detection latency depends on the detection mode.

Table 28-3. Detection Latency

Detection mode	Latency (worst case)
Level without filter	Five CLK_EIC_APB periods
Level with filter	Four GCLK_EIC/CLK_ULP32K periods + five CLK_EIC_APB periods
Edge without filter	Four GCLK_EIC/CLK_ULP32K periods + five CLK_EIC_APB periods
Edge with filter	Six GCLK_EIC/CLK_ULP32K periods + five CLK_EIC_APB periods

28.6.4 Additional Features

28.6.4.1 Non-Maskable Interrupt (NMI)

The non-maskable interrupt pin can also generate an interrupt on edge or level detection, but it is configured with the dedicated NMI Control register (NMICTRL). To select the sense for NMI, write to the NMISENSE bit group in the NMI Control register (NMICTRL.NMISENSE). NMI filtering is enabled by writing a '1' to the NMI Filter Enable bit (NMICTRL.NMIFILTEN).

If edge detection or filtering is required, enable GCLK_EIC or CLK_ULP32K.

NMI detection is enabled only by the NMICTRL.NMISENSE value, and the EIC 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.

The NMI interrupt does not require the interrupt controller to be configured.

28.6.4.2 Asynchronous Edge Detection Mode (No Debouncing)

The EXTINT edge detection can be operated synchronously or asynchronously, selected by the Asynchronous Control Mode bit for external pin x in the External Interrupt Asynchronous Mode register (28.7.9. ASYNCH.ASYNCH[x]). The EIC edge detection is operated synchronously when the Asynchronous Control Mode bit (ASYNCH.ASYNCH[x]) is '0' (default value). It is operated asynchronously when ASYNCH.ASYNCH[x] is written to '1'.

In *Synchronous Edge Detection Mode*, the external interrupt (EXTINT) or the non-maskable interrupt (NMI) pins are sampled using the EIC clock as defined by the Clock Selection bit in the Control A register (28.7.1. 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. In this mode, the EIC clock is required.

The Synchronous Edge Detection Mode can be used in Idle and Standby sleep modes.

In *Asynchronous Edge Detection Mode*, the external interrupt (EXTINT) pins or the non-maskable interrupt (NMI) pins set the External Interrupt flag or Non-Maskable Interrupt flag (INTFLAG.EXTINT[x] or NMIFLAG) directly. In this mode, the EIC clock is not requested.

The asynchronous edge detection mode can be used in Idle and Standby sleep modes.

28.6.4.3 Interrupt Pin Debouncing

The external interrupt pin (EXTINT) edge detection can use a debouncer to improve input noise immunity. When selected, the debouncer can work in the synchronous mode or the asynchronous mode, depending on the configuration of the ASYNCH.ASYNCH[x] bit for the pin. The debouncer uses the EIC clock as defined by the bit CTRLA.CKSEL to clock the debouncing circuitry. The debouncing time frame is set with the debouncer prescaler DPRESCALER.DPRESCALERn, 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 (CONFIGn.SENSEx) are set to RISE, FALL or BOTH. If the debouncing mode for pin EXTINT x is selected, the filter mode for that pin (CONFIGn.FILTENx) can not be selected.

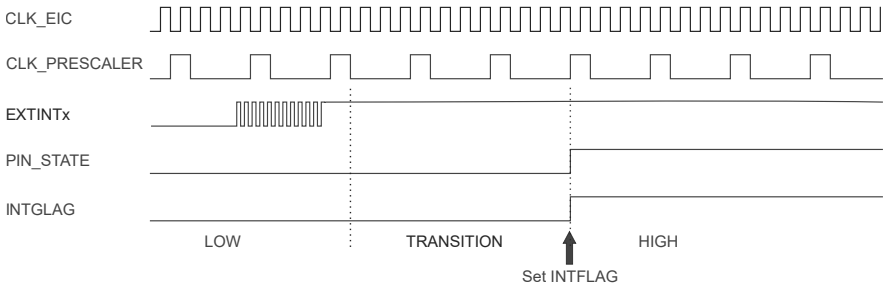
The debouncer manages an internal “valid pin state” that depends on the external interrupt (EXTINT) pin transitions, the debouncing mode and the debouncer prescaler frequency. The valid pin state reflects the pin value after debouncing. The external interrupt pin (EXTINT) is sampled continously on EIC clock. The sampled value is evaluated on each *low frequency clock* tick to detect a transitional edge when the sampled value is different of the current valid pin state. The sampled value is evaluated on each EIC clock when 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 bitfield. 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 both synchronous or asynchronous debouncing mode.

Synchronous edge detection In this mode the external interrupt (EXTINT) pin is sampled continously on EIC clock.

1. A pin edge transition will be validated when the sampled value is consistently different of the current valid pin state for 4 (or 8 depending on bit 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 will increment the transition counter.
3. Any pin sample, at 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 will return the transition counter to zero.
4. When the transition counter meets the count threshold, the pin edge transition is validated and the pin state PINSTATE.PINSTATE[x] is changed to the detected level.
5. The external interrupt flag (INTFLAG.EXTINT[x]) is set when the pin state PINSTATE.PINSTATE[x] is changed.

Figure 28-3. EXTINT Pin Synchronous Debouncing (Rising Edge)

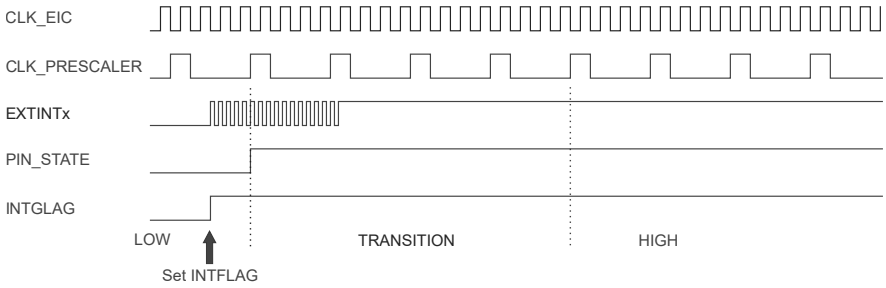


In the synchronous edge detection mode, the EIC clock is required. The synchronous edge detection mode can be used in Idle and Standby sleep modes.

Asynchronous edge detection In this mode, the external interrupt (EXTINT) pin directly drives an asynchronous edges detector which triggers any rising or falling edge on the pin:

1. Any edge detected that indicates a transition from the current valid pin state will immediately set the valid pin state PINSTATE.PINSTATE[x] to the detected level.
2. The external interrupt flag (INTFLAG.EXTINT[x] is immediately changed.
3. The edge detector will then be idle until no other rising or falling edge transition is detected during 4 consecutive ticks of the low frequency clock.
4. Any rising or falling edge transition detected during the idle state will return the transition counter to 0.
5. After 4 consecutive ticks of the low frequency clock without bounce detected, the edge detector is ready for a new detection.

Figure 28-4. 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.

28.6.5 Interrupts

The EIC has the following interrupt sources:

- External interrupt pins (EXTINTx). See 28.6.2. Basic Operation.
- Non-maskable interrupt pin (NMI). See 28.6.4. Additional Features.
- Non-secure check interrupt pin (NSCHK). See 28.7.8. INTFLAG

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).

An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until the interrupt flag is cleared, the interrupt is disabled, or the EIC is reset. See the INTFLAG register for details on how to clear interrupt flags. The EIC has at least one common interrupt request line for all the interrupt sources, and one interrupt request line for the NMI. The user must read the INTFLAG (or NMIFLAG) register to determine which interrupt condition is present.

Note: Interrupts must be globally enabled for interrupt requests to be generated.

For more information, refer to11. Processor and Architecture.

28.6.6 Events

The EIC can generate the following output events:

- External event from pin (EXTINT0-15).

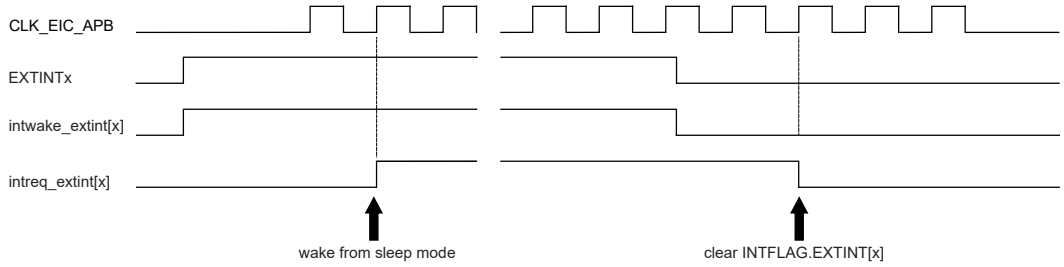
Setting an Event Output Control register (EVCTRL.EXTINTEO) enables the corresponding output event. Clearing this bit disables the corresponding output event. Refer to Event System for details on configuring the Event System.

When the condition on pin EXTINTx matches the configuration in the CONFIGn register, the corresponding event is generated, if enabled.

28.6.7 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 (28.7.7. INTENSET) is written to '1'.

Figure 28-5. Wake-up Operation Example (High-Level Detection, No Filter, Interrupt Enable Set)



28.6.8 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.

28.6.9 PIC32CM LS00/LS60 Secure Access Rights

Non-secure write to CTRLA register or DPRESCALER register is prohibited. Non-secure read to CTRLA or DPRESCALER register or SYNCBUSY register will return zero with no error resulting. Non-secure write to a bit of EVCTRL, ASYNCH, DEBOUNCEN, INTENCLR, INTENSET, INTFLAG and CONFIG registers is prohibited if the related bit of NONSEC.EXTINT is zero. Non-secure write to NMICTRL and NMIFLAG registers is prohibited if NONSECNMI.NMI is zero. Bits relating to secure EXTINT read as zero in non-secure mode with no error resulting.

28.6.10 Synchronization

Some registers (or bit fields within a register) require synchronization when read and/or written.

Synchronization is denoted by the "Read-Synchronized" (or "Read-Synchronized Bits”) and/or "Write-Synchronized" (or "Write-Synchronized Bits”) property in each individual register description.

For more details, refer to Register Synchronization.

28.7 Register Summary

Refer to the Registers Description section for more details on register properties and access permissions.

Important: (PIC32CM LS00 only)
The peripheral register map is automatically duplicated in a Secure and Non-Secure alias:

- The Non-Secure alias is at the peripheral base address
- The Secure alias is located at the peripheral base address + 0x200

Refer to Mix-Secure Peripherals for more information on register access rights.

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	31:24								
		23:16								
		15:8								
		7:0							ENABLE	SWRST
0x08	EVCTRL	31:24								
		23:16								
		15:8	EXTINTEO15	EXTINTEO14	EXTINTEO13	EXTINTEO12	EXTINTEO11	EXTINTEO10	EXTINTEO9	EXTINTEO8
		7:0	EXTINTEO7	EXTINTEO6	EXTINTEO5	EXTINTEO4	EXTINTEO3	EXTINTEO2	EXTINTEO1	EXTINTEO0
0x0C	INTENCLR	31:24	NSCHK							
		23:16								
		15:8	EXTINT15	EXTINT14	EXTINT13	EXTINT12	EXTINT11	EXTINT10	EXTINT9	EXTINT8
		7:0	EXTINT7	EXTINT6	EXTINT5	EXTINT4	EXTINT3	EXTINT2	EXTINT1	EXTINT0
0x10	INTENSET	31:24	NSCHK							
		23:16								
		15:8	EXTINT15	EXTINT14	EXTINT13	EXTINT12	EXTINT11	EXTINT10	EXTINT9	EXTINT8
		7:0	EXTINT7	EXTINT6	EXTINT5	EXTINT4	EXTINT3	EXTINT2	EXTINT1	EXTINT0
0x14	INTFLAG	31:24	NSCHK							
		23:16								
		15:8	EXTINT15	EXTINT14	EXTINT13	EXTINT12	EXTINT11	EXTINT10	EXTINT9	EXTINT8
		7:0	EXTINT7	EXTINT6	EXTINT5	EXTINT4	EXTINT3	EXTINT2	EXTINT1	EXTINT0
0x18	ASYNCH	31:24								
		23:16								
		15:8	ASYNCH15	ASYNCH14	ASYNCH13	ASYNCH12	ASYNCH11	ASYNCH10	ASYNCH9	ASYNCH8
		7:0	ASYNCH7	ASYNCH6	ASYNCH5	ASYNCH4	ASYNCH3	ASYNCH2	ASYNCH1	ASYNCH0
0x1C	CONFIG0	31:24	FILTEN7	SENSE7[2:0]			FILTEN6	SENSE6[2:0]		
		23:16	FILTEN5	SENSE5[2:0]			FILTEN4	SENSE4[2:0]		
		15:8	FILTEN3	SENSE3[2:0]			FILTEN2	SENSE2[2:0]		
		7:0	FILTEN1	SENSE1[2:0]			FILTEN0	SENSE0[2:0]		
0x20	CONFIG1	31:24	FILTEN15	SENSE15[2:0]			FILTEN14	SENSE14[2:0]		
		23:16	FILTEN13	SENSE13[2:0]			FILTEN12	SENSE12[2:0]		
		15:8	FILTEN11	SENSE11[2:0]			FILTEN10	SENSE10[2:0]		
		7:0	FILTEN9	SENSE9[2:0]			FILTEN8	SENSE8[2:0]		
0x24 ... 0x2F	Reserved									
0x30	DEBOUNCEN	31:24								
		23:16								
		15:8	DEBOUNCEN 15	DEBOUNCEN 14	DEBOUNCEN 13	DEBOUNCEN 12	DEBOUNCEN 11	DEBOUNCEN 10	DEBOUNCEN 9	DEBOUNCEN 8
		7:0	DEBOUNCEN 7	DEBOUNCEN 6	DEBOUNCEN 5	DEBOUNCEN 4	DEBOUNCEN 3	DEBOUNCEN 2	DEBOUNCEN 1	DEBOUNCEN 0

.....continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x34	DPRESCALER	31:24								
		23:16								TICKON
		15:8								
		7:0	STATES1	PRESCALER1[2:0]			STATES0	PRESCALER0[2:0]		
0x38	PINSTATE	31:24								
		23:16								
		15:8	PINSTATE15	PINSTATE14	PINSTATE13	PINSTATE12	PINSTATE11	PINSTATE10	PINSTATE9	PINSTATE8
		7:0	PINSTATE7	PINSTATE6	PINSTATE5	PINSTATE4	PINSTATE3	PINSTATE2	PINSTATE1	PINSTATE0
0x3C	NSCHK	31:24	NMI							
		23:16								
		15:8	EXTINT15	EXTINT14	EXTINT13	EXTINT12	EXTINT11	EXTINT10	EXTINT9	EXTINT8
		7:0	EXTINT7	EXTINT6	EXTINT5	EXTINT4	EXTINT3	EXTINT2	EXTINT1	EXTINT0
0x40	NONSEC	31:24	NMI							
		23:16								
		15:8	EXTINT15	EXTINT14	EXTINT13	EXTINT12	EXTINT11	EXTINT10	EXTINT9	EXTINT8
		7:0	EXTINT7	EXTINT6	EXTINT5	EXTINT4	EXTINT3	EXTINT2	EXTINT1	EXTINT0

28.7.1 Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00
Property: PAC Write-Protection, Enable-Protected Bits, Write-Synchronized Bits, Secure

Bit	7	6	5	4	3	2	1	0
				CKSEL			ENABLE	SWRST
Access				RW/-/RW			RW/-/RW	W/-/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 CLK_ULP32K (when power consumption is the priority).
Note: This bit is enable-protected. This bit is not synchronized.

Value	Description
0	The EIC is clocked by GCLK_EIC.
1	The EIC is clocked by CLK_ULP32K.

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 will be set (SYNCBUSY.ENABLE=1). SYNCBUSY.ENABLE will be cleared when the operation is complete.
Notes:

1. This bit is write-synchronized: SYNCBUSY.ENABLE must be checked to ensure the CTRLA.ENABLE synchronization is complete.
2. This bit is not enable-protected.

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 will be disabled.
Writing a '1' to CTRLA.SWRST will always take precedence, meaning that all other writes in the same write operation will be discarded.
Notes:

1. When the CTRLA.SWRST is written, the user must poll SYNCBUSY.SWRST bit to know when the reset operation is complete.
2. During a SWRST, access to registers/bits without SWRST are disallowed until SYNCBUSY.SWRST is cleared by hardware.
3. This bit is not enable protected.

Value	Description
0	There is no ongoing reset operation.
1	The reset operation is ongoing.

28.7.2 Non-Maskable Interrupt Control

Name: NMICTRL
Offset: 0x01
Reset: 0x00
Property: PAC Write-Protection, Mix-Secure

Important: For **PIC32CM LS00/LS60 Non-Secure** accesses, read and write accesses (RW*) are allowed only if the NMI interrupt is set as Non-Secure in the NONSEC register (NONSEC.NMI bit).

Bit	7	6	5	4	3	2	1	0
				NMIASYNCH	NMIFILTEN	NMISENSE[2:0]		
Access				RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW
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.

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 on which edge or level the NMI triggers.

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

28.7.3 Non-Maskable Interrupt Flag Status and Clear

Name: NMIFLAG
Offset: 0x2
Reset: 0x00
Property: Mix-Secure

Important: For **PIC32CM LS00/LS60 Non-Secure** accesses, read and write accesses (RW*) are allowed only if the NMI interrupt is set as Non-Secure in the NONSEC register (NONSEC.NMI bit).

Bit	7	6	5	4	3	2	1	0
								NMI
Access								RW/RW*/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 will generate an interrupt request.
Writing a '0' to this bit has no effect.

28.7.4 Synchronization Busy

Name: SYNCBUSY
Offset: 0x04
Reset: 0x00000000
Property: Secure

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	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.

28.7.5 Event Control

Name: EVCTRL
Offset: 0x08
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected, Mix-Secure

Important: For **PIC32CM LS00/LS60 Non-Secure** accesses, read and write accesses (RW*) are allowed only if the external interrupt x (EXTINTx) is set as Non-Secure in the NONSEC register (NONSEC.EXTINTx bit). Some restrictions apply for the Non-Secure accesses to an Enable-Protected register as it will not be possible for the Non-Secure to configure it once this register is enabled by the Secure application. This will require some veneers to be implemented on Secure side.

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
	EXTINTEO15	EXTINTEO14	EXTINTEO13	EXTINTEO12	EXTINTEO11	EXTINTEO10	EXTINTEO9	EXTINTEO8
Access	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/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
	EXTINTEO7	EXTINTEO6	EXTINTEO5	EXTINTEO4	EXTINTEO3	EXTINTEO2	EXTINTEO1	EXTINTEO0
Access	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW
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 – EXTINTEO 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.

28.7.6 Interrupt Enable Clear

Name: INTENCLR
Offset: 0x0C
Reset: 0x00000000
Property: PAC Write-Protection, Mix-Secure

Important: For **PIC32CM LS00/LS60 Non-Secure** accesses, read and write accesses (RW*) are allowed only if the external interrupt x (EXTINTx) is set as Non-Secure in the NONSEC register (NONSEC.EXTINTx bit).

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
	NSCHK							
Access	RW/RW/RW							
Reset	0							

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	EXTINT15	EXTINT14	EXTINT13	EXTINT12	EXTINT11	EXTINT10	EXTINT9	EXTINT8
Access	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/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
	EXTINT7	EXTINT6	EXTINT5	EXTINT4	EXTINT3	EXTINT2	EXTINT1	EXTINT0
Access	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW
Reset	0	0	0	0	0	0	0	0

Bit 31 – NSCHK Non-secure Check Interrupt Enable
Note: This bit field is only available for **PIC32CM LS00/LS60** and has no effect for **PIC32CM LE00**.

Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the NSCHK Interrupt Enable bit.

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 – EXTINT 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.

Value	Description
0	The external interrupt x is disabled.
1	The external interrupt x is enabled.

28.7.7 Interrupt Enable Set

Name: INTENSET
Offset: 0x10
Reset: 0x00000000
Property: PAC Write-Protection, Mix-Secure

Important: For **PIC32CM LS00/LS60 Non-Secure** accesses, read and write accesses (RW*) are allowed only if the external interrupt x (EXTINTx) is set as Non-Secure in the NONSEC register (NONSEC.EXTINTx bit).

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
	NSCHK							
Access	RW/RW/RW							
Reset	0							

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	EXTINT15	EXTINT14	EXTINT13	EXTINT12	EXTINT11	EXTINT10	EXTINT9	EXTINT8
Access	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/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
	EXTINT7	EXTINT6	EXTINT5	EXTINT4	EXTINT3	EXTINT2	EXTINT1	EXTINT0
Access	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW
Reset	0	0	0	0	0	0	0	0

Bit 31 – NSCHK Non-secure Check Interrupt Enable

Note: This bit field is only available for **PIC32CM LS00/LS60** and has no effect for **PIC32CM LE00**.

Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the NSCHK Interrupt Enable bit.

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 – EXTINT 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.

28.7.8 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x14
Reset: 0x00000000
Property: Mix-Secure

Important: For **PIC32CM LS00/LS60 Non-Secure** accesses, read and write accesses (RW*) are allowed only if the external interrupt x (EXTINTx) is set as Non-Secure in the NONSEC register (NONSEC.EXTINTx bit).

Bit	31	30	29	28	27	26	25	24
	NSCHK							
Access	RW/RW/RW							
Reset	0							

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	EXTINT15	EXTINT14	EXTINT13	EXTINT12	EXTINT11	EXTINT10	EXTINT9	EXTINT8
Access	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/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
	EXTINT7	EXTINT6	EXTINT5	EXTINT4	EXTINT3	EXTINT2	EXTINT1	EXTINT0
Access	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW
Reset	0	0	0	0	0	0	0	0

Bit 31 – NSCHK Non-secure Check Interrupt

Note: This bit field is only available for **PIC32CM LS00/LS60** and has no effect for **PIC32CM LE00**.

The flag is cleared by writing a '1' to it. This flag is set when write to either NONSEC and NSCHK register and if the related bit of NSCHK is enabled and the related bit of NONSEC is zero.

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 – EXTINT 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 INTENSET.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.

28.7.9 External Interrupt Asynchronous Mode

Name: ASYNCH
Offset: 0x18
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected, Mix-Secure

Important: For **PIC32CM LS00/LS60 Non-Secure** accesses, read and write accesses (RW*) are allowed only if the external interrupt x (EXTINTx) is set as Non-Secure in the NONSEC register (NONSEC.EXTINTx bit). Some restrictions apply for the Non-Secure accesses to an Enable-Protected register as it will not be possible for the Non-Secure to configure it once this register is enabled by the Secure application. This will require some veneers to be implemented on Secure side.

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
	ASYNCH15	ASYNCH14	ASYNCH13	ASYNCH12	ASYNCH11	ASYNCH10	ASYNCH9	ASYNCH8
Access	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/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
	ASYNCH7	ASYNCH6	ASYNCH5	ASYNCH4	ASYNCH3	ASYNCH2	ASYNCH1	ASYNCH0
Access	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW
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 – ASYNCH 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.

28.7.10 External Interrupt Sense Configuration n

Name: CONFIG0
Offset: 0x1C
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected, Mix-Secure

Important: For **PIC32CM LS00/LS60 Non-Secure** accesses, read and write accesses (RW*) are allowed only if the external interrupt x (EXTINTx) is set as Non-Secure in the NONSEC register (NONSEC.EXTINTx bit). Some restrictions apply for the Non-Secure accesses to an Enable-Protected register as it will not be possible for the Non-Secure to configure it once this register is enabled by the Secure application. This will require some veneers to be implemented on the Secure side.

Bit	31	30	29	28	27	26	25	24
	FILTEN7	SENSE7[2:0]			FILTEN6	SENSE6[2:0]		
Access	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	FILTEN5	SENSE5[2:0]			FILTEN4	SENSE4[2:0]		
Access	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/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
	FILTEN3	SENSE3[2:0]			FILTEN2	SENSE2[2:0]		
Access	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/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*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW
Reset	0	0	0	0	0	0	0	0

Bits 3, 7, 11, 15, 19, 23, 27, 31 – FILTEN Filter Enable x [x=7..0]

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, 16:18, 20:22, 24:26, 28:30 – SENSE Input Sense Configuration x [x=7..0]
These bits define on which edge or level the interrupt or event for EXTINT[x] will be generated.

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

28.7.11 External Interrupt Sense Configuration n

Name: CONFIG1
Offset: 0x20
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected, Mix-Secure

Important: For **PIC32CM LS00/LS60 Non-Secure** accesses, read and write accesses (RW*) are allowed only if the external interrupt x (EXTINTx) is set as Non-Secure in the NONSEC register (NONSEC.EXTINTx bit). Some restrictions apply for the Non-Secure accesses to an Enable-Protected register as it will not be possible for the Non-Secure to configure it once this register is enabled by the Secure application. This will require some veneers to be implemented on the Secure side.

Bit	31	30	29	28	27	26	25	24
	FILTEN15	SENSE15[2:0]			FILTEN14	SENSE14[2:0]		
Access	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	FILTEN13	SENSE13[2:0]			FILTEN12	SENSE12[2:0]		
Access	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/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
	FILTEN11	SENSE11[2:0]			FILTEN10	SENSE10[2:0]		
Access	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/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
	FILTEN9	SENSE9[2:0]			FILTEN8	SENSE8[2:0]		
Access	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW
Reset	0	0	0	0	0	0	0	0

Bits 3, 7, 11, 15, 19, 23, 27, 31 – FILTEN Filter Enable x [x=15..8]

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, 16:18, 20:22, 24:26, 28:30 – SENSE Input Sense Configuration x [x=15..8]
These bits define on which edge or level the interrupt or event for EXTINT[x] will be generated.

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

28.7.12 Debouncer Enable

Name: DEBOUNCEN
Offset: 0x30
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected, Mix-Secure

Important: For **PIC32CM LS00/LS60 Non-Secure** accesses, read and write accesses (RW*) are allowed only if the external interrupt x (EXTINTx) is set as Non-Secure in the NONSEC register (NONSEC.EXTINTx bit). Some restrictions apply for the Non-Secure accesses to an Enable-Protected register as it will not be possible for the Non-Secure to configure it once this register is enabled by the Secure application. This will require some veneers to be implemented on Secure side.

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
	DEBOUNCEN15	DEBOUNCEN14	DEBOUNCEN13	DEBOUNCEN12	DEBOUNCEN11	DEBOUNCEN10	DEBOUNCEN9	DEBOUNCEN8
Access	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/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
	DEBOUNCEN7	DEBOUNCEN6	DEBOUNCEN5	DEBOUNCEN4	DEBOUNCEN3	DEBOUNCEN2	DEBOUNCEN1	DEBOUNCEN0
Access	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW
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 – DEBOUNCEN 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.

28.7.13 Debouncer Prescaler

Name: DPRESCALER
Offset: 0x34
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected, Secure

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
								TICKON
Access								RW/-/RW
Reset								0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	STATES1	PRESCALER1[2:0]			STATES0	PRESCALER0[2:0]		
Access	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW
Reset	0	0	0	0	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.

Bits 3, 7 – STATESx Debouncer number of states x

This bit selects the number of samples by the debouncer low frequency clock needed to validate a transition from current pin state to next pin state in synchronous debouncing mode for pins EXTINT[7+(8x):8x].

Value	Description
0	The number of low frequency samples is 3.
1	The number of low frequency samples is 7.

Bits 0:2, 4:6 – PRESCALERx Debouncer Prescaler x

These bits select the debouncer low frequency clock for pins EXTINT[7+(8x):8x].

Value	Name	Description
0x0	DIV2	EIC clock divided by 2
0x1	DIV4	EIC clock divided by 4
0x2	DIV8	EIC clock divided by 8
0x3	DIV16	EIC clock divided by 16
0x4	DIV32	EIC clock divided by 32
0x5	DIV64	EIC clock divided by 64
0x6	DIV128	EIC clock divided by 128
0x7	DIV256	EIC clock divided by 256

28.7.14 Pin State

Name: PINSTATE
Offset: 0x38
Reset: 0x00000000
Property: PAC Mix-Secure

Important: For **PIC32CM LS00/LS60 Non-Secure** accesses, read accesses (R*) are allowed only if the external interrupt x (EXTINTx) is set as Non-Secure in the NONSEC register (NONSEC.EXTINTx 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
	PINSTATE15	PINSTATE14	PINSTATE13	PINSTATE12	PINSTATE11	PINSTATE10	PINSTATE9	PINSTATE8
Access	R/R*/R	R/R*/R	R/R*/R	R/R*/R	R/R*/R	R/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
	PINSTATE7	PINSTATE6	PINSTATE5	PINSTATE4	PINSTATE3	PINSTATE2	PINSTATE1	PINSTATE0
Access	R/R*/R	R/R*/R	R/R*/R	R/R*/R	R/R*/R	R/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, 14, 15 – PINSTATE Pin State
These bits return the valid pin state of the debounced external interrupt pin EXTINTx.

28.7.15 Security Attribution Check

Name: NSCHK
Offset: 0x3C
Reset: 0x00000000
Property: PAC Write-Protection

This register allows the user to select one or more external pins to check their security attribution as non-secured.

Important: This register is only available for **PIC32CM LS00/LS60** and has no effect for **PIC32CM LE00**.

Bit	31	30	29	28	27	26	25	24
	NMI							
Access	RW/RW/RW							
Reset	0							

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	EXTINT15	EXTINT14	EXTINT13	EXTINT12	EXTINT11	EXTINT10	EXTINT9	EXTINT8
Access	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/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
	EXTINT7	EXTINT6	EXTINT5	EXTINT4	EXTINT3	EXTINT2	EXTINT1	EXTINT0
Access	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/RW/RW
Reset	0	0	0	0	0	0	0	0

Bit 31 – NMI Non-Maskable Interrupt Security Attribution Check
This bit selects the Non-Maskable Interrupt pin for security attribution check. If the NMI bit in NONSECNMI is set to the opposite value, then the NSCHK interrupt flag will be set.

Value	Description
0	0-to-1 transition will be detected on corresponding NONSEC bit.
1	1-to-0 transition will be detected on corresponding NONSEC bit.

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 – EXTINT External Interrupts Security Attribution Check
These bits select the individual pins for security attribution check. If any pin selected in NSCHK has the corresponding bit in NONSEC set to the opposite value, then the NSCHK interrupt flag will be set.

Value	Description
0	0-to-1 transition will be detected on corresponding NONSEC bit.
1	1-to-0 transition will be detected on corresponding NONSEC bit.

28.7.16 Non-secure Interrupt

Name: NONSEC
Offset: 0x40
Reset: 0x00000000
Property: PAC Write-Protection, Write-Secure

This register allows to set the NMI or external interrupt control and status registers in non-secure mode, individually per interrupt pin.

Important: This register is only available for **PIC32CM LS00/LS60** and has no effect for **PIC32CM LE00**.

Bit	31	30	29	28	27	26	25	24
	NMI							
Access	RW/R/RW							
Reset	0							

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	EXTINT15	EXTINT14	EXTINT13	EXTINT12	EXTINT11	EXTINT10	EXTINT9	EXTINT8
Access	RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	EXTINT7	EXTINT6	EXTINT5	EXTINT4	EXTINT3	EXTINT2	EXTINT1	EXTINT0
Access	RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW
Reset	0	0	0	0	0	0	0	0

Bit 31 – NMI Non-Secure Non-Maskable Interrupt
This bit enables the non-secure mode of NMI.
The registers whose content is set in non-secure mode by NONSEC.NMI are NMICTRL and NMIFLAG registers.

Value	Description
0	NMI is secure.
1	NMI is non-secure.

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 – EXTINT Non-Secure External Interrupt
The bit x of EXTINT enables the non-secure mode of EXTINTx.
The registers whose EXTINT bit or bitfield x is set in non-secure mode by NONSEC.EXTINTx are EVCTRL, ASYNCH, IDEBOUNCEN, NTENCLR, INTENSET, INTFLAG and CONFIG registers.

Value	Description
0	EXTINTx is secure.
1	EXTINTx is non-secure.

29. Non-volatile Memory Controller (NVMCTRL)

29.1 Overview

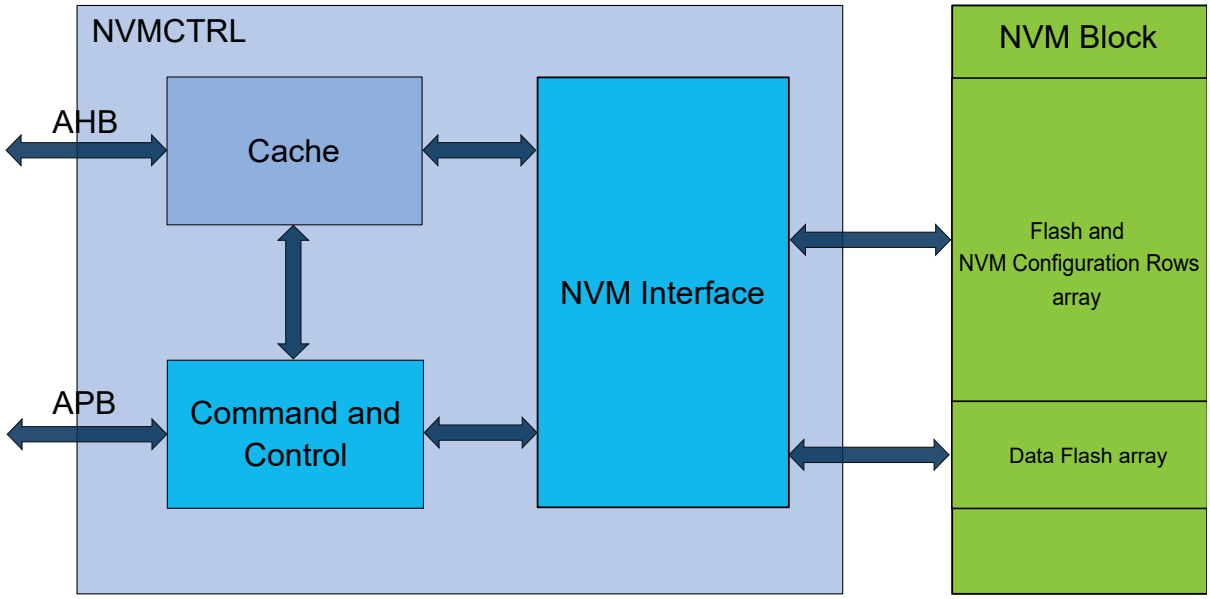
Non-Volatile Memory (NVM) is a reprogrammable Flash memory that retains program and data storage even with power off. It embeds three separate arrays, namely Flash, Data Flash and NVM Configuration Rows. The Data Flash array can be programmed while reading the Flash array. It is intended to store data while executing from the Flash without stalling. NVM Configuration Rows store the data needed during the device start-up, such as calibration and system configuration. The NVM Controller (NVMCTRL) connects to the AHB and APB bus interfaces for system access to the NVM block. The AHB interface is used for reads and writes to the NVM block, while the APB interface is used for commands and configuration.

29.2 Features

- 32-bit AHB interface for reads and writes
- Write-While-Read (WWR) Data Flash
- All NVM sections are memory mapped to the AHB, including calibration and system configuration
- 32-bit APB interface for commands and control
- Programmable wait states for read optimization
- Up to 5 regions can be individually protected or unprotected
- Additional protection for bootloader
- Interface to power manager for power-down of Flash blocks in sleep modes
- Can optionally wake-up on exit from sleep or on first access
- Direct-mapped cache
- TrustZone support (PIC32CM LS00/LS60)

29.3 Block Diagram

Figure 29-1. Block Diagram



29.4 Peripheral Dependencies

Table 29-1. NVMCTRL Configuration Summary

Peripheral name	Base address	NVIC IRQ Index	MCLK AHB/APB Clocks	PAC Peripheral Identifier Index (PAC.WRCTRL)	Events (EVSYS)		Power Domain (PM.STDBYCFG)
					Users (EVSYS.USER)	Generators (EVSYS.CHANNEL)	
NVMCTRL	0x41004000	13: DONE, PROGE, LOCKE, NVME, KEYE, NSCHK	CLK_NVMCTRL_AHB CLK_NVMCTRL_APB	34	2: AUTOW	—	PDSW

29.5 Functional Description

29.5.1 Principle of Operation

The NVM Controller is a client on the AHB and APB buses. It responds to commands, read requests and write requests, based on user configuration.

29.5.1.1 Initialization

After power up, the NVM Controller goes through a power-up sequence. During this time, access to the NVM Controller from the AHB bus is halted. Upon power-up completion, the NVM Controller is operational without any need for user configuration.

29.5.1.2 Power Management

The NVMCTRL will continue to operate in any sleep mode where the selected source clock is running. The NVMCTRL interrupts can be used to wake up the device from sleep modes.

The Power Manager will automatically put the NVM block into a low-power state when entering sleep mode. This is based on the Control B register (CTRLB) SLEEPPRM bit setting. Refer to the 29.6.2. CTRLB.SLEEPPRM register description for more details. The NVM block goes into low-power mode automatically when the device enters STANDBY mode regardless of SLEEPPRM. The NVM Page Buffer is lost when the NVM goes into low power mode therefore a write command must be issued prior entering the NVM low power mode. NVMCTRL SLEEPPRM can be disabled to avoid such loss when the CPU goes into sleep except if the device goes into STANDBY mode for which there is no way to retain the Page Buffer.

29.5.1.3 Clocks

Two synchronous clocks are used by the NVMCTRL. One is provided by the AHB bus (CLK_NVMCTRL_AHB) and the other is provided by the APB bus (CLK_NVMCTRL_APB). For higher system frequencies, a programmable number of wait states can be used to optimize performance. When changing the AHB bus frequency, the user must ensure that the NVM Controller is configured with the proper number of wait states.

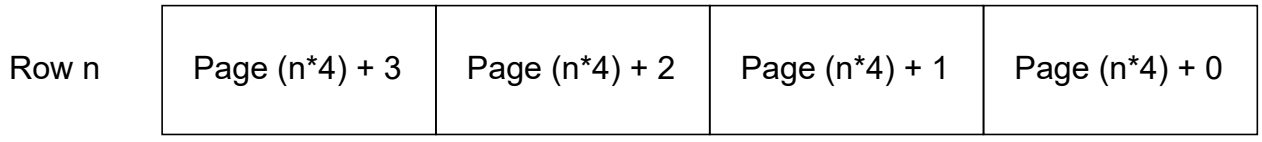
Number of wait states to be used for a particular frequency range can be found on the NVM Block Electrical Specifications from Electrical Characteristics chapter.

29.5.2 Memory Organization

Refer to the Physical Memory Map for memory sizes and addresses for each device.

The NVM is organized into rows, where each row contains four pages, as shown in the NVM Configuration Row Organization figure. The NVM has a row-erase granularity, while the write granularity is by page. In other words, a single row erase will erase all four pages in the row, while four write operations are used to write the complete row.

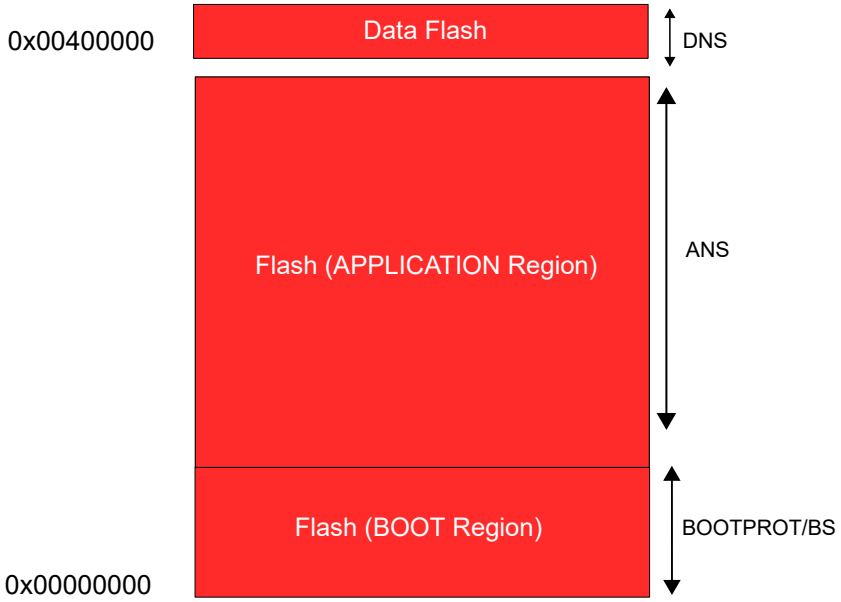
Figure 29-2. NVM Configuration Row Organization



The NVM block contains the NVM Configuration Rows which contain calibration and system configuration, the FLASH area intended to store code and a separate array dedicated to data storage called Data FLASH that can be modified while the FLASH is read (no bus stall). All these areas are memory mapped. Refer to the NVM Organization figure below for details.

The NVM Configuration Rows contain factory calibration and system configuration information. These spaces can be read from the AHB bus in the same way as the FLASH. Note that Data FLASH requires more cycles to be read. The Data FLASH can be executable, however this is not recommended as it can weaken an application security and also affect performances.

Figure 29-3. NVM Memory Organization



The lower rows in the FLASH can be allocated as a boot loader section by using the BOOTPROT fuses.

The boot loader section size is defined by the BOOTPROT fuses expressed in number of rows.

Important: Refer to the Boot ROM section to get Chip Erase commands effects for this specific BOOT area.

29.5.3 Region Unlock Bits

The NVMCTRL can lock regions defined in the NVM Memory Organization figures.

When a region is locked all modify (that is, write or erase) commands directed to these regions are discarded. When such an operation occurs a LOCKE error is reported in the INTFLAG register and can generate an interruption.

Important: The Chip Erase commands, executed by specific Boot ROM commands, are not discarded and will be executed whether the region is locked or not. Refer to Chip Erase in the Boot ROM chapter.

To lock or unlock a region, write a one to the bitfield corresponding to the selected regions in the SULCK and NSULCK registers with the correct key. Writes to these registers are silently discarded when the key is not correct. Writing these registers with the correct key will temporarily lock or unlock the corresponding regions. The new setting will stay in effect until the next Reset, or until the setting is changed again while writing SULCK and NSULCK. The current status of the lock can be determined by reading the SULCK and NSULCK registers. To change the default lock/unlock setting for a region, the NVM User Row (UROW) must be written using the Write Page command. Writing to the NVM User Row (UROW) will take effect after the next Reset. Therefore, a boot of the device is needed for

changes in the lock or unlock setting to take effect. Refer to the ‘Physical Memory Map’ for NVM User Row (UROW) space address mapping.

29.5.4 Command and Data Interface

The NVM Controller is addressable from the APB bus, while the NVM main address space is addressable from the AHB bus. Read and automatic page write operations are performed by addressing the FLASH, Data FLASH and NVM Configuration Rows arrays directly, while other operations such as manual page writes and row erases must be performed by issuing commands through the NVM Controller.

To issue a command, the CTRLA.CMD bits must be written along with the CTRLA.CMDEX value. When a command is issued, STATUS.READY is cleared and rises again when the command has completed. INTFLAG.DONE is also set when a command completes. Any commands written while INTFLAG.READY is low will be ignored.

The CTRLB and CTRLC registers must be used to control the power reduction mode, read wait states, and the write mode.

29.5.4.1 FLASH Read

Reading from the FLASH is performed via the AHB bus. Read data is available after the configured number of read wait states (CTRLB.RWS) set in the NVM Controller.

Reading the Flash while a programming or erase operation is ongoing on the Flash results in an AHB bus stall until the end of the operation. Reading the Flash does not stall the bus when the Data FLASH is being programmed or erased.

29.5.4.2 DATA FLASH Read

Reading from the Data FLASH is performed via the AHB bus by addressing the Data FLASH address space directly.

Read timings are increased by one cycle compared to regular FLASH read timings when access size is Byte or half-Word. The AHB data phase is twice as long in case of full-Word-size access.

It is not possible to read the Data FLASH while the Flash is being written or erased (the read is stalled), whereas the Data FLASH can be written or erased while the Flash is being read.

The Data FLASH address space is not cached, therefore it is recommended to limit access to this area for performance and power consumption considerations.

29.5.4.3 FLASH, DATA FLASH Write

Data to be written to the NVM block are first written to and stored in an internal buffer called the page buffer. The page buffer contains the same number of bytes as an NVM page. Writes to the page buffer must be 16 or 32 bits. 8-bit writes to the page buffer are not allowed and will cause a bus error.

Both FLASH and Data FLASH share the same page buffer. Writing to the NVM block via the AHB bus is performed by a load operation to the page buffer. For each AHB bus write, the address is stored in the ADDR register. After the page buffer has been loaded with the required number of bytes, the page can be written to the array pointed by ADDR by setting CTRLA.CMD to 'Write Page' and setting the key value to CMDEX. The LOAD bit in the STATUS register indicates whether the page buffer has been loaded or not.

If the NVMCTRL is busy processing a write command (STATUS.READY=0), then the AHB bus is stalled upon AHB write until the ongoing command completes.

The NVM Controller requires that an erase must be done before programming. Rows can be individually erased by the Erase Row command to erase a row.

Automatic page writes are enabled by writing the manual write bit to zero (CTRLC.MANW=0). This will trigger a write operation to the page addressed by ADDR when the last location of the page is written.

Because the address is automatically stored in ADDR during the APB bus write operation, the last given address will be present in the ADDR register. There is no need to load the ADDR register manually, unless a different page in memory is to be written. The page buffer is automatically cleared upon a 'Write Page' command completion.

29.5.4.3.1 Procedure for Manual Page Writes (CTRLC.MANW=1)

The row to be written to must be erased before the write command is given.

- Write to the page buffer by addressing the NVM main address space directly
- Write the page buffer to memory: CTRL.CMD='Write Page' and CMDEX

- The READY bit in the INTFLAG register will be low while programming is in progress, and access through the AHB will be stalled

29.5.4.3.2 Procedure for Automatic Page Writes (CTRLC.MANW=0)

The row to be written to must be erased before the last write to the page buffer is performed.

Note that partially written pages must be written with a manual write.

- Write to the page buffer by addressing the NVM main address space directly.
  When the last location in the page buffer is written, the page is automatically written to NVM main address space.
- INTFLAG.READY will be zero while programming is in progress and access through the AHB will be stalled.

29.5.4.4 Page Buffer Clear

The page buffer is automatically set to all '1' after a page write is performed. If a partial page has been written and it is desired to clear the contents of the page buffer, the Page Buffer Clear command can be used.

The status of the Page Buffer is reflected by the STATUS.LOAD bitfield, when the PBC command is issued successfuly, STATUS.LOAD reads 0.

29.5.4.5 Erase Row

Before a page can be written, the row containing that page must be erased. The Erase Row command can be used to erase the desired row in the NVM (same command for FLASH, Data FLASH and NVM Configuration Rows). Erasing the row sets all bits to '1'. If the row resides in a region that is locked, the erase will not be performed and the Lock Error bit in the INTFLAG register (INTFLAG.LOCKE) will be set.

29.5.4.5.1 Procedure for Erase Row

- Write the address of the row to erase to ADDR. Any address within the row can be used.
- Issue an Erase Row command.

Note: The NVM Address bit field in the Address register (ADDR.ADDR) uses 16-bit addressing.

29.5.4.6 Set and Clear Power Reduction Mode

The NVM Controller and block can be taken in and out of power reduction mode through the Set and Clear Power Reduction Mode commands. When the NVM Controller and block are in power reduction mode, the Power Reduction Mode bit in the Status register (STATUS.PRM) is set.

29.5.5 NVM Configuration Rows Operations

Reading from or writing/erasing to the NVM Configuration rows is performed in the same manner as Flash memory, the only difference being the addressed space.

29.5.6 Events

The NVMCTRL can take the following actions on an input event:

- Write zeroes in one Data FLASH row: Refer to 29.5.8. Tamper Erase for details.
- Write a page in the FLASH or in the Data FLASH: Refer to 29.5.7. Event Automatic Write for details.

The NVMCTRL uses only asynchronous events, so the asynchronous Event System channel path must be configured. By default, the NVMCTRL will detect a rising edge on the incoming event. If the NVMCTRL action must be performed on the falling edge of the incoming event, the event line must be inverted first. This is done by setting the corresponding Event Invert Enable bit in Event Control register (NVMCTRL.AUTOWINV=1).

29.5.7 Event Automatic Write

The Event Automatic Write feature is enabled by setting EVCTRL.AUTOWEI = 1. When enabled, an event input from EVSYS will trigger a page programming command. The polarity of the input event can be inverted by setting EVCTRL.AUTOWINV. The page written is addressed by the address register (ADDR) and can reside in program or data memory. To use this feature, the row must be previously erased and the page buffer must contain the desired data to be written.

As the Page Buffer is lost when the NVM enters low power mode (refer to 29.5.1.2. Power Management) cannot be used if the device enters STANDBY mode or if the NVM uses power reduction modes.

The cache coherency is not ensured after an Event Automatic Write in a FLASH page. The FLASH region is cacheable, it is the user responsibility to clear the cache after such an action. Note that the Data FLASH is not subject to cache coherency issues since it is not cacheable.

29.5.8 Tamper Erase

Tamper Erase ensures rapid overwrite on tamper of a Data FLASH row selected by SECCTRL.TEROW.

When a RTC tamper event occur while tamper erase is enabled (SECCTRL.TAMPEEN=1):

- the Tamper Erase row in data space addressed by SECCTRL.TEROW is written to zero.

This is performed using a special overwrite mechanism in the NVM block that overwrites the complete row with zero. The RTC must be configured to generate the tamper erase event.

Note: Data Flash endurance is affected by the tamper erase feature. Refer to the 49.35. NVM Block (Flash, Data Flash, NVM Configuration Rows) Electrical Specifications from Electrical Characteristics chapter.

29.5.9 Silent Access

When enabled (SECCTRL.SILACC = 1), the silent access feature allows to store data and their 1's complement in one Data Flash row (selected by SECCTRL.TEROW), thus reducing the overall data reading noise, as always the same number of '0' and '1' will be read for each read access.

When Silent Access is enabled, the logical size of the TEROW Flash row is divided by two to store each byte of Data and its 1's complement (CompData) in the whole physical TEROW size.

The data stored in the selected TEROW must be accessed using the logical mapping shown in the TEROW logical mapping table.

Table 29-2. TEROW logical mapping (SECCTRL.SILACC=1)

Byte 63	Byte 62	Byte 61	…	Byte 3	Byte 2	Byte 1	Byte 0	Page
Data	Data	Data	Data	Data	Data	Data	Data	0
Data	Data	Data	Data	Data	Data	Data	Data	1
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	2
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	3

Note: All accesses to the reserved area of the TEROW are discarded and generate a bus error.

The physical mapping of the TEROW, when silent access feature is enabled, is represented in the TEROW physical mapping table:

Table 29-3. TEROW physical mapping (SECCTRL.SILACC=1)

Byte 63	Byte 62	Byte 61	…	Byte 3	Byte 2	Byte 1	Byte 0	Page
CompData	Data	CompData	Data	CompData	Data	CompData	Data	0
CompData	Data	CompData	Data	CompData	Data	CompData	Data	1
CompData	Data	CompData	Data	CompData	Data	CompData	Data	2
CompData	Data	CompData	Data	CompData	Data	CompData	Data	3

The NVMCTRL automatically manages both scrambling and differential data storage if the tamper row resides in the secure Data Flash area and both are enabled (SECCTRL.DSCEN = 1 and SECCTRL.SILACC = 1).

29.5.10 Chip Erase

The various chip erase operations are managed by the boot ROM code. For more details, refer to the Boot ROM section.

29.5.11 Cache

The NVM Controller cache reduces the device power consumption and improves system performance when wait states are required. Only the Flash area is cached (Data Flash is not). It is a direct-mapped cache that implements 64 lines of 64 bits (that is, 512 Bytes). NVM Controller cache can be enabled by writing a '0' to the Cache Disable bit in the Control B register (CTRLB.CACHEDIS).

The cache can be configured to three different modes using the Read Mode bit group in the Control B register (CTRLB.READMODE).

The INVALL command can be issued using the Command bits in the Control A register to invalidate all cache lines (CTRLA.CMD=INVALL). Commands affecting NVM content automatically invalidate cache lines.

29.5.12 Debug Operation

When an external debugger forces the CPU into debug mode, the peripheral continues normal operation except that FLASH reads are not cached so that the cache state is not altered by debug tools.

29.5.13 Debugger Access Level

The Debugger Access Level (DSU STATUSB.DAL) defines the access rights of a debugger connected to the device.

- 0x0 = Access to very limited features (basically only the DSU external address space)
- 0x1 = Access to all non-secure memory; can debug non-secure CPU code (**PIC32CM LS00/LS60 only**)
- 0x2 = Access to all memory; can debug secure and non-secure CPU code

DAL can be set to a lower setting using a SDAL command (CTRLA register).

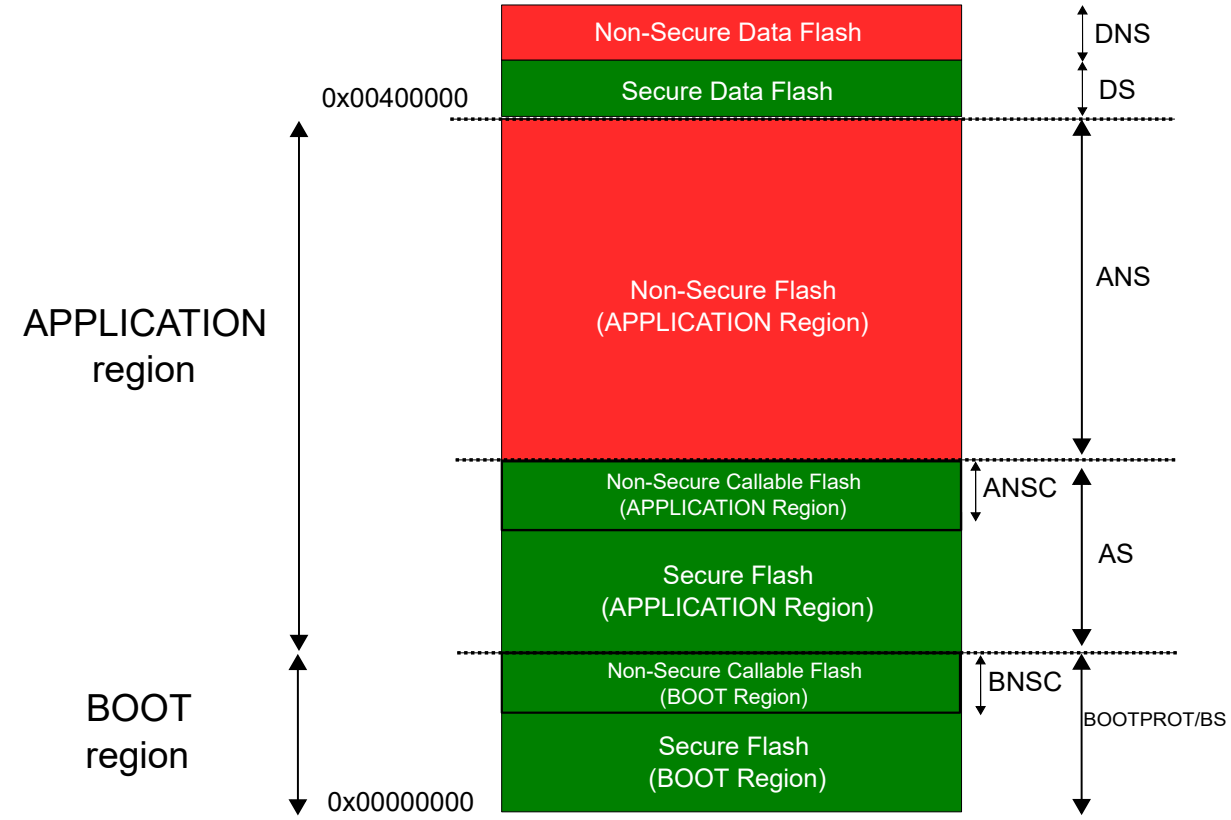
Important:

- Issuing a SDAL command to set a higher setting for DAL is only possible if NVMCTRL.SECCTRL.DALUN == 1
- If NVMCTRL.SECCTRL.DALUN == 0 and a SDAL command to set a higher value is performed, an INTFLAG.PROGE error is issued. In this case, only a Chip Erase can change DAL to a higher setting

29.5.14 PIC32CM LS00/LS60 TrustZone Protection Considerations

On TrustZone protected devices, the Flash and Data Flash areas are partitioned into secure, non-secure, and non-secure callable sections to accommodate with TrustZone core capability.

Figure 29-4. NVM Memory Organization



The various memory regions and attributes are provides in the table below.

Table 29-4. Memory Regions and Attributes

Memory Region	Base Address	Size	Attribute
Secure Flash (BOOT region)	0x00000000	BOOTPROT*ROWSIZE - BNSC*0x20	Secure
Non-Secure Callable Flash (BOOT region)	BOOTPROT*ROWSIZE- BNSC*0x20	BNSC*0x20	Secure
Secure Flash (APPLICATION region)	BOOTPROT*ROWSIZE	AS*ROWSIZE - ANSC*0x20	Secure
Non-Secure Callable Flash (APPLICATION region)	BOOTPROT*ROWSIZE- ANSC*0x20	ANSC*0x20	Secure
Non-Secure Flash (APPLICATION region)	(BOOTPROT+AS)*ROWSIZE	(remaining APPLICATION region)	Non-secure
Secure Data Flash	0x00400000	DS*ROWSIZE	Secure
Non-Secure Data Flash	0x00400000 + DS*ROWSIZE	(remaining Data NVM area)	Non-secure

Access to various sections is restricted as shown in the following table. All sections can be read and write without restriction when the access is secure. When the access is non-secure the secure sections are not accessible. When defined non-secure callable sections have the same attributes as the secure sections, therefore the NVMCTRL considers them as secure regions. The system may also have a secure callable boot and application regions. These regions have the same attributes as the secure sections, so there is no special treatment needed in NVMCTRL.

Any illegal access will result in a bus error. The boot and application non-secure callable regions are shown for reference but have no effect on the NVMCTRL. These regions are included in secure regions therefore the NVMCTRL considers them as secure regions.

Table 29-5. Memory Regions AHB Access Limitations

Memory Region	Secure Access	Non-Secure Access	Limitations
Secure Flash (BOOT region)	R+W	-	-
Non-Secure Callable Flash (BOOT region)	R+W	-	-
Secure Flash (APPLICATION region)	R+W	-	
Non-Secure Callable Flash (APPLICATION region)	R+W	-	-
Non-Secure Flash (APPLICATION region)	R+W	R+W	
Secure Data Flash	R+W	-	
Non-Secure Data Flash	R+W	R+W	
NVM Software Calibration Row	R+W	R	
NVM User Row (UROW)	R+W	R	
NVM Boot Configuration Row(BOCOR)	R+W	-	No read if BCREN is cleared. No write if BCWEN is cleared.

The Boot Configuration row (BOCOR) contains information that is read by the boot ROM and written to the IDAU and NVMCTRL registers. The BOCOR is read/writable if SCFGB.BCREN/BCWEN are set, respectively.

Important: SCFGB.BCREN/BCWEN are copied from BOCOR during the Boot ROM execution.

Table 29-6. Memory Regions Modify operations Limitations (WP, EP commands)

Memory Region	Secure Access	Non-Secure Access	Limitations
Secure Flash (BOOT region)	Y	N	No if SULCK.BS=0
Non-Secure Callable Flash (BOOT region)	Y	N	No if SULCK.BS=0
Secure Flash (APPLICATION region)	Y	N	No if SULCK.AS=0
Non-Secure Callable Flash (APPLICATION region)	Y	N	No if SULCK.AS=0

.....continued

Memory Region	Secure Access	Non-Secure Access	Limitations
Non-Secure Flash (APPLICATION region)	Y	Y	No if NSULCK.ANS=0
Secure Data Flash	Y	N	No if SULCK.DS=0
Non-Secure Data Flash	Y	Y	No if NSULCK.DNS=0
NVM User Row (UROW)	Y	N	No if BOCOR.URWEN=0
NVM Boot Configuration Row (BOCOR)	Y	N	No if BOCOR.BCWEN=0

The NSULCK SULCK bitfields in the user row define the NSULCK and SULCK register default value after a reset.

Special care must be taken when sharing the NVMCTRL between the secure and non-secure domains. When the secure code modifies the NVM, it is recommended that it disables all write accesses to the APB non-secure alias and writes to AHB non-secure regions by writing a ‘0’ to NONSEC.WRITE. This avoids any interference with non-secure modify operations. In this case, even a secure application cannot write the page buffer at a non-secure location because the IDAU changes security attributions of Non-Secure transactions to Non-Secure regions to Non-Secure.

The NONSEC.WRITE reset value is '1', meaning that it is always possible to program a Non-Secure Flash or Data Flash region after a debugger probe cold-plugging. But if the debugger connects with the hot-plugging procedure then NONSEC.WRITE must be '1' to let the debugger program Non-Secure regions otherwise the transaction will cause a hardfault (seen as a DAP fault at DAP level).

For applications that do not require Non-Secure regions programming other than from a secure code, it is recommended to always disable Non-Secure writes by disabling NONSEC.WRITE. When disabled secure code needs to enable it to be able to modify Non-Secure regions following this procedure:

1. Disable the interrupt.
2. Write a ‘1” to NONSEC.WRITE to allow writes to the non-secure region.
3. Write the page buffer.
4. Write a ‘0’ to NONSEC.WRITE.
5. Enable the interrupt.

If the NSCHK interrupt is enabled, a NONSEC.WRITE modification will generate an interrupt so that the non-secure world is aware of this change. Depending on NSCHK.WRITE and INTFLAG.NSCHK will rise upon a rising or falling NONSEC.WRITE transition. The interrupt can be configured as secure or non-secure in the NVIC. If secure then a software mechanism can be implemented to call a non-secure NVMCTRL IRQ handler from the secure world.

The NVMCTRL monitors the Page Buffer write accesses and accepts only writes to non-secure regions when the transaction is non-secure. Moreover it checks that any write to the page buffer is in the same page as the previous write when the Page Buffer is not empty. When this check fails, an error is returned to the bus host that initiated the transaction. This ensures that it is not possible to mix different page writes into the Page Buffer. Therefore, any Page Buffer write access must at some point be followed by a manual or automatic Write Page (WP) that automatically clears the page buffer or a Clear Page Buffer (PBC) command.

For security reasons, the ADDR register is not accessible from the non-secure alias. The only way to change it is to write a data to the Page Buffer. If the intention is to issue a command that doesn't write the NVM (for instance an Erase Row command (ER)) then the PBC command must be issued to avoid locking further write accesses (even secure writes). The status of the Page Buffer is reflected by the STATUS.LOAD bitfield.

29.5.14.1 Page Buffer Clear

When Page Buffer Clear command is issued from the non-secure APB alias, ADDR must point on a non-secure region otherwise the command is silently discarded. For security reasons, the ADDR register is not accessible from the non-secure alias. The only way to change it is to write a data to the Page Buffer. If the intention is to issue a command that doesn't write the NVM (for instance an Erase Row command (ER)) then the PBC command must be

issued to avoid locking further write accesses (even secure writes). ADDR must point to a non-secure NVM region when PBC is issued from the non-secure alias.

29.5.14.2 Page Write

The NVMCTRL monitors the Page Buffer write accesses and accepts only writes to non-secure regions when the transaction is non-secure. Moreover it checks that any write to the page buffer is in the same page as the previous write when the Page Buffer is not empty. When this check fails, an error is returned to the bus host that initiated the transaction. This ensures that it is not possible to mix different page writes into the Page Buffer. Therefore, any Page Buffer write access must at some point be followed by a manual or automatic Write Page (WP) that automatically clears the page buffer or a Clear Page Buffer (PBC) command.

For security reasons, the ADDR register is not accessible from the non-secure alias. The only way to change it is to write a data to the Page Buffer. If the intention is to issue a command that doesn't write the NVM (for instance an Erase Row command (ER)) then the PBC command must be issued to avoid locking further write accesses (even secure writes). The status of the Page Buffer is reflected by the STATUS.LOAD bitfield.

29.5.14.3 Erase Row

ADDR must point to a non-secure region when an ER command is issued from the non-secure APB alias.

29.5.14.4 Lock Regions

The NVMCTRL has the ability to lock regions with respect to the IDAU memory mapping:

- FLASH Boot Secure and Non-Secure Callable regions
- FLASH Application secure region
- FLASH Application non-Secure and Non-Secure Callable regions
- Data FLASH Secure region
- Data FLASH Non-Secure region

When a region is locked, all modify commands (i.e. write or erase) directed to this region are discarded. A LOCKE error is reported in the INTFLAG register and can generate an interrupt.

To lock or unlock a region, write a one to the corresponding bitfield in SULCK and NSULCK registers Writes to these registers are silently discarded if the key is not correct. Writing these registers with the correct key will temporarily lock or unlock the corresponding regions. The new lock setting will stay in effect until the next reset, or until the setting is changed again while writing SULCK and NSULCK.

Note: Writes to these registers are silently discarded if the key is not correct.

The current status of the lock can be determined by reading the SULCK and NSULCK registers. To change the default lock/unlock setting for a region, the NVM User Row (UROW) must be written using the Write Page command. Writing to the NVM User Row (UROW) will take effect after the next Reset. Therefore, a boot of the device is needed for changes in the lock/unlock setting to take effect. Refer to the Physical Memory Map for NVM User Row (UROW) space address mapping.

SULCK is a Write-Secure register:

- This register can only be written by secure hosts from the secure alias
- This register is always readable by secure or non-secure hosts from their respective alias

NSULCK is a Write-Mix-Secure register:

- This register can always be written by a secure host from the secure alias
- Or, by a non-secure host from the non-secure alias only if NONSEC.WRITE is set
- This register is always readable by secure or non-secure hosts in their respective alias

29.5.14.5 Cache

When a line is cached, the type of transaction is stored in the cache. If the line has been updated upon a secure transaction, only secure transaction can hit, otherwise there is a cache miss and the transaction propagates to the NVMCTRL which enforces the security. If the line has been updated upon a non-secure transaction, it can be hit by both the secure or non-secure transactions. In case of a non-secure transaction cache miss, a line is replaced even if it contained a secure data.

29.5.14.6 Data Flash Scrambling

When Data Flash scrambling is enabled (SECCTRL.DSCEN = 1), address and data in the secure portion of the Data Flash are scrambled when written, and de-scrambled when read.

The NVMCTRL automatically manage both scrambling and differential data storage if the tamper row resides in the secure Data Flash area and both are enabled (SECCTRL.DSCEN = 1 and SECCTRL.SILACC = 1).
Note: In this case, scrambling is started followed by silent access on writes and the reverse on reads.

29.5.14.7 Tamper Erase

The scrambling key stored in DSCC is written to zero when a RTC tamper event occurs in addition to the erase of the row.

29.6 Register Summary

Refer to the Registers Description section for more details on register properties and access permissions.

Important: (PIC32CM LS00 only)
The peripheral register map is automatically duplicated in a Secure and Non-Secure alias:

- The Non-Secure alias is at the peripheral base address
- The Secure alias is located at the peripheral base address + 0x1000

Refer to Mix-Secure Peripherals for more information on register access rights.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRLA	15:8	CMDEX[7:0]							
		7:0		CMD[6:0]						
0x02 ... 0x03	Reserved									
0x04	CTRLB	31:24								
		23:16					Reserved	CACHEDIS	READMODE[1:0]	
		15:8					FWUP		SLEEPPRM[1:0]	
		7:0		Reserved		RWS[3:0]				Reserved
0x08	CTRLC	7:0								MANW
0x09	Reserved									
0x0A	EVCTRL	7:0							AUTOWINV	AUTOWEI
0x0B	Reserved									
0x0C	INTENCLR	7:0			NSCHK	KEYE	NVME	LOCKE	PROGE	DONE
0x0D ... 0x0F	Reserved									
0x10	INTENSET	7:0			NSCHK	KEYE	NVME	LOCKE	PROGE	DONE
0x11 ... 0x13	Reserved									
0x14	INTFLAG	7:0			NSCHK	KEYE	NVME	LOCKE	PROGE	DONE
0x15 ... 0x17	Reserved									
0x18	STATUS	15:8	Reserved							
		7:0				DALFUSE[1:0]		READY	LOAD	PRM
0x1A ... 0x1B	Reserved									
0x1C	ADDR	31:24								
		23:16	ARRAY[1:0]							
		15:8	AOFFSET[15:8]							
		7:0	AOFFSET[7:0]							
0x20	SULCK	15:8	SLKEY[7:0]							
		7:0						DS	AS	BS
0x22	NSULCK	15:8	NSLKEY[7:0]							
		7:0						DNS	ANS	
0x24	PARAM	31:24	DFLASHP[11:4]							
		23:16	DFLASHP[3:0]					PSZ[2:0]		
		15:8	FLASHP[15:8]							
		7:0	FLASHP[7:0]							
0x28 ... 0x2F	Reserved									

.....continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x30	DSCC	31:24			DSCKEY[29:24]					
		23:16	DSCKEY[23:16]							
		15:8	DSCKEY[15:8]							
		7:0	DSCKEY[7:0]							
0x34	SECCTRL	31:24	KEY[7:0]							
		23:16								
		15:8						TEROW[2:0]		
		7:0		DXN	DALUN	SCFGWEN	DSCEN	SILACC		TAMPEEN
0x38	SCFGB	31:24								
		23:16								
		15:8								
		7:0							BCWEN	BCREN
0x3C	SCFGAD	31:24								
		23:16								
		15:8								
		7:0								URWEN
0x40	NONSEC	31:24								
		23:16								
		15:8								
		7:0								WRITE
0x44	NSCHK	31:24								
		23:16								
		15:8								
		7:0								WRITE

29.6.1 Control A

Name: CTRLA
Offset: 0x00
Reset: 0x0000
Property: PAC Write-Protection, Write-Mix-Secure

Important: For **PIC32CM LS00/LS60 Non-Secure** accesses, write accesses (W*) are allowed only if Non-Secure Write is set in the NONSEC register.

Bit	15	14	13	12	11	10	9	8
	CMDEX[7:0]							
Access	W/W*/W	W/W*/W	W/W*/W	W/W*/W	W/W*/W	W/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
		CMD[6:0]						
Access		W/W/W	W/W/W	W/W/W	W/W/W	W/W/W	W/W/W	W/W/W
Reset		0	0	0	0	0	0	0

Bits 15:8 – CMDEX[7:0] Command Execution
When this bit group is written to the key value 0xA5, the command written to CMD will be executed. If a value different from the key value is tried, the write will not be performed and the key error interrupt (INTFLAG.KEYE) will be set. PROGE is set if a previously written command is not completed yet or in case of bad conditions.
The key value must be written at the same time as CMD. If a command is issued through the APB bus on the same cycle as an AHB bus access, the AHB bus access will be given priority. The command will then be executed when the NVM block and the AHB bus are idle.
STATUS.READY must be '1' when the command has issued.
Note: The NVM Address bit field in the Address register (ADDR.ADDR) is driving the hardware (8-bit) address to the NVM when a command is executed using CMDEX.

Bits 6:0 – CMD[6:0] Command
These bits define the command to be executed when the CMDEX key is written.

Important: For **PIC32CM LS00/LS60**, only ER, WP, PBC, SDAL0 commands are available from the non-secure alias. Non-secure ER, WP, PBC are processed only if ADDR points to a non secure address, otherwise a PROGE error is issued.

Value	Group Configuration	Description
0x00-0x01	-	Reserved
0x02	ER	Erase Row - Erases the row addressed by the ADDR register in the Flash, Data Flash or NVM Configuration Rows.
0x03	-	Reserved
0x04	WP	Write Page - Writes the contents of the page buffer to the page addressed by the ADDR register.
0x05-0x41	-	Reserved
0x42	SPRM	Sets the Power Reduction Mode.
0x43	CPRM	Clears the Power Reduction Mode.
0x44	PBC	Page Buffer Clear - Clears the page buffer.
0x45	-	Reserved
0x46	INVALL	Invalidates all cache lines.

.....continued		
Value	**Group Configuration**	**Description**
0x47-0x4A	-	Reserved
0x4B	SDAL0	Set DAL=0
0x4C (**PIC32CM LE00**)	-	Reserved
0x4C (**PIC32CM LS00/ LS60**)	SDAL1	Set DAL=1
0x4D (**PIC32CM LE00**)	-	Reserved
0x4D (**PIC32CM LS00/ LS60**)	SDAL2	Set DAL=2 (if BOCOR.SECCFGLOCK == 1) Reserved (if BOCOR.SECCFGLOCK == 0)
0x4E-0x7F	-	Reserved

29.6.2 Control B

Name: CTRLB
Offset: 0x04
Reset: 0x00000000
Property: PAC Write-Protection, Write-Secure

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					Reserved	CACHEDIS	READMODE[1:0]	
Access					RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
					FWUP		SLEEPPRM[1:0]	
Access					RW/R/RW		RW/R/RW	RW/R/RW
Reset					0		0	0

Bit	7	6	5	4	3	2	1	0
		Reserved		RWS[3:0]				Reserved
Access		RW/R/RW		RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW
Reset		0		0	0	0	0	0

Bit 19 – Reserved Must Be Set to 0
This bit must always be set to '0' when programming the register.

Bit 18 – CACHEDIS Cache Disable
This bit is used to disable the cache.

Value	Description
0	The cache is enabled
1	The cache is disabled

Bits 17:16 – READMODE[1:0] NVMCTRL Read Mode

Value	Name	Description
0x0	NO_MISS_PENALTY	The NVM Controller (cache system) does not insert wait states on a cache miss. Gives the best system performance.
0x1	LOW_POWER	Reduces power consumption of the cache system, but inserts a wait state each time there is a cache miss. This mode may not be relevant if CPU performance is required, as the application will be stalled and may lead to increased run time.
0x2	DETERMINISTIC	The cache system ensures that a cache hit or miss takes the same amount of time, determined by the number of programmed Flash wait states. This mode can be used for real-time applications that require deterministic execution timings.
0x3	Reserved	-

Bit 11 – FWUP Fast Wake-Up Enable
This bit is used to enable the fast wake-up mode.

Value	Description
0	Fast wake-up is turned off
1	Fast wake-up is turned on

Bits 9:8 – SLEEPPRM[1:0] Power Reduction Mode during Sleep

Indicates the Power Reduction Mode during sleep.

Value	Name	Description
0x0	WAKEONACCESS	NVM block enters low-power mode when entering sleep. NVM block exits low-power mode upon first access.
0x1	WAKEUPINSTANT	NVM block enters low-power mode when entering sleep. NVM block exits low-power mode when exiting sleep.
0x2	Reserved	-
0x3	DISABLED	Auto power reduction disabled.

Bit 6 – Reserved Must Be Set to 0

This bit must always be set to ‘0’ when programming the register.

Bits 4:1 – RWS[3:0] NVM Read Wait States

These bits control the number of wait states for a read operation. '0' indicates zero wait states, '1' indicates one wait state, etc., up to 15 wait states.

This register is initialized to 0 wait states. Software can change this value based on the NVM access time and system frequency.

Bit 0 – Reserved Must Be Set to 0

This bit must always be set to ‘0’ when programming the register.

29.6.3 Control C

Name: CTRLC
Offset: 0x08
Reset: 0x01
Property: PAC Write-Protection, Write-Mix-Secure

Important: For **PIC32CM LS00/LS60 Non-Secure** accesses, write accesses (W*) are allowed only if Non-Secure Write is set in the NONSEC register.

Bit	7	6	5	4	3	2	1	0
								MANW
Access								RW/RW*/RW
Reset								1

Bit 0 – MANW Manual Write

Value	Description
0	Writing to the last word in the page buffer will initiate a write operation to the page addressed by the last write operation. This includes writes to FLASH, Data FLASH and NVM Configuration rows.
1	Write commands must be issued through the CTRLA.CMD register.

29.6.4 Event Control

Name: EVCTRL
Offset: 0x0A
Reset: 0x00
Property: PAC Write-Protection, Secure

Bit	7	6	5	4	3	2	1	0
							AUTOWINV	AUTOWEI
Access							RW/-/RW	RW/-/RW
Reset							0	0

Bit 1 – AUTOWINV Event Action

Value	Description
0	Input event polarity is not inverted.
1	Input event polarity is inverted.

Bit 0 – AUTOWEI Event Action

Value	Description
0	Input event has no effect.
1	Input event triggers an Automatic Page Write

29.6.5 Interrupt Enable Clear

Name: INTENCLR
Offset: 0x0C
Reset: 0x00
Property: PAC Write-Protection, Write-Mix-Secure

Important: For **PIC32CM LS00/LS60 Non-Secure** accesses, write accesses (W*) are allowed only if Non-Secure Write is set in the NONSEC register.

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
			NSCHK	KEYE	NVME	LOCKE	PROGE	DONE
Access			RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW
Reset			0	0	0	0	0	0

Bit 5 – NSCHK Non-secure Check Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the NSCHK interrupt enable.

Value	Description
0	The NSCHK interrupt is disabled
1	The NSCHK interrupt is enabled

Bit 4 – KEYE Key Error Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the KEYE interrupt enable.

Value	Description
0	The KEYE interrupt is disabled
1	The KEYE interrupt is enabled

Bit 3 – NVME NVM internal Error Interrupt Clear
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the NVME interrupt enable.

Value	Description
0	The NVME interrupt is disabled
1	The NVME interrupt is enabled

Bit 2 – LOCKE Lock Error Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit sets the LOCKE interrupt enable.

Value	Description
0	The LOCKE interrupt is disabled
1	The LOCKE interrupt is enabled

Bit 1 – PROGE Programming Error Interrupt Clear
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the PROGE interrupt enable.

Value	Description
0	The PROGE interrupt is disabled
1	The PROGE interrupt is enabled

Bit 0 – DONE NVM Done Interrupt Enable

Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the DONE interrupt enable.

Value	Description
0	The DONE interrupt is disabled
1	The DONE interrupt is enabled

29.6.6 Interrupt Enable Set

Name: INTENSET
Offset: 0x10
Reset: 0x00
Property: PAC Write-Protection, Write-Mix-Secure

Important: For **PIC32CM LS00/LS60 Non-Secure** accesses, write accesses (W*) are allowed only if Non-Secure Write is set in the NONSEC register.

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
			NSCHK	KEYE	NVME	LOCKE	PROGE	DONE
Access			RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW
Reset			0	0	0	0	0	0

Bit 5 – NSCHK Non-secure Check Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit sets the NSCHK interrupt enable.

Value	Description
0	The NSCHK interrupt is disabled
1	The NSCHK interrupt is enabled

Bit 4 – KEYE Key Error Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit sets the KEYE interrupt enable.

Value	Description
0	The KEYE interrupt is disabled
1	The KEYE interrupt is enabled

Bit 3 – NVME NVM internal Error Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit sets the NVME interrupt enable.

Value	Description
0	The NVME interrupt is disabled
1	The NVME interrupt is enabled

Bit 2 – LOCKE Lock Error Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit sets the LOCKE interrupt enable.

Value	Description
0	The LOCKE interrupt is disabled
1	The LOCKE interrupt is enabled

Bit 1 – PROGE Programming Error Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit sets the PROGE interrupt enable.

Value	Description
0	The PROGE interrupt is disabled
1	The PROGE interrupt is enabled

Bit 0 – DONE NVM Done Interrupt Enable

Writing a '0' to this bit has no effect.
Writing a '1' to this bit sets the DONE interrupt enable.

Value	Description
0	The DONE interrupt is disabled
1	The DONE interrupt is enabled

29.6.7 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x14
Reset: 0x00
Property: Write-Mix-Secure

Important: For **PIC32CM LS00/LS60 Non-Secure** accesses, write accesses (W*) are allowed only if Non-Secure Write is set in the NONSEC register.

Bit	7	6	5	4	3	2	1	0
			NSCHK	KEYE	NVME	LOCKE	PROGE	DONE
Access			RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW
Reset			0	0	0	0	0	0

Bit 5 – NSCHK Non-Secure Check
This flag is set when the NONSEC register is changed and the new value differs from the NSCHK value.
This bit can be cleared by writing a '1' to its bit location.

Value	Description
0	The NONSEC configuration has not changed since last clear.
1	At least one change has been made to the NONSEC configuration since the last clear.

Bit 4 – KEYE Key Error
This flag is set when a key write-protected register has been accessed in write with a bad key. A one indicates that at least one write access has been discarded.
This bit can be cleared by writing a '1' to its bit location.

Value	Description
0	No key error occured since the last clear.
1	At least one key error occured since the last clear.

Bit 3 – NVME NVM internal Error
This flag is set on the occurrence of a NVM internal error.
This bit can be cleared by writing a '1' to its bit location.

Value	Description
0	No NVM internal error has happened since this bit was last cleared.
1	At least one NVM internal error has happened since this bit was last cleared.

Bit 2 – LOCKE Lock Error
This flag is set on the occurrence of a LOCKE error.
This bit can be cleared by writing a '1' to its bit location.

Value	Description
0	No programming of any locked lock region has happened since this bit was last cleared.
1	Programming of at least one locked lock region has happened since this bit was last cleared.

Bit 1 – PROGE Programming Error
This flag is set on the occurrence of a PROGE error.
This bit can be cleared by writing a '1' to its bit location.

Value	Description
0	No invalid commands or bad keywords were written in the NVM Command register since this bit was last cleared.
1	An invalid command and/or a bad keyword was/were written in the NVM Command register since this bit was last cleared.

Bit 0 – DONE NVM Command Done

This bit can be cleared by writing a one to its bit location

Value	Description
0	The NVM controller has not completed any commands since the last clear.
1	At least one command has completed since the last clear.

29.6.8 Status

Name: STATUS
Offset: 0x18
Reset: 0x00xx (x determined from latest Set DAL or Chip Erase command)
Property: Write-Secure

Important: For **PIC32CM LS00/LS60 Non-Secure** accesses, write accesses (W*) are allowed only if Non-Secure Write is set in NONSEC register.

Bit	15	14	13	12	11	10	9	8
	Reserved							
Access	R/R/R							
Reset	0							

Bit	7	6	5	4	3	2	1	0
				DALFUSE[1:0]		READY	LOAD	PRM
Access				R/R/R	R/R/R	R/R/R	R/R/R	R/R/R
Reset				x	x	0	0	0

Bit 15 – Reserved

Bits 4:3 – DALFUSE[1:0] DAL Fuse Value
This field is the current Debug Access Level fuse value.

⚠ CAUTION This bit field does not reflect the current Debug Access Level (DAL) if DALUN bit == 1 (NVMCTRL.SECCTRL) but the Debug Access Level which will be applied once DALUN == 0.

Value	Name	Description
0	DAL0	DAL = 0 : Access to very limited features.
1	DAL1	DAL = 1 (**PIC32CM LS00/LS60 only**): Access to all non-secure memory. Can debug non-secure CPU code.
2	DAL2	DAL = 2 : Access to all memory. Can debug Secure and non-secure CPU code.
3	-	Reserved

Bit 2 – READY NVM Ready

Value	Description
0	The NVM controller is busy programming or erasing.
1	The NVM controller is ready to accept a new command.

Bit 1 – LOAD NVM Page Buffer Active Loading
This bit indicates that the NVM page buffer has been loaded with one or more words. Immediately after an NVM load has been performed, this flag is set. It remains set until a page write or a page buffer clear (PBC) command is given.

Bit 0 – PRM Power Reduction Mode
This bit indicates the current NVM power reduction state. The NVM block can be set in power reduction mode in two ways: through the command interface or automatically when entering sleep with SLEEPPRM set accordingly.
PRM can be cleared in three ways: through AHB access to the NVM block, through the command interface (SPRM and CPRM) or when exiting sleep with SLEEPPRM set accordingly.

Value	Description
0	NVM is not in power reduction mode.

Value	Description
1	NVM is in power reduction mode.

29.6.9 Address

Name: ADDR
Offset: 0x1C
Reset: 0x00000000
Property: PAC Write-Protection, Secure

ADDR drives the hardware address to the NVM when a command is executed using CMDEX. This is a Byte aligned address. This register is automatically updated upon AHB writes to the page buffer.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	ARRAY[1:0]							
Access	RW/-/RW	RW/-/RW						
Reset	0	0						

Bit	15	14	13	12	11	10	9	8
	AOFFSET[15:8]							
Access	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW	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
	AOFFSET[7:0]							
Access	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW
Reset	0	0	0	0	0	0	0	0

Bits 23:22 – ARRAY[1:0] Array Select

Value	Name	Description
0x0	FLASH	Flash
0x1	DATAFLASH	Data Flash
0x2	NVMROWS	NVM Configuration Rows
0x3	-	Reserved

Bits 15:0 – AOFFSET[15:0] Array Offset
Address offset

29.6.10 Secure Region Unlock Bits

Name: SULCK
Offset: 0x20
Reset: x initially determined from NVM User Row after reset
Property: PAC Write-Protection, Write-Secure

Bit	15	14	13	12	11	10	9	8
	SLKEY[7:0]							
Access	W/-/W	W/-/W	W/-/W	W/-/W	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
						DS	AS	BS
Access						RW/R/RW	RW/R/RW	RW/R/RW
Reset						x	x	x

Bits 15:8 – SLKEY[7:0] Secure Unlock Key
When this bit group is written to the key value 0xA5, the write will be performed. If a value different from the key value is tried, the write will be discarded and INTFLAG.KEYE set.

Bit 2 – DS Secure Data Flash (DS region) Unlock Bit
Note: This bit is only available for PIC32CM LS00/LS60 and has no effect for PIC32CM LE00.

Value	Description
0	The Secure Data Flash region (DS region) is locked
1	The Secure Data Flash region (DS region) is not locked

Bit 1 – AS Secure Flash (AS region) Unlock Bit
Note: This bit is only available for PIC32CM LS00/LS60 and has no effect for PIC32CM LE00.

Value	Description
0	The Secure + NSC Flash (AS region) is locked
1	The Secure + NSC Flash (AS region) is not locked

Bit 0 – BS (Secure) Flash (BOOTPROT region) Unlock Bit
Note: For PIC32CM LE00 devices, the Secure Flash (BOOT region) corresponds to the Flash (BOOT region).

Value	Description
0	The Secure + NSC Flash (BOOTPROT/BS region) is locked
1	The Secure + NSC Flash (BOOTPROT/BS region) is not locked

29.6.11 Non-Secure Region Unlock Bits

Name: NSULCK
Offset: 0x22
Reset: x initially determined from NVM User Row after reset
Property: PAC Write-Protection, Write-Mix-Secure

Important: For **PIC32CM LS00/LS60 Non-Secure** accesses, write accesses (W*) are allowed only if Non-Secure Write is set in the NONSEC register.

Bit	15	14	13	12	11	10	9	8
	NSLKEY[7:0]							
Access	W/W*/W	W/W*/W	W/W*/W	W/W*/W	W/W*/W	W/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
						DNS	ANS	
Access						RW/RW*/RW	RW/RW*/RW	
Reset						x	x	

Bits 15:8 – NSLKEY[7:0] (Non-Secure) Unlock Key
When this bit group is written to the key value 0xA5, the write will be performed. If a value different from the key value is tried, the write will be discarded and INTFLAG.KEYE set.

Bit 2 – DNS (Non-Secure) Data Flash (DNS region) Unlock Bit
Note: For **PIC32CM LE00** devices, the Non-Secure Data Flash region corresponds to the entire Data Flash region.

Value	Description
0	The Non-Secure Data Flash region (DNS region) is locked
1	The Non-Secure Data Flash region (DNS region) is not locked

Bit 1 – ANS (Non-Secure) Flash (ANS region) Unlock Bit
Note: For **PIC32CM LE00** devices, the Non-Secure Flash (APPLICATION region) corresponds to the Flash (APPLICATION region).

Value	Description
0	The Non-Secure Flash (ANS region) is locked
1	The Non-Secure Flash (ANS region) is not locked

29.6.12 NVM Parameter

Name: PARAM
Offset: 0x24
Reset: x determined from (Data) Flash Memory Parameters tables
Property: Write-Secure

Bit	31	30	29	28	27	26	25	24
	DFLASHP[11:4]							
Access	R/R/R	R/R/R	R/R/R	R/R/R	R/R/R	R/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
	DFLASHP[3:0]					PSZ[2:0]		
Access	R/R/R	R/R/R	R/R/R	R/R/R		R/R/R	R/R/R	R/R/R
Reset	x	x	x	x		x	x	x

Bit	15	14	13	12	11	10	9	8
	FLASHP[15:8]							
Access	R/R/R	R/R/R	R/R/R	R/R/R	R/R/R	R/R/R	R/R/R	R/R/R
Reset	x	x	x	x	x	x	x	x

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

Bits 31:20 – DFLASHP[11:0] Data FLASH area Pages
Indicates the number of pages in the Data FLASH array.

Bits 18:16 – PSZ[2:0] Page Size
Indicates the page size. Not all devices of the device families will provide all the page sizes indicated in the table.

Value	Name	Description
0x0	8	8 bytes
0x1	16	16 bytes
0x2	32	32 bytes
0x3	64	64 bytes
0x4	128	128 bytes
0x5	256	256 bytes
0x6	512	512 bytes
0x7	1024	1024 bytes

Bits 15:0 – FLASHP[15:0] FLASH Pages
Indicates the number of pages in the FLASH array.

29.6.13 Data Flash Scramble Control

Name: DSCC
Offset: 0x30
Reset: 0x00000000
Property: PAC Write-Protection, Secure, Enable-Protected

Important: This register is only available for **PIC32CM LS00/LS60** and has no effect for **PIC32CM LE00**.

Bit	31	30	29	28	27	26	25	24
			DSCKEY[29:24]					
Access			W/-/W	W/-/W	W/-/W	W/-/W	W/-/W	W/-/W
Reset			0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	DSCKEY[23:16]							
Access	W/-/W	W/-/W	W/-/W	W/-/W	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
	DSCKEY[15:8]							
Access	W/-/W	W/-/W	W/-/W	W/-/W	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
	DSCKEY[7:0]							
Access	W/-/W	W/-/W	W/-/W	W/-/W	W/-/W	W/-/W	W/-/W	W/-/W
Reset	0	0	0	0	0	0	0	0

Bits 29:0 – DSCKEY[29:0] Data Flash Scramble Key
This key value is used for address and data scrambling of the Secure Data Flash. After reset the key is 0. When written, the new value in the register is an XOR of the value written and the previous value of DSCC.DSCKEY. This register is write only and will always read back as zero.
This register is Enable-Protected with SECCTRL.DSCEN meaning that it can't be modified when DSCEN=1 otherwise a PAC error is generated.
Updated DSCC.DSCKEY contents <- DSCC.DSCKEY XOR value written.

29.6.14 Security Control

Name: SECCTRL
Offset: 0x34
Reset: x/y initially determined after Reset from NVM User Row (UROW) / BOCOR.SECCFGLOCK
Property: PAC Write-Protection, Secure

Important: if BOCOR.SECCFGLOCK = 0 after exiting the Boot ROM:

- The secure boot flash code, before exiting, has the responsibility to lock the NVMCTRL security configurations by clearing the NVMCTRL.SECCTRL.SCFGWEN bit.
- Write accesses (W*) are allowed.

Bit	31	30	29	28	27	26	25	24
	KEY[7:0]							
Access	W/-/W	W/-/W	W/-/W	W/-/W	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
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
						TEROW[2:0]		
Access						RW/-/RW	RW/-/RW	RW/-/RW
Reset						0	0	0

Bit	7	6	5	4	3	2	1	0
		DXN	DALUN	SCFGWEN	DSCEN	SILACC		TAMPEEN
Access		R/-/RW*	-/-/RW*	RW/-/RW*	RW/-/RW	RW/-/RW		RW/-/RW
Reset		x	y	y	0	0		0

Bits 31:24 – KEY[7:0] Write Key
When this bit group is written to the key value 0xA5, the write will be performed. If a value different from the key value is tried, the write will be discarded and INTFLAG.KEYE set.

Bits 10:8 – TEROW[2:0] Tamper Erase Row
Row address of the row in data space to be erased on RTC tamper event.

Bit 6 – DXN Data eXecute Never
Note: This bit field is only available for **PIC32CM LS00/LS60** and has no effect for **PIC32CM LE00**.

This bit status is loaded from UROW during Boot ROM execution.

Value	Description
0	Execution out of Data Flash is authorized.
1	Execution out of Data Flash is not authorized.

Bit 5 – DALUN DAL Unlock
Note: This bit field is only available for **PIC32CM LS00/LS60** and has no effect for **PIC32CM LE00**.

Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the DAL Unlock bit.
After Boot ROM execution:

- DALUN=0 if BOCOR.SECCFGLOCK = 1

- DALUN=1 if BOCOR.SECCFGLOCK = 0

Important: If DALUN=1, the secure software code of the Flash BOOT region MUST clear this bit before passing control on to the secure software code of the Flash APPLICATION region.

Value	Description
0	DAL is forced to the DSU STATUSB.DAL value. DAL can only be set to lower values. **DALUN cannot be written until the next erase.**
1	DAL is forced to DAL0. All SDAL commands are allowed.

Bit 4 – SCFGWEN Security Configuration Write Enable

Note: This bit field is only available for **PIC32CM LS00/LS60** and has no effect for **PIC32CM LE00**.

After Boot ROM execution, this bit is:

- Cleared if BOCOR.SECCFGLOCK = 1
- Set if BOCOR.SECCFGLOCK = 0

Important: If SCFGWEN = 1, the secure software code of the Flash BOOT region has the responsibility to clear this bit before passing control on to the secure software code of the Flash APPLICATION region in order to lock the NVMCTRL security configurations.

Value	Description
0	SCFGB, SCFGAD and SECCTRL.SCFGWEN cannot be written until the next reset.
1	SCFGB, SCFGAD and SECCTRL.SCFGWEN can be written.

Bit 3 – DSCEN Data Flash Scramble Enable

Note: This bit field is only available for **PIC32CM LS00/LS60** and has no effect for **PIC32CM LE00**.

Value	Description
0	Secure Data FLASH is not scrambled.
1	Secure Data FLASH is scrambled.

Bit 2 – SILACC Silent Access

Value	Description
0	Data in Tamper Erase Row is not mapped as differential data.
1	Data in Tamper Erase Row is mapped as differential data.

Bit 0 – TAMPEEN Tamper Erase Enable

Value	Description
0	RTC tamper event has no effect.
1	RTC tamper event triggers a Tamper Erase.

29.6.15 Secure Boot Configuration

Name: SCFGB
Offset: 0x38
Reset: 'x' initially determined from NVM Boot Configuration Row (BOCOR) after Reset.
Property: -

This register is loaded from BOCOR during Boot ROM execution.

Important: if BOCOR.SECCFGLOCK == 0 after exiting the Boot ROM:

- The secure software code of the Flash BOOT region, before passing control on to the secure software code of the Flash APPLICATION region, **MUST** lock the NVMCTRL memory security configurations by clearing the NVMCTRL.SECCTRL.SCFGWEN bit.
- Write accesses (W*) are allowed.

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
							BCWEN	BCREN
Access							R/-/RW*	R/-/RW*
Reset							x	x

Bit 1 – BCWEN Boot Configuration Row Write Enable

Value	Description
0	BOCOR is not writable.
1	BOCOR is writable.

Bit 0 – BCREN Boot Configuration Row Read Enable

Value	Description
0	BOCOR is not readable.
1	BOCOR is readable.

29.6.16 Secure Application and Data Configuration

Name: SCFGAD
Offset: 0x3C
Reset: x initially determined from NVM User Row after reset
Property: -

This register is loaded from UROW during Boot ROM execution.

Important: if BOCOR.SECCFGLOCK == 0 after exiting the Boot ROM:

- The secure software code of the Flash BOOT region, before passing control on to the secure software code of the Flash APPLICATION region, **MUST** lock the NVMCTRL memory security configurations by clearing the NVMCTRL.SECCTRL.SCFGWEN bit.
- Write accesses (W*) are allowed.

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
								URWEN
Access								R/-/RW*
Reset								x

Bit 0 – URWEN User Row Write Enable

Value	Description
0	UROW is not writable.
1	UROW is writable.

29.6.17 Non-secure Write Enable

Name: NONSEC
Offset: 0x40
Reset: 0x00000001
Property: PAC Write-Protection, Write-Secure

This register allows the non-secure writes to the non-secure APB alias and also non-secure AHB writes to the Page Buffer.

Important: This register is only available for **PIC32CM LS00/LS60** and has no effect for **PIC32CM LE00**.

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
								WRITE
Access								RW/R/RW
Reset								1

Bit 0 – WRITE Non-secure Write Enable
Non-secure APB alias write enable, non-secure AHB writes to non-secure regions enable

Value	Description
0	The non-secure APB alias is not writable, AHB secure or non-secure writes to non-secure regions (Page Buffer) return a hardfault.
1	No restriction.

29.6.18 Non-secure Write Enable Check

Name: NSCHK
Offset: 0x44
Reset: 0x00000001
Property: PAC Write-Protection

Important: This register is only available for **PIC32CM LS00/LS60** and has no effect for **PIC32CM LE00**.

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
								WRITE
Access								RW/RW/RW
Reset								1

Bit 0 – WRITE Non-secure Write Transition Select

This bitfield selects whether to generate a NSCHK interrupt on a NONSEC.WRITE rising of falling transition.

Value	Description
0	INTFLAG.NSCHK rises if NONSEC.WRITE transitions from 0 to 1.
1	INTFLAG.NSCHK rises if NONSEC.WRITE transitions from 1 to 0.

30. TrustRAM (TRAM)

30.1 Overview

The TrustRAM (TRAM) is the controller interface for a 512-byte security RAM. This RAM is intended for volatile secret data. The TRAM is capable of performing address map scrambling as well as data scrambling for both write and read access to the security RAM.

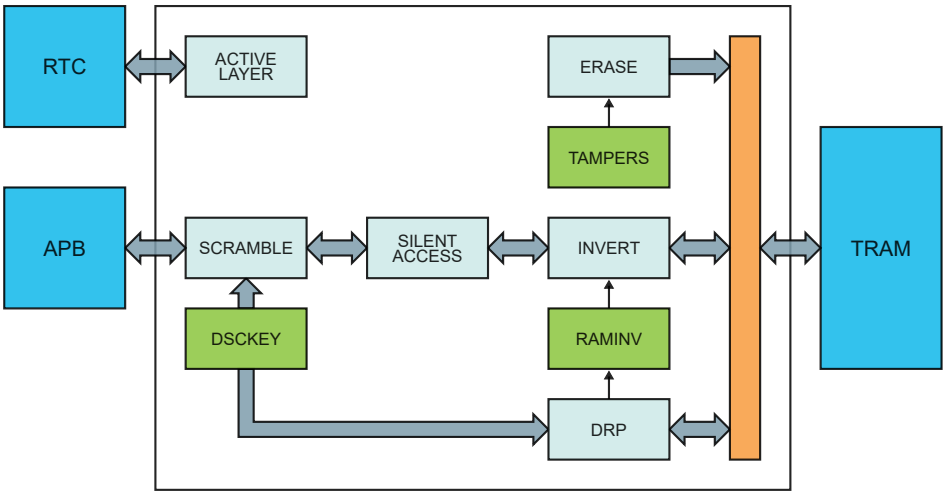
The TRAM can execute two automated tasks that are triggered by external events: remanence prevention and erase. When a remanence periodic event occurs, the physical data stored in the RAM is inverted in order to prevent physical “burn-in” signatures. When a tamper event occurs, the TRAM executes a full erase of the control signals as well as the data in the security RAM. Both automated tasks do not require CPU interaction and can be performed in all sleep modes.

30.2 Features

- Address scrambling to the RAM
- Data scrambling to/from the RAM
- Silent access of data
- Data remanence prevention
- Anti-tamper Active Shield on physical TrustRAM
- Full erasure of scramble key and RAM data on tamper detection

30.3 Block Diagram

Figure 30-1. Block Diagram



30.4 Peripheral Dependencies

Table 30-1. TRAM Configuration Summary

Peripheral name	Base address	NVIC IRQ Index	MCLK AHB/APB Clocks	PAC Peripheral Identifier Index (PAC.WRCTRL)	Power Domain (PM.STDBYCFG)
TRAM	0x42005400	70: DRP, ERR	CLK_TRAM_AHB	85	PDSW

30.5 Functional Description

30.5.1 Principle of Operation

System bus transactions from the CPU to the security RAM undergoes a scrambling routine. Both address and data buses information are modified through an algorithm determined by a scrambling key. This is performed on both write and read transactions.

30.5.2 Basic Operation

30.5.2.1 Initialization

The following bits are enable-protected, meaning that they can only be written when the TRAM is disabled (CTRLA.ENABLE is zero):

- Tamper Erase bit in the Control A register (CTRLA.TAMPERS)
- Data Remanence Protection bit in the Control A register (CTRLA.DRP)
- Silent Access bit in the Control A register (CTRLA.SILACC)

The following registers are enable-protected:

- Data Scramble Control register (DSCC)

Enable-protected bits in the CTRLA register can be written at the same time as CTRLA.ENABLE is written to one, but not at the same time as CTRLA.ENABLE is written to zero.

Enable-protection is denoted by the Enable-Protected property in the register description.

30.5.2.2 Enabling, Disabling and Resetting

The TRAM is enabled by writing a one to the Enable bit in the Control A register (CTRLA.ENABLE). The TRAM is disabled by writing a zero to CTRLA.ENABLE.

The TRAM is reset by writing a one to the Software Reset bit in the Control A register (CTRLA.SWRST). All registers in the TRAM will be reset to their initial state, and the TRAM will be disabled. All data in the security RAM will be cleared to ‘0’.

30.5.2.3 Scrambling

The Data Scramble Control (DSCC) must be configured before the CTRLA.ENABLE is set. These settings cannot be changed while the module is enabled.

The scrambling logic is enabled by writing one to the enable bit in the Data Scramble Control register (DSCC.DSCEN). Scrambling is disabled by writing a zero to DSCC.DSCEN. Writing a zero to CTRLA.ENABLE will also disable the scrambling, but will not clear the DSCC.DSCEN bit.

30.5.2.4 Silent Access

When enabled (CTRLA.SILACC = 1), the silent access feature allows to store data and their 1's complement in the TRAM thus reducing the overall data reading noise, as always the same number of '0' and '1' will be read for each read access.
Note: Silent access bit must be configured before CTRLA.ENABLE is set. This setting cannot be changed while the module is enabled.

When silent access is enabled , the logical size of the TRAM is divided by two to store each byte of data and its 1's complement (CompData) in the whole physical TRAM size .

The differential write and silent read processes are transparent and managed by the TRAM controller , The TRAM executes the following process on data access :

• When the CPU writes to the TRAM, the data and its bit-wise invert are stored into the RAM.

• When the CPU reads from the TRAM, both the data and its bit-wise invert are retrieved from the RAM. If the TRAM cannot verify that both values complement each other, a bus error is returned.

When silent access is enabled , the data stored in the TRAM must be accessed using the logical mapping shown in TRAM logical mapping table:

Table 30-2. TRAM logical mapping (CTRLA.SILACC=1)

Byte 3	Byte 2	Byte 1	Byte 0	TRAM Register
Data	Data	Data	Data	RAM0
...	...	...	...	...
Data	Data	Data	Data	RAM[63]
Reserved	Reserved	Reserved	Reserved	RAM[64]
...	...	...	...	...
Reserved	Reserved	Reserved	Reserved	RAM[127]

Note:
Only 8-bit (byte) access and 16-bit (half-word) access are supported in this mode. 32-bit (word) write accesses are ignored and 32-bit (word) read accesses return 0.

All accesses to the reserved area of the TRAM are discarded and generate a bus error.

The physical mapping of the TRAM when silent access is enabled, is represented in TRAM physical mapping table:

Table 30-3. TRAM physical mapping (CTRLA.SILACC=1)

Byte 3	Byte 2	Byte 1	Byte 0	TRAM Register
CompData	Data	CompData	Data	RAM0
CompData	Data	CompData	Data	RAM1
CompData	Data	CompData	Data	RAM2
...	...	...	...	...
CompData	Data	CompData	Data	RAM[127]

When used in addition of data scrambling , the TRAM controller automatically manage both scrambling and differential data storage operation on specific data access.

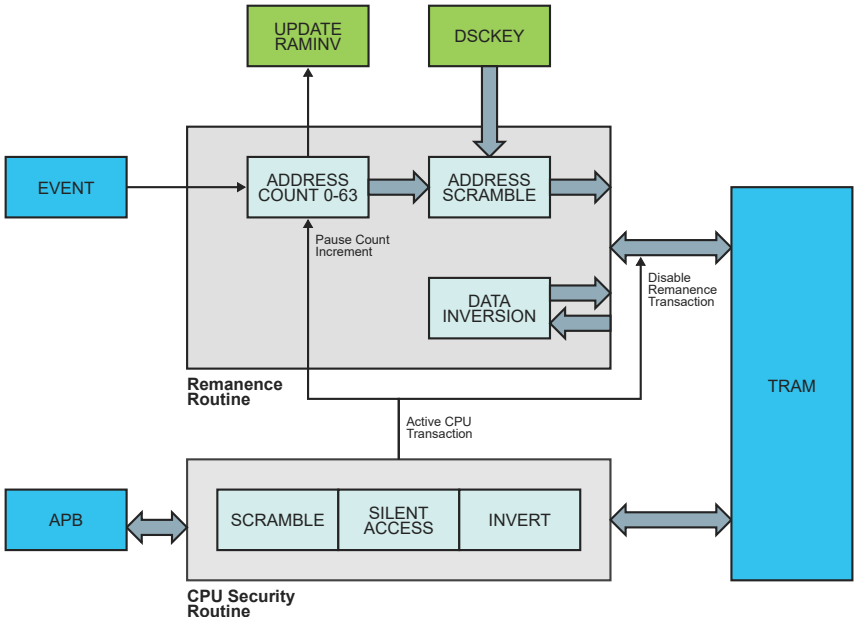
30.5.2.5 Data Remanence Prevention

Data remanence prevention bit (CTRLA.DRP) must be configured before CTRLA.ENABLE is set. This setting cannot be changed while the module is enabled. When this feature is enabled, the RTC Periodic Interval Daily Event (RTC_PERD) will trigger the automated data remanence routine. An internal counter will count from 0 to 63 and serves as the address access bus to the security RAM. For every address iteration, the TRAM reads the word data from the security RAM, inverts the value and writes back to the same address. To prevent linear access to the security RAM, the remanence address value is scrambled using the same protocols as a CPU address scramble. After remanence has updated all address locations, the routine will end by toggling the RAM inversion status bit (STATUS.RAMINV).

Note:
Data Remanence Prevention is connected directly from the RTC to the TRAM, without going through the Event System.

Data remanence is a low-priority routine. If the CPU attempts to access the security RAM while remanence is active, the routine is temporarily paused until the CPU access is completed. If a tamper full erase event is detected, the remanence routine is aborted and the internal address counter will reset to 0.

Figure 30-2. Remanence Routine



30.5.2.6 Tamper Full Erase

Tamper full erase bit (CTRLA.TAMPERS) must be configured before CTRLA.ENABLE is set. This setting cannot be changed while the module is enabled. When this feature is enabled, the RTC Tamper Event (RTC_TAMPER) will trigger the full erase equivalent to a TRAM software reset and the reset of the Data Scramble Key (DSCC.DSCKEY) register. All TRAM registers are reverted to the default reset value. Data inside the security RAM is written to '0' for all address locations.

Note: Tamper events are connected directly from the RTC to the TRAM, without going through the Event System.

The tamper full erase routine operates at the highest priority. If a remanence routine executing when a tamper full erase occurs, the remanence routine is immediately terminated. If the CPU attempts to write a new scramble key at the same time the tamper key erase routine is active, the CPU data is ignored, but no bus error will occur. If a CPU security routine access is requested during a tamper full erase, the CPU transaction will be ignored and treated as a bus error similar to accessing the module during a software reset.

Important: In STANDBY low power mode, it is mandatory to enable the dynamic power gating feature (STDBYCFG.DPGPDSW) to ensure TrustRAM erasing when the power domain PDSW is in a retention state.

30.5.3 Interrupts

The TRAM has the following interrupt sources:

- Data Remanence Prevention (DRP): Indicates that the data remanence prevention routine has ended
- Data Read Error (ERR): Indicates when there is a RAM readout error

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 TRAM is reset. See 22.6.6. INTFLAG 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. Refer to Nested Vector Interrupt Controller for details. 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. Refer to Nested Vector Interrupt Controller for details.

30.5.4 Sleep Mode Operation

The TRAM continues to operate during sleep. When it receives events from the Event System, it will request its own clock in order to perform the requested operation.

An interrupt request will be generated after the wake-up if the Interrupt Controller is configured accordingly. Otherwise the CPU will wake up directly, without triggering an interrupt. In this case, the CPU will continue executing from the instruction following 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 EVSYS – Event System for more information.

30.5.5 Synchronization

Some registers (or bit fields within a register) require synchronization when read and/or written.

Synchronization is denoted by the "Read-Synchronized" (or "Read-Synchronized Bits”) and/or "Write-Synchronized" (or "Write-Synchronized Bits”) property in each individual register description.

For more details, refer to Register Synchronization.

30.6 Register Summary

Refer to the Registers Description section for more details on register properties and access permissions.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRLA	7:0	SILACC	DRP		TAMPERS			ENABLE	SWRST
0x01 ... 0x03	Reserved									
0x04	INTENCLR	7:0							DRP	ERR
0x05	INTENSET	7:0							DRP	ERR
0x06	INTFLAG	7:0							DRP	ERR
0x07	STATUS	7:0							DRP	RAMINV
0x08	SYNCBUSY	31:24								
		23:16								
		15:8								
		7:0							ENABLE	SWRST
0x0C	DSCC	31:24	DSCEN		DSCKEY[29:24]					
		23:16	DSCKEY[23:16]							
		15:8	DSCKEY[15:8]							
		7:0	DSCKEY[7:0]							
0x10 ... 0x01FF	Reserved									
0x0200	RAM0	31:24	DATA[31:24]							
		23:16	DATA[23:16]							
		15:8	DATA[15:8]							
		7:0	DATA[7:0]							
...										
0x03FC	RAM127	31:24	DATA[31:24]							
		23:16	DATA[23:16]							
		15:8	DATA[15:8]							
		7:0	DATA[7:0]							

30.6.1 Control A

Name: CTRLA
Offset: 0x000
Reset: 0x00
Property: PAC Write Protection, Enable-Protected Bits, Write-Synchronized Bits

Bit	7	6	5	4	3	2	1	0
	SILACC	DRP		TAMPERS			ENABLE	SWRST
Access	R/W	R/W		R/W			R/W	R/W
Reset	0	0		0			0	0

Bit 7 – SILACC Silent Access
Enables differential storage of data.
Note: This bit is enable-protected. This bit is not synchronized.

Value	Description
0	Silent access is disabled.
1	Silent access is enabled.

Bit 6 – DRP Data Remanence Prevention
Enables periodic Data Remanence Prevention in TrustRAM.
Note: This bit is enable-protected. This bit is not synchronized.

Value	Description
0	Data remanence prevention is disabled.
1	Data remanence prevention is enabled.

Bit 4 – TAMPERS Tamper Erase
Auto-erases TrustRAM and DSCC.DSCKEY on tamper event.
Note: This bit is enable-protected. This bit is not synchronized.

Value	Description
0	Tamper erase is disabled.
1	Tamper erase is enabled.

Bit 1 – ENABLE Enable
Notes:
1. This bit is write-synchronized: SYNCBUSY.ENABLE must be checked to ensure the CTRLA.ENABLE synchronization is complete.
2. This bit is not enable-protected.

Value	Description
0	The TRAM is disabled.
1	The TRAM is enabled.

Bit 0 – SWRST Software Reset
Writing a zero to this bit has no effect.
Writing a one to this bit resets all registers in the TRAM to their initial state, and the TRAM will be disabled. This bit can also be set via hardware when a tamper occurs while CTRLA.TAMPERS is set.
Writing a one to CTRLA.SWRST will always take precedence, meaning that all other writes in the same write operation will be discarded.

Notes:

1. When the CTRLA.SWRST is written, the user must poll SYNCBUSY.SWRST bit to know when the reset operation is complete.
2. During a SWRST, access to registers/bits without SWRST are disallowed until SYNCBUSY.SWRST is cleared by hardware.
3. This bit is not enable protected.

Value	Description
0	There is no reset operation ongoing.
1	The reset operation is ongoing.

30.6.2 Interrupt Enable Clear

Name: INTENCLR
Offset: 0x004
Reset: 0x00
Property: PAC Write-Protection

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

Bit 1 – DRP Data Remanence Prevention Complete Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will clear the Data Remanence Prevention Complete Interrupt Enable bit, which disables the data remanence prevention complete interrupt.

Value	Description
0	Data remanence prevention complete interrupt is disabled.
1	Data remanence prevention complete interrupt is enabled.

Bit 0 – ERR TrustRAM Read Error Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will clear the TrustRAM Read Error Interrupt Enable bit, which disables the TrustRAM read error interrupt.

Value	Description
0	TrustRAM read error interrupt is disabled.
1	TrustRAM read error interrupt is enabled.

30.6.3 Interrupt Enable Set

Name: INTENSET
Offset: 0x005
Reset: 0x00
Property: PAC Write-Protection

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

Bit 1 – DRP Data Remanence Prevention Complete Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will set the Data Remanence Prevention Complete Interrupt Enable bit, which enables the data remanence prevention complete interrupt.

Value	Description
0	Data remanence prevention complete interrupt is disabled.
1	Data remanence prevention complete interrupt is enabled.

Bit 0 – ERR TrustRAM Read Error Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will set the TrustRAM Read Error Interrupt Enable bit, which enables the TrustRAM read error interrupt.

Value	Description
0	TrustRAM read error interrupt is disabled.
1	TrustRAM read error interrupt is enabled.

30.6.4 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x006
Reset: 0x00
Property: -

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

Bit 1 – DRP Data Remanence Prevention Complete Interrupt

This flag is set when the data remanence prevention routine has completed, and an interrupt request will be generated if INTENSET.DRP is one.
Writing a zero to this bit has no effect.
Writing a one to this bit clears the data remanence prevention complete interrupt flag.

Value	Description
0	Data remanence prevention complete interrupt is disabled.
1	Data remanence prevention complete interrupt is enabled.

Bit 0 – ERR TrustRAM Read Error Interrupt

This flag is set when an error is detected in the TrustRAM readout, and an interrupt request will be generated if INTENSET.ERR is one.
Writing a zero to this bit has no effect.
Writing a one to this bit clears the TrustRAM read error interrupt flag.

Value	Description
0	TrustRAM read error interrupt is disabled.
1	TrustRAM read error interrupt is enabled.

30.6.5 Status

Name: STATUS
Offset: 0x007
Reset: 0x00
Property: -

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

Bit 1 – DRP Data Remanence Prevention Routine

This bit identifies if the data remanence prevention routine is running.

Value	Description
0	The data remanence prevention routine is not running.
1	The data remanence prevention routine is running.

Bit 0 – RAMINV RAM Inversion Bit

This bit identifies if the TrustRAM bit values are inverted.

Value	Description
0	The TrustRAM physical bit information is normal.
1	The TrustRAM physical bit information is inverted.

30.6.6 Synchronization Busy

Name: SYNCBUSY
Offset: 0x008
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

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

This bit will set in two ways:

- Writing ‘1’ to CTRLA.SWRST
- A tamper event occurs when CTRLA.TAMPERS = ‘1’

Value	Description
0	Write synchronization for CTRLA.SWRST bit is complete.
1	Write synchronization for CTRLA.SWRST bit is ongoing.

30.6.7 Data Scramble Control

Name: DSCC
Offset: 0x00C
Reset: 0x00000000
Property: PAC Write-Protected, Enable-Protected

Bit	31	30	29	28	27	26	25	24
	DSCEN		DSCKEY[29:24]					
Access	R/W		W	W	W	W	W	W
Reset	0		0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	DSCKEY[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
	DSCKEY[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
	DSCKEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit 31 – DSCEN Data Scramble Enable

Value	Description
0	TrustRAM is not scrambled.
1	TrustRAM is scrambled.

Bits 29:0 – DSCKEY[29:0] Data Scramble Key
The key value used for data scrambling. Any value written to this field is XOR'ed with the previous data. Writing '1' to CTRLA.SWRST will reset this field to 0. These bits will always return zero when read.

30.6.8 Security RAM n

Name: RAM
Offset: 0x0200 + n*0x04 [n=0..127]
Reset: 0x00000000
Property: PAC Write-Protected

Access to the Security RAM is only permitted when CTRLA.ENABLE = 1.

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] RAM Data

31. I/O Pin Controller (PORT)

31.1 Overview

The IO Pin Controller (PORT) controls the I/O pins of the device. The I/O pins are organized in a series of groups, collectively referred to as a PORT group. Each PORT group can have up to 32 pins that can be configured and controlled individually or as a group. The number of PORT groups on a device may depend on the package/number of pins. Each pin may either be used for general-purpose I/O under direct application control or be assigned to an embedded device peripheral. When used for general-purpose I/O, each pin can be configured as input or output, with highly configurable driver and pull settings. Each pin can be defined as secured or non-secured, where secured pins can only be handled by secure accesses.

All I/O pins have true read-modify-write functionality when used for general-purpose I/O; the direction or the output value of one or more pins may be changed (set, reset or toggled) explicitly without unintentionally changing the state of any other pins in the same port group by a single, atomic 8-, 16- or 32-bit write.

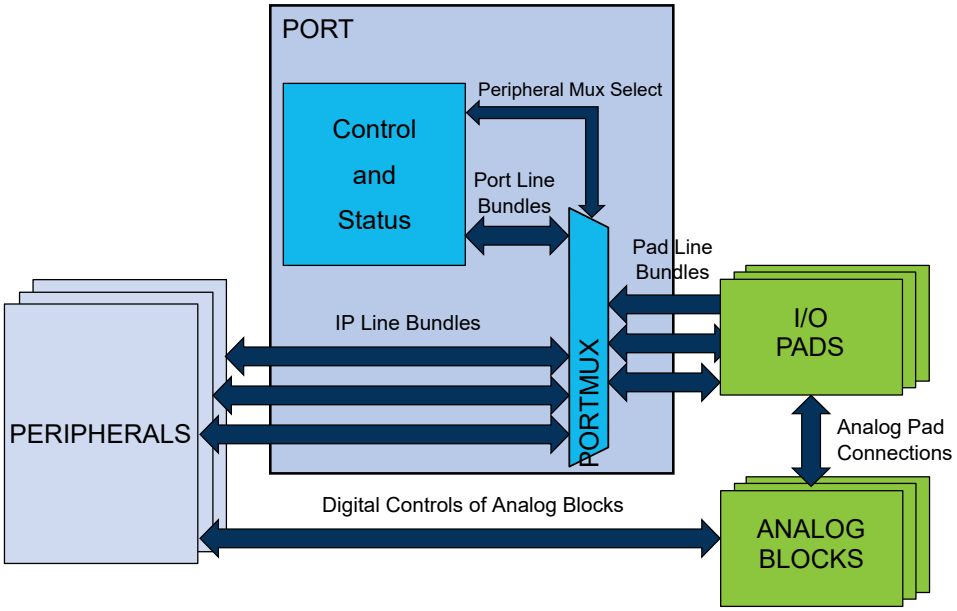
The PORT is connected to the high-speed bus matrix through an AHB/APB bridge. The Pin Direction, Data Output Value and Data Input Value registers may also be accessed using the low-latency CPU local bus (IOBUS; Arm single-cycle I/O port).

31.2 Features

- Selectable input and output configuration for each individual pin
- Software-controlled multiplexing of peripheral functions on I/O pins
- Flexible pin configuration through a dedicated Pin Configuration register
- Configurable output driver and pull settings:
  - Totem-pole (push-pull)
  - Pull configuration
  - Driver strength
- Configurable input buffer and pull settings:
  - Internal pull-up or pull-down
  - Input sampling criteria
  - Input buffer can be disabled if not needed for lower power consumption
- Input event:
  - Up to four input event pins for each PORT group
  - SET/CLEAR/TOGGLE event actions for each event input on output value of a pin
  - Can be output to pin
- Selectable secured or non-secured attribution for each individual pin **(PIC32CM LS00/LS60)**

31.3 Block Diagram

Figure 31-1. PORT Block Diagram



31.4 Signal Description

Table 31-1. Signal description for PORT

Signal name	Type	Description
Pxy	Digital I/O	General-purpose I/O pin y in group x

Refer to the Pinout for details on the pin mapping for this peripheral. One signal can be mapped on several pins.

31.5 Peripheral Dependencies

Table 31-2. PORT Configuration Summary

Peripheral name	Base address	NVIC IRQ Index	MCLK AHB/APB Clocks	PAC Peripheral Identifier Index (PAC.WRCTRL)	Events (EVSYS)		Power Domain (PM.STDBYCFG)
					Users (EVSYS.USER)	Generators (EVSYS.CHANNEL)	
PORT	0x40003000	14: NSCHK	CLK_PORT_APB	12	3-6 : EVU0-3	—	PDAO

31.6 Functional Description

The I/O lines of the PORT are mapped to pins of the physical device. The following naming scheme is used:

Each line bundle with up to 32 lines is assigned an identifier 'xy', with letter x=A, B, C… and two-digit number y=00, 01, …31. Examples: A24, C03.

PORT pins are labeled 'Pxy' accordingly, for example PA24, PC03. This identifies each pin in the device uniquely.

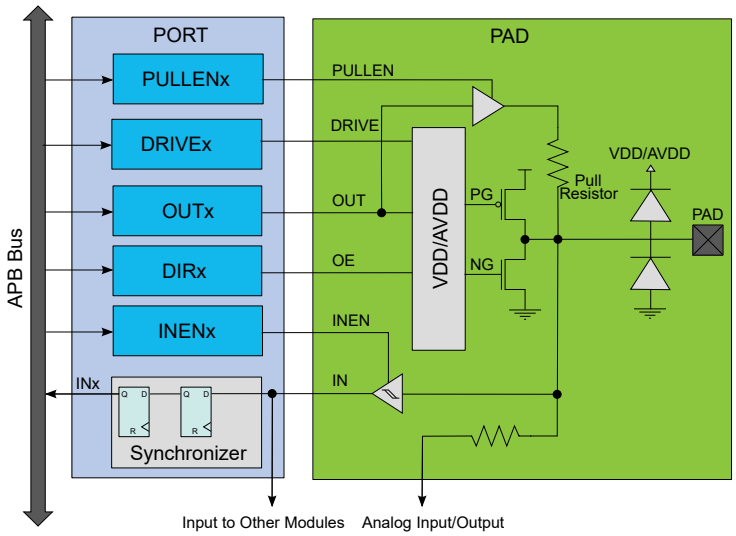
Each pin may be controlled by one or more peripheral multiplexer settings, which allow the pad to be routed internally to a dedicated peripheral function. When the setting is enabled, the selected peripheral has control over the output state of the pad, as well as the ability to read the current physical pad state. Refer to the Pinout for details.

Analog functions are connected directly between the analog blocks and the I/O pads using analog buses. However, selecting an analog peripheral function for a given pin will disable the corresponding digital features of the pad.

Each pin may be secured or non-secured, with secured pins only accessible by secure accesses.

Device-specific configurations may cause some lines (and the corresponding Pxy pin) not to be implemented.

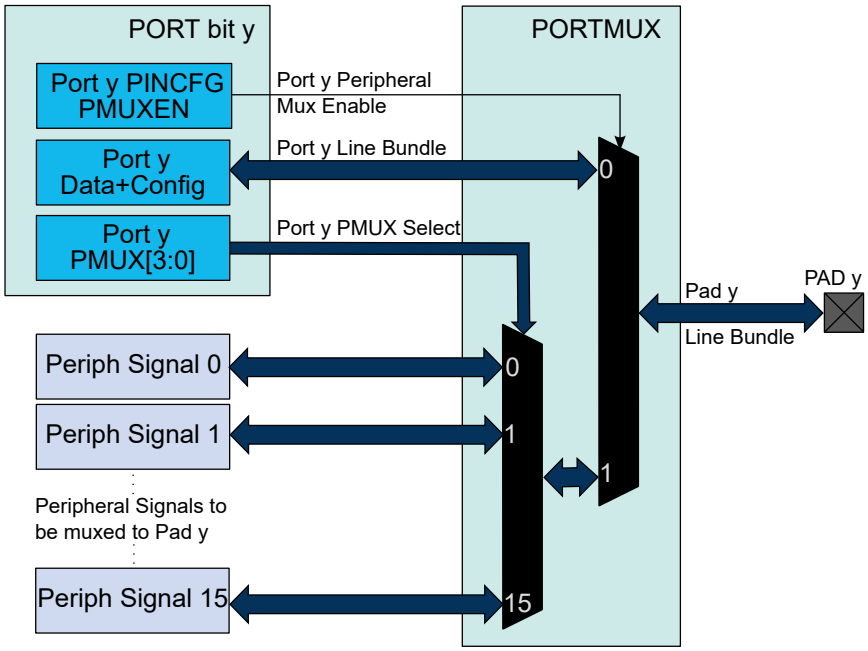
Figure 31-2. Overview of the PORT



31.6.1 Principle of Operation

Each PORT group of up to 32 pins is controlled by the registers in PORT, as described in the figure. These registers in PORT are duplicated for each PORT group, with increasing base addresses. The number of PORT groups may depend on the package/number of pins.

Figure 31-3. Overview of the peripheral functions multiplexing



The I/O pins of the device are controlled by PORT peripheral registers. Each port pin has a corresponding bit in the Data Direction (DIR) and Data Output Value (OUT) registers to enable that pin as an output and to define the output state.

The direction of each pin in a PORT group is configured by the DIR register. If a bit in DIR is set to '1', the corresponding pin is configured as an output pin. If a bit in DIR is set to '0', the corresponding pin is configured as an input pin.

When the direction is set as output, the corresponding bit in the OUT register will set the level of the pin. If bit y in OUT is written to '1', pin y is driven HIGH. If bit y in OUT is written to '0', pin y is driven LOW. Pin configuration can be set by Pin Configuration (PINCFGy) registers, with y=00, 01, ..31 representing the bit position.

The Data Input Value (IN) is set as the input value of a port pin with resynchronization to the PORT clock. To reduce power consumption, these input synchronizers are clocked only when system requires reading the input value. The value of the pin can always be read, whether the pin is configured as input or output. If the Input Enable bit in the Pin Configuration registers (PINCFGy.INEN) is '0', the input value will not be sampled.

In PORT, the Peripheral Multiplexer Enable bit in the PINCFGy register (PINCFGy.PMUXEN) can be written to '1' to enable the connection between peripheral functions and individual I/O pins. The Peripheral Multiplexing n (PMUXn) registers select the peripheral function for the corresponding pin. This will override the connection between the PORT and that I/O pin, and connect the selected peripheral signal to the particular I/O pin instead of the PORT line bundle.

The security attribution of each pin in a PORT group is configured by the NONSEC register. If a bit in the NONSEC register is set to '0', the corresponding pin is configured as a secured pin and can only be handled by secure accesses. If a bit in the NONSEC register is set to '1', the corresponding pin is configured as a non-secured pin. Only secure accesses are allowed to write to the NONSEC register.

31.6.2 Basic Operation

31.6.2.1 Initialization

After reset, all standard function device I/O pads are connected to the PORT with outputs tri-stated and input buffers disabled, even if there is no clock running, except:

1. PA30 pin: configured in peripheral mode with pull-up enabled (SWCLK peripheral function selected for debugger probe detection support).

31.6.2.2 Operation

Each I/O pin y can be controlled by the registers in PORT. Each PORT group has its own set of PORT registers, the base address of the register set for pin y is at byte address PORT + ([y] * 0x4). The index within that register set is [y].

Refer to the Pinout for details on available pin configuration and PORT groups.

Configuring Pins as Output

To use pin number y as an *output*, write bit y of the DIR register to '1'. This can also be done by writing bit y in the DIRSET register to '1' - this will avoid disturbing the configuration of other pins in that group. The y bit in the OUT register must be written to the desired output value.

Similarly, writing an OUTSET bit to '1' will set the corresponding bit in the OUT register to '1'. Writing a bit in OUTCLR to '1' will set that bit in OUT to zero. Writing a bit in OUTTGL to '1' will toggle that bit in OUT.

Configuring Pins as Input

To use pin y as an *input*, bit y in the DIR register must be written to '0'. This can also be done by writing bit y in the DIRCLR register to '1' - this will avoid disturbing the configuration of other pins in that group. The input value can be read from bit y in register IN as soon as the INEN bit in the Pin Configuration register (PINCFGy.INEN) is written to '1'.

By default, the input synchronizer is clocked only when an input read is requested. This will delay the read operation by two CLK_PORT cycles. To remove the delay, the input synchronizers for each PORT group of eight pins can be configured to be always active, but this will increase power consumption. This is enabled by writing '1' to the corresponding SAMPLINGn bit field of the CTRL register, see CTRL.SAMPLING for details.

Using Alternative Peripheral Functions

To use pin y as one of the available peripheral functions, the corresponding PMUXEN bit of the PINCFGy register must be '1'. The PINCFGy register for pin y is at byte offset (PINCFG0 + [y]).

The peripheral function can be selected by setting the PMUXO or PMUXE in the PMUXn register. The PMUXO/PMUXE is at byte offset PMUX0 + (y/2). The chosen peripheral must also be configured and enabled.

31.6.3 I/O Pin Configuration

The Pin Configuration register (PINCFGy) is used for additional I/O pin configuration. A pin can be set in a totem-pole or pull configuration.

As pull configuration is done through the Pin Configuration register, all intermediate PORT states during switching of pin direction and pin values are avoided.

The I/O pin configurations are described further in this chapter, and summarized in Table 31-3.

31.6.3.1 Pin Configurations Summary

Table 31-3. Pin Configurations Summary

DIR	INEN	PULLEN	OUT	Configuration
0	0	0	X	Reset or analog I/O: all digital disabled
0	0	1	0	Pull-down; input disabled
0	0	1	1	Pull-up; input disabled
0	1	0	X	Input
0	1	1	0	Input with pull-down
0	1	1	1	Input with pull-up
1	0	X	X	Output; input disabled
1	1	X	X	Output; input enabled

31.6.3.2 Input Configuration

Figure 31-4. I/O configuration - Standard Input

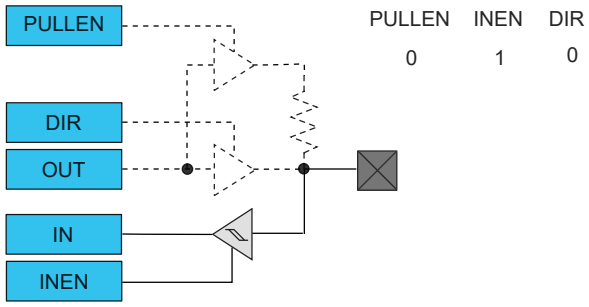
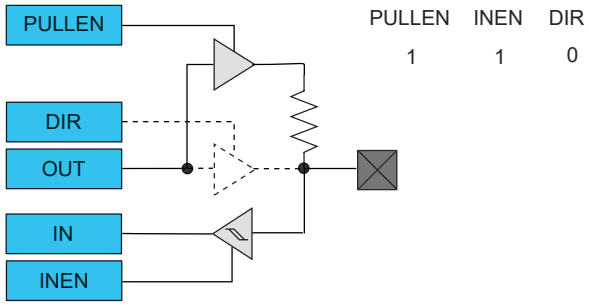


Figure 31-5. I/O Configuration - Input with Pull



Note: When pull is enabled, the pull value is defined by the OUT value.

31.6.3.3 Totem-Pole Output

When configured for totem-pole (push-pull) output, the pin is driven low or high according to the corresponding bit setting in the OUT register. In this configuration there is no current limitation for sink or source other than what the pin is capable of. If the pin is configured for input, the pin will float if no external pull is connected.

Note: Enabling the output driver will automatically disable pull.

Figure 31-6. I/O Configuration - Totem-Pole Output with Disabled Input

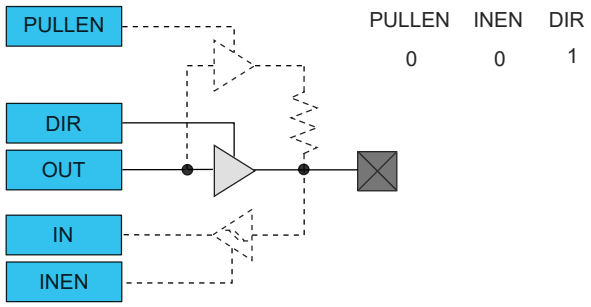


Figure 31-7. I/O Configuration - Totem-Pole Output with Enabled Input

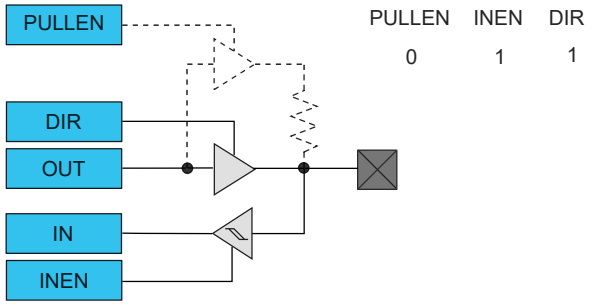
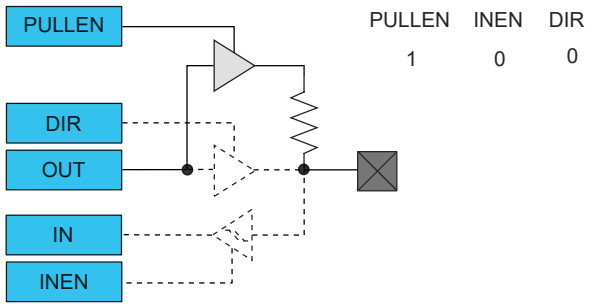


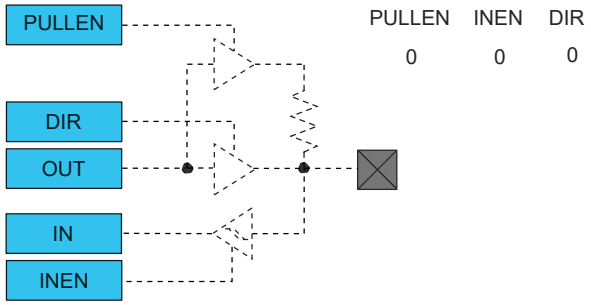
Figure 31-8. I/O Configuration - Output with Pull



31.6.3.4 Digital Functionality Disabled

Neither Input nor Output functionality are enabled.

Figure 31-9. I/O Configuration - Reset or Analog I/O: Digital Output, Input and Pull Disabled



31.6.4 PIC32CM LS00/LS60 Secure Access Rights

Non-secure write to CTRL, EVCTRL, or NONSEC registers is prohibited.

Non-secure read to CTRL or EVCTRL registers will return zero with no error resulting.

Non-secure write to a bit of DIR, DIRCLR, DIRSET, DIRTGL, OUT, OUTCLR, OUTSET, or OUTTGL registers is prohibited if the corresponding bit in NONSEC is zero.

Non-secure write to a PINCFGn register or to a PMUXn register field, either directly or through a write to the WRCONFIG register, is prohibited if the corresponding bit in NONSEC is zero.

DIR, DIRCLR, DIRSET, DIRTGL, IN, OUT, OUTCLR, OUTSET, or OUTTGL bits, PINCFGn registers, or PMUXn register fields relating to secure I/O pins (i.e. the corresponding bits in NONSEC are zero), read as zero in non-secure mode, with no error resulting.

INTFLAG.NSCHK is set to 1 when NSCHK and NONSEC register values are different. Writing a 1 to INTFLAG.NSCHK will clear it and clear the PORT interrupt if enabled.

Secure code should initially write a 1 for all non-secure pins into both NONSEC and NSCHK registers. Then, whenever secure code writes a different value into NONSEC, INTFLAG.NSCHK will be set to 1 and a PORT interrupt will occur, if enabled. The non-secure code can then compare the values of the NONSEC and NSCHK registers to determine which PORT pins have just changed to secure or to non-secure. It should then copy the current NONSEC register value into NSCHK.

31.6.5 Events

The PORT allows input events to control individual I/O pins. These input events (EVU0-3) are generated by the EVSYS module and can originate from a different clock domain than the PORT module.

The PORT can perform the following actions:

- Output (OUT): I/O pin will be set when the incoming event has a high level ('1') and cleared when the incoming event has a low-level ('0').
- Set (SET): I/O pin will be set when an incoming event is detected.
- Clear (CLR): I/O pin will be cleared when an incoming event is detected.
- Toggle (TGL): I/O pin will toggle when an incoming event is detected.

The event is output to pin without any internal latency. For SET, CLEAR and TOGGLE event actions, the action will be executed up to three clock cycles after a rising edge.

The event actions can be configured with the Event Action m bit group in the Event Input Control register(EVCTRL.EVACTn). Writing a '1' to a PORT Event Enable Input m of the Event Control register (EVCTRL.PORTEIn) enables the corresponding action on input event. Writing '0' to this bit disables the corresponding action on input event. Note that several actions can be enabled for incoming events. If several events are connected to the peripheral, any enabled action will be taken for any of the incoming events. Refer to the 32. Event System (EVSYS) for details on configuring the Event System.

Each event input can address one and only one I/O pin at a time. The selection of the pin is indicated by the PORT Event Pin Identifier of the Event Input Control register (EVCTR.PIDn). On the other hand, one I/O pin can be addressed by up to four different input events. To avoid action conflict on the output value of the register (OUT) of this particular I/O pin, only one action is performed according to the table below.

Note that this truth table can be applied to any SET/CLR/TGL configuration from two to four active input events.

Table 31-4. Priority on Simultaneous SET/CLR/TGL Event Actions

EVACT0	EVACT1	EVACT2	EVACT3	Executed Event Action
SET	SET	SET	SET	SET
CLR	CLR	CLR	CLR	CLR
All Other Combinations				TGL

Be careful when the event is output to pin. Due to the fact the events are received asynchronously, the I/O pin may have unpredictable levels, depending on the timing of when the events are received. When several events are output to the same pin, the lowest event line will get the access. All other events will be ignored.

31.6.6 PORT Access Priority

The PORT is accessed by different systems:

- The ARM® CPU through the ARM® single-cycle I/O port (IOBUS)
- The ARM® CPU through the high-speed matrix and the AHB/APB bridge (APB)
- EVSYS through four asynchronous input events

The CPU local bus (IOBUS) is an interface that connects the CPU directly to the PORT. It is a single-cycle bus interface, which does not support wait states. It supports 8-bit, 16-bit and 32-bit sizes.

This bus is generally used for low latency operation. The Data Direction (DIR) and Data Output Value (OUT) registers can be read, written, set, cleared or be toggled using this bus, and the Data Input Value (IN) registers can be read.

Since the IOBUS cannot wait for IN register resynchronization, the Control register (CTRL) must be configured to continuous sampling of all pins that need to be read via the IOBUS in order to prevent stale data from being read.

Note: Refer to the Product Mapping chapter for the IOBUS address.

The following priority is adopted:

1. ARM® CPU IOBUS (No wait tolerated)
2. APB
3. EVSYS input events

Note: One clock cycle latency can be observed on the APB access in case of concurrent PORT accesses.

For input events that require different actions on the same I/O pin, refer to 31.6.5. Events.

31.6.7 Debug Operation

When the CPU is halted in Debug mode, this peripheral will continue normal operation.

31.7 Register Summary

Refer to the Registers Description section for more details on register properties and access permissions.

Important: (PIC32CM LS00 only)
The peripheral register map is automatically duplicated in a Secure and Non-Secure alias:

- The Non-Secure alias is at the peripheral base address
- The Secure alias is located at the peripheral base address + 0x200

Refer to Mix-Secure Peripherals for more information on register access rights.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	DIR	31:24	DIR[31:24]							
		23:16	DIR[23:16]							
		15:8	DIR[15:8]							
		7:0	DIR[7:0]							
0x04	DIRCLR	31:24	DIRCLR[31:24]							
		23:16	DIRCLR[23:16]							
		15:8	DIRCLR[15:8]							
		7:0	DIRCLR[7:0]							
0x08	DIRSET	31:24	DIRSET[31:24]							
		23:16	DIRSET[23:16]							
		15:8	DIRSET[15:8]							
		7:0	DIRSET[7:0]							
0x0C	DIRTGL	31:24	DIRTGL[31:24]							
		23:16	DIRTGL[23:16]							
		15:8	DIRTGL[15:8]							
		7:0	DIRTGL[7:0]							
0x10	OUT	31:24	OUT[31:24]							
		23:16	OUT[23:16]							
		15:8	OUT[15:8]							
		7:0	OUT[7:0]							
0x14	OUTCLR	31:24	OUTCLR[31:24]							
		23:16	OUTCLR[23:16]							
		15:8	OUTCLR[15:8]							
		7:0	OUTCLR[7:0]							
0x18	OUTSET	31:24	OUTSET[31:24]							
		23:16	OUTSET[23:16]							
		15:8	OUTSET[15:8]							
		7:0	OUTSET[7:0]							
0x1C	OUTTGL	31:24	OUTTGL[31:24]							
		23:16	OUTTGL[23:16]							
		15:8	OUTTGL[15:8]							
		7:0	OUTTGL[7:0]							
0x20	IN	31:24	IN[31:24]							
		23:16	IN[23:16]							
		15:8	IN[15:8]							
		7:0	IN[7:0]							
0x24	CTRL	31:24	SAMPLING[31:24]							
		23:16	SAMPLING[23:16]							
		15:8	SAMPLING[15:8]							
		7:0	SAMPLING[7:0]							
0x28	WRCONFIG	31:24	HWSEL	WRPINCFG		WRPMUX	PMUX[3:0]			
		23:16		DRVSTR				PULLEN	INEN	PMUXEN
		15:8	PINMASK[15:8]							
		7:0	PINMASK[7:0]							

.....continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x2C	EVCTRL	31:24	PORTEI3	EVACT3[1:0]		PID3[4:0]				
		23:16	PORTEI2	EVACT2[1:0]		PID2[4:0]				
		15:8	PORTEI1	EVACT1[1:0]		PID1[4:0]				
		7:0	PORTEI0	EVACT0[1:0]		PID0[4:0]				
0x30	PMUX0	7:0	PMUXO[3:0]				PMUXE[3:0]			
...										
0x3F	PMUX15	7:0	PMUXO[3:0]				PMUXE[3:0]			
0x40	PINCFG0	7:0		DRVSTR				PULLEN	INEN	PMUXEN
...										
0x5F	PINCFG31	7:0		DRVSTR				PULLEN	INEN	PMUXEN
0x60	INTENCLR	31:24								
		23:16								
		15:8								
		7:0								NSCHK
0x64	INTENSET	31:24								
		23:16								
		15:8								
		7:0								NSCHK
0x68	INTFLAG	31:24								
		23:16								
		15:8								
		7:0								NSCHK
0x6C	NONSEC	31:24	NONSEC[31:24]							
		23:16	NONSEC[23:16]							
		15:8	NONSEC[15:8]							
		7:0	NONSEC[7:0]							
0x70	NSCHK	31:24	NSCHK[31:24]							
		23:16	NSCHK[23:16]							
		15:8	NSCHK[15:8]							
		7:0	NSCHK[7:0]							

31.7.1 Data Direction

Name: DIR
Offset: 0x00
Reset: 0x00000000
Property: PAC Write-Protection, Mix-Secure

Important: For **PIC32CM LS00/LS60 Non-Secure** accesses, read and write write accesses (RW*) are allowed only if the security attribution for the corresponding I/O pin is set as Non-Secured in the NONSEC register.

This register allows the user to configure one or more I/O pins as an input or output. This register can be manipulated without doing a read-modify-write operation by using the Data Direction Toggle (DIRTGL), Data Direction Clear (DIRCLR) and Data Direction Set (DIRSET) registers.

Tip: The I/O pins are assembled in pin groups ("PORT groups") with up to 32 pins. Group 0 consists of the PA pins, group 1 is for the PB pins, etc. Each pin group has its own PORT registers, with a 0x80 address spacing. For example, the register address offset for the Data Direction (DIR) register for group 0 (PA00 to PA31) is 0x00, and the register address offset for the DIR register for group 1 (PB00 to PB31) is 0x80.

Bit	31	30	29	28	27	26	25	24
	DIR[31:24]							
Access	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	DIR[23:16]							
Access	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/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
	DIR[15:8]							
Access	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/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
	DIR[7:0]							
Access	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – DIR[31:0] Port Data Direction
These bits set the data direction for the individual I/O pins in the PORT group.

Value	Description
0	The corresponding I/O pin in the PORT group is configured as an input.
1	The corresponding I/O pin in the PORT group is configured as an output.

31.7.2 Data Direction Clear

Name: DIRCLR
Offset: 0x04
Reset: 0x00000000
Property: PAC Write-Protection, Mix-Secure

Important: For **PIC32CM LS00/LS60 Non-Secure** accesses, read and write accesses (RW*) are allowed only if the security attribution for the corresponding I/O pin is set as Non-Secured in the NONSEC register.

This register allows the user to set one or more I/O pins as an input, without doing a read-modify-write operation. Changes in this register will also be reflected in the Data Direction (DIR), Data Direction Toggle (DIRTGL) and Data Direction Set (DIRSET) registers.

Tip: The I/O pins are assembled in pin groups ("PORT groups") with up to 32 pins. Group 0 consists of the PA pins, group 1 is for the PB pins, etc. Each pin group has its own PORT registers, with a 0x80 address spacing. For example, the register address offset for the Data Direction (DIR) register for group 0 (PA00 to PA31) is 0x00, and the register address offset for the DIR register for group 1 (PB00 to PB31) is 0x80

Bit	31	30	29	28	27	26	25	24
	DIRCLR[31:24]							
Access	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	DIRCLR[23:16]							
Access	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/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
	DIRCLR[15:8]							
Access	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/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
	DIRCLR[7:0]							
Access	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – DIRCLR[31:0] Port Data Direction Clear
Writing a '0' to a bit has no effect.
Writing a '1' to a bit will clear the corresponding bit in the DIR register, which configures the I/O pin as an input.

Value	Description
0	The corresponding I/O pin in the PORT group will keep its configuration.
1	The corresponding I/O pin in the PORT group is configured as input.

31.7.3 Data Direction Set

Name: DIRSET
Offset: 0x08
Reset: 0x00000000
Property: PAC Write-Protection, Mix-Secure

Important: For **PIC32CM LS00/LS60 Non-Secure** accesses, read and write accesses (RW*) are allowed only if the security attribution for the corresponding I/O pin is set as Non-Secured in the NONSEC register.

This register allows the user to set one or more I/O pins as an output, without doing a read-modify-write operation. Changes in this register will also be reflected in the Data Direction (DIR), Data Direction Toggle (DIRTGL) and Data Direction Clear (DIRCLR) registers.

Tip: The I/O pins are assembled in pin groups ("PORT groups") with up to 32 pins. Group 0 consists of the PA pins, group 1 is for the PB pins, etc. Each pin group has its own PORT registers, with a 0x80 address spacing. For example, the register address offset for the Data Direction (DIR) register for group 0 (PA00 to PA31) is 0x00, and the register address offset for the DIR register for group 1 (PB00 to PB31) is 0x80

Bit	31	30	29	28	27	26	25	24
	DIRSET[31:24]							
Access	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	DIRSET[23:16]							
Access	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/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
	DIRSET[15:8]							
Access	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/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
	DIRSET[7:0]							
Access	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – DIRSET[31:0] Port Data Direction Set
Writing '0' to a bit has no effect.
Writing '1' to a bit will set the corresponding bit in the DIR register, which configures the I/O pin as an output.

Value	Description
0	The corresponding I/O pin in the PORT group will keep its configuration.
1	The corresponding I/O pin in the PORT group is configured as an output.

31.7.4 Data Direction Toggle

Name: DIRTGL
Offset: 0x0C
Reset: 0x00000000
Property: PAC Write-Protection, Mix-Secure

Important: For **PIC32CM LS00/LS60 Non-Secure** accesses, read and write accesses (RW*) are allowed only if the security attribution for the corresponding I/O pin is set as Non-Secured in the NONSEC register.

This register allows the user to toggle the direction of one or more I/O pins, without doing a read-modify-write operation. Changes in this register will also be reflected in the Data Direction (DIR), Data Direction Set (DIRSET) and Data Direction Clear (DIRCLR) registers.

Tip: The I/O pins are assembled in pin groups (”PORT groups”) with up to 32 pins. Group 0 consists of the PA pins, group 1 is for the PB pins, etc. Each pin group has its own PORT registers, with a 0x80 address spacing. For example, the register address offset for the Data Direction (DIR) register for group 0 (PA00 to PA31) is 0x00, and the register address offset for the DIR register for group 1 (PB00 to PB31) is 0x80

Bit	31	30	29	28	27	26	25	24
	DIRTGL[31:24]							
Access	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	DIRTGL[23:16]							
Access	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/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
	DIRTGL[15:8]							
Access	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/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
	DIRTGL[7:0]							
Access	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – DIRTGL[31:0] Port Data Direction Toggle
Writing '0' to a bit has no effect.
Writing '1' to a bit will toggle the corresponding bit in the DIR register, which reverses the direction of the I/O pin.

Value	Description
0	The corresponding I/O pin in the PORT group will keep its configuration.
1	The direction of the corresponding I/O pin is toggled.

31.7.5 Data Output Value

Name: OUT
Offset: 0x10
Reset: 0x00000000
Property: PAC Write-Protection, Mix-Secure

Important: For **PIC32CM LS00/LS60 Non-Secure** accesses, read and write accesses (RW*) are allowed only if the security attribution for the corresponding I/O pin is set as Non-Secured in the NONSEC register.

This register sets the data output drive value for the individual I/O pins in the PORT.

This register can be manipulated without doing a read-modify-write operation by using the Data Output Value Clear (OUTCLR), Data Output Value Set (OUTSET), and Data Output Value Toggle (OUTTGL) registers.

Tip: The I/O pins are assembled in pin groups ("PORT groups") with up to 32 pins. Group 0 consists of the PA pins, group 1 is for the PB pins, etc. Each pin group has its own PORT registers, with a 0x80 address spacing. For example, the register address offset for the Data Direction (DIR) register for group 0 (PA00 to PA31) is 0x00, and the register address offset for the DIR register for group 1 (PB00 to PB31) is 0x80

Bit	31	30	29	28	27	26	25	24
	OUT[31:24]							
Access	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	OUT[23:16]							
Access	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/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
	OUT[15:8]							
Access	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/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
	OUT[7:0]							
Access	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – OUT[31:0] PORT Data Output Value
For pins configured as outputs via the Data Direction register (DIR), these bits set the logical output drive level.
For pins configured as inputs via the Data Direction register (DIR) and with pull enabled via the Pull Enable bit in the Pin Configuration register (PINCFG.PULLEN), these bits will set the input pull direction.

Value	Description
0	The I/O pin output is driven low, or the input is connected to an internal pull-down.
1	The I/O pin output is driven high, or the input is connected to an internal pull-up.

31.7.6 Data Output Value Clear

Name: OUTCLR
Offset: 0x14
Reset: 0x00000000
Property: PAC Write-Protection, Mix-Secure

Important: For **PIC32CM LS00/LS60 Non-Secure** accesses, read and write accesses (RW*) are allowed only if the security attribution for the corresponding I/O pin is set as Non-Secured in the NONSEC register.

This register allows the user to set one or more output I/O pin drive levels low, without doing a read-modify-write operation. Changes in this register will also be reflected in the Data Output Value (OUT), Data Output Value Toggle (OUTTGL) and Data Output Value Set (OUTSET) registers.

Tip: The I/O pins are assembled in pin groups ("PORT groups") with up to 32 pins. Group 0 consists of the PA pins, group 1 is for the PB pins, etc. Each pin group has its own PORT registers, with a 0x80 address spacing. For example, the register address offset for the Data Direction (DIR) register for group 0 (PA00 to PA31) is 0x00, and the register address offset for the DIR register for group 1 (PB00 to PB31) is 0x80

Bit	31	30	29	28	27	26	25	24
	OUTCLR[31:24]							
Access	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	OUTCLR[23:16]							
Access	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/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
	OUTCLR[15:8]							
Access	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/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
	OUTCLR[7:0]							
Access	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – OUTCLR[31:0] PORT Data Output Value Clear
Writing '0' to a bit has no effect.
Writing '1' to a bit will clear the corresponding bit in the OUT register. Pins configured as outputs via the Data Direction register (DIR) will be set to low output drive level. Pins configured as inputs via DIR and with pull enabled via the Pull Enable bit in the Pin Configuration register (PINCFG.PULLEN) will set the input pull direction to an internal pull-down.

Value	Description
0	The corresponding I/O pin in the PORT group will keep its configuration.
1	The corresponding I/O pin output is driven low, or the input is connected to an internal pull-down.

31.7.7 Data Output Value Set

Name: OUTSET
Offset: 0x18
Reset: 0x00000000
Property: PAC Write-Protection, Mix-Secure

Important: For **PIC32CM LS00/LS60 Non-Secure** accesses, read and write accesses (RW*) are allowed only if the security attribution for the corresponding I/O pin is set as Non-Secured in the NONSEC register.

This register allows the user to set one or more output I/O pin drive levels high, without doing a read-modify-write operation. Changes in this register will also be reflected in the Data Output Value (OUT), Data Output Value Toggle (OUTTGL) and Data Output Value Clear (OUTCLR) registers.

Tip: The I/O pins are assembled in pin groups (”PORT groups”) with up to 32 pins. Group 0 consists of the PA pins, group 1 is for the PB pins, etc. Each pin group has its own PORT registers, with a 0x80 address spacing. For example, the register address offset for the Data Direction (DIR) register for group 0 (PA00 to PA31) is 0x00, and the register address offset for the DIR register for group 1 (PB00 to PB31) is 0x80

Bit	31	30	29	28	27	26	25	24
	OUTSET[31:24]							
Access	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	OUTSET[23:16]							
Access	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/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
	OUTSET[15:8]							
Access	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/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
	OUTSET[7:0]							
Access	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – OUTSET[31:0] PORT Data Output Value Set
Writing '0' to a bit has no effect.
Writing '1' to a bit will set the corresponding bit in the OUT register, which sets the output drive level high for I/O pins configured as outputs via the Data Direction register (DIR). For pins configured as inputs via Data Direction register (DIR) with pull enabled via the Pull Enable register (PULLEN), these bits will set the input pull direction to an internal pull-up.

Value	Description
0	The corresponding I/O pin in the group will keep its configuration.
1	The corresponding I/O pin output is driven high, or the input is connected to an internal pull-up.

31.7.8 Data Output Value Toggle

Name: OUTTGL
Offset: 0x1C
Reset: 0x00000000
Property: PAC Write-Protection, Mix-Secure

Important: For **PIC32CM LS00/LS60 Non-Secure** accesses, read and write accesses (RW*) are allowed only if the security attribution for the corresponding I/O pin is set as Non-Secured in the NONSEC register.

This register allows the user to toggle the drive level of one or more output I/O pins, without doing a read-modify-write operation. Changes in this register will also be reflected in the Data Output Value (OUT), Data Output Value Set (OUTSET) and Data Output Value Clear (OUTCLR) registers.

Tip: The I/O pins are assembled in pin groups ("PORT groups") with up to 32 pins. Group 0 consists of the PA pins, group 1 is for the PB pins, etc. Each pin group has its own PORT registers, with a 0x80 address spacing. For example, the register address offset for the Data Direction (DIR) register for group 0 (PA00 to PA31) is 0x00, and the register address offset for the DIR register for group 1 (PB00 to PB31) is 0x80

Bit	31	30	29	28	27	26	25	24
	OUTTGL[31:24]							
Access	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	OUTTGL[23:16]							
Access	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/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
	OUTTGL[15:8]							
Access	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/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
	OUTTGL[7:0]							
Access	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – OUTTGL[31:0] PORT Data Output Value Toggle
Writing '0' to a bit has no effect.
Writing '1' to a bit will toggle the corresponding bit in the OUT register, which inverts the output drive level for I/O pins configured as outputs via the Data Direction register (DIR). For pins configured as inputs via Data Direction register (DIR) with pull enabled via the Pull Enable register (PULLEN), these bits will toggle the input pull direction.

Value	Description
0	The corresponding I/O pin in the PORT group will keep its configuration.
1	The corresponding OUT bit value is toggled.

31.7.9 Data Input Value

Name: IN
Offset: 0x20
Reset: 0x40000000
Property: Mix-Secure

Important: For **PIC32CM LS00/LS60 Non-Secure** accesses, read accesses (R*) are allowed only if the security attribution for the corresponding I/O pin is set as Non-Secured in the NONSEC register.

Tip: The I/O pins are assembled in pin groups ("PORT groups") with up to 32 pins. Group 0 consists of the PA pins, group 1 is for the PB pins, etc. Each pin group has its own PORT registers, with a 0x80 address spacing. For example, the register address offset for the Data Direction (DIR) register for group 0 (PA00 to PA31) is 0x00, and the register address offset for the DIR register for group 1 (PB00 to PB31) is 0x80

Bit	31	30	29	28	27	26	25	24
	IN[31:24]							
Access	R/R*/R	R/R*/R	R/R*/R	R/R*/R	R/R*/R	R/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
	IN[23:16]							
Access	R/R*/R	R/R*/R	R/R*/R	R/R*/R	R/R*/R	R/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
	IN[15:8]							
Access	R/R*/R	R/R*/R	R/R*/R	R/R*/R	R/R*/R	R/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
	IN[7:0]							
Access	R/R*/R	R/R*/R	R/R*/R	R/R*/R	R/R*/R	R/R*/R	R/R*/R	R/R*/R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – IN[31:0] PORT Data Input Value
These bits are cleared when the corresponding I/O pin input sampler detects a logical low level on the input pin.
These bits are set when the corresponding I/O pin input sampler detects a logical high level on the input pin.

31.7.10 Control

Name: CTRL
Offset: 0x24
Reset: 0x00000000
Property: PAC Write-Protection, Secure

Tip: The I/O pins are assembled in pin groups (”PORT groups”) with up to 32 pins. Group 0 consists of the PA pins, group 1 is for the PB pins, etc. Each pin group has its own PORT registers, with a 0x80 address spacing. For example, the register address offset for the Data Direction (DIR) register for group 0 (PA00 to PA31) is 0x00, and the register address offset for the DIR register for group 1 (PB00 to PB31) is 0x80

Bit	31	30	29	28	27	26	25	24
	SAMPLING[31:24]							
Access	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	SAMPLING[23:16]							
Access	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW	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
	SAMPLING[15:8]							
Access	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW	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
	SAMPLING[7:0]							
Access	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – SAMPLING[31:0] Input Sampling Mode
Configures the input sampling functionality of the I/O pin input samplers, for pins configured as inputs via the Data Direction register (DIR).
The input samplers are enabled and disabled in sub-groups of eight. Thus if any pins within a byte request continuous sampling, all pins in that eight pin sub-group will be continuously sampled.

Value	Description
0	The I/O pin input synchronizer is disabled.
1	The I/O pin input synchronizer is enabled.

31.7.11 Write Configuration

Name: WRCONFIG
Offset: 0x28
Reset: 0x00000000
Property: PAC Write-Protection, Mix-Secure

Important: For **PIC32CM LS00/LS60 Non-Secure** accesses, write accesses (W*) are allowed only if the security attribution for the corresponding I/O pin is set as Non-Secured in the NONSEC register.

Tip: The I/O pins are assembled in pin groups ("PORT groups") with up to 32 pins. Group 0 consists of the PA pins, group 1 is for the PB pins, etc. Each pin group has its own PORT registers, with a 0x80 address spacing. For example, the register address offset for the Data Direction (DIR) register for group 0 (PA00 to PA31) is 0x00, and the register address offset for the DIR register for group 1 (PB00 to PB31) is 0x80

This write-only register is used to configure several pins simultaneously with the same configuration and/or peripheral multiplexing.

In order to avoid side effect of non-atomic access, 8-bit or 16-bit writes to this register will have no effect. Reading this register always returns zero.

Bit	31	30	29	28	27	26	25	24
	HWSEL	WRPINCFG		WRPMUX	PMUX[3:0]			
Access	W/W*/W	W/W*/W		W/W*/W	W/W*/W	W/W*/W	W/W*/W	W/W*/W
Reset	0	0		0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
		DRVSTR				PULLEN	INEN	PMUXEN
Access		W/W*/W				W/W*/W	W/W*/W	W/W*/W
Reset		0				0	0	0

Bit	15	14	13	12	11	10	9	8
	PINMASK[15:8]							
Access	W/W*/W	W/W*/W	W/W*/W	W/W*/W	W/W*/W	W/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
	PINMASK[7:0]							
Access	W/W*/W	W/W*/W	W/W*/W	W/W*/W	W/W*/W	W/W*/W	W/W*/W	W/W*/W
Reset	0	0	0	0	0	0	0	0

Bit 31 – HWSEL Half-Word Select
This bit selects the half-word field of a 32-PORT group to be reconfigured in the atomic write operation.
This bit will always read as zero.

Value	Description
0	The lower 16 pins of the PORT group will be configured.
1	The upper 16 pins of the PORT group will be configured.

Bit 30 – WRPINCFG Write Pin Configuration Register (PINCFGy)
This bit determines whether the atomic write operation will update the Pin Configuration register (PINCFGy) or not for all pins selected by the WRCONFIG.PINMASK and WRCONFIG.HWSEL bits.
Writing '0' to this bit has no effect.

Writing '1' to this bit updates the configuration of the selected pins with the written WRCONFIG.DRVSTR, WRCONFIG.PULLEN, WRCONFIG.INEN, WRCONFIG.PMUXEN, and WRCONFIG.PINMASK values.
This bit will always read as zero.

Value	Description
0	The PINCFGy registers of the selected pins will not be updated.
1	The PINCFGy registers of the selected pins will be updated.

Bit 28 – WRPMUX Write Peripheral Multiplexing Register (PMUXn)

This bit determines whether the atomic write operation will update the Peripheral Multiplexing register (PMUXn) or not for all pins selected by the WRCONFIG.PINMASK and WRCONFIG.HWSEL bits.
Writing '0' to this bit has no effect.
Writing '1' to this bit updates the pin multiplexer configuration of the selected pins with the written WRCONFIG. PMUX value.
This bit will always read as zero.

Value	Description
0	The PMUXn registers of the selected pins will not be updated.
1	The PMUXn registers of the selected pins will be updated.

Bits 27:24 – PMUX[3:0] Peripheral Multiplexing

These bits determine the new value written to the Peripheral Multiplexing register (PMUXn) for all pins selected by the WRCONFIG.PINMASK and WRCONFIG.HWSEL bits, when the WRCONFIG.WRPMUX bit is set.
These bits will always read as zero.

Bit 22 – DRVSTR Output Driver Strength Selection

This bit determines the new value written to PINCFGy.DRVSTR for all pins selected by the WRCONFIG.PINMASK and WRCONFIG.HWSEL bits, when the WRCONFIG.WRPINCFG bit is set.
This bit will always read as zero.

Bit 18 – PULLEN Pull Enable

This bit determines the new value written to PINCFGy.PULLEN for all pins selected by the WRCONFIG.PINMASK and WRCONFIG.HWSEL bits, when the WRCONFIG.WRPINCFG bit is set.
This bit will always read as zero.

Bit 17 – INEN Input Enable

This bit determines the new value written to PINCFGy.INEN for all pins selected by the WRCONFIG.PINMASK and WRCONFIG.HWSEL bits, when the WRCONFIG.WRPINCFG bit is set.
This bit will always read as zero.

Bit 16 – PMUXEN Peripheral Multiplexer Enable

This bit determines the new value written to PINCFGy.PMUXEN for all pins selected by the WRCONFIG.PINMASK and WRCONFIG.HWSEL bits, when the WRCONFIG.WRPINCFG bit is set.
This bit will always read as zero.

Bits 15:0 – PINMASK[15:0] Pin Mask for Multiple Pin Configuration

These bits select the pins to be configured within the half-word group selected by the WRCONFIG.HWSEL bit.
These bits will always read as zero.

Value	Description
0	The configuration of the corresponding I/O pin in the half-word group will be left unchanged.
1	The configuration of the corresponding I/O pin in the half-word PORT group will be updated.

31.7.12 Event Input Control

Name: EVCTRL
Offset: 0x2C
Reset: 0x00000000
Property: PAC Write-Protection, Secure

Tip: The I/O pins are assembled in pin groups (”PORT groups”) with up to 32 pins. Group 0 consists of the PA pins, group 1 is for the PB pins, etc. Each pin group has its own PORT registers, with a 0x80 address spacing. For example, the register address offset for the Data Direction (DIR) register for group 0 (PA00 to PA31) is 0x00, and the register address offset for the DIR register for group 1 (PB00 to PB31) is 0x80.

For each PORT group, there are up to four input event pins: EVU0-3.

Each byte of this register addresses one Event input pin.

Bit	31	30	29	28	27	26	25	24
	PORTEI3	EVACT3[1:0]		PID3[4:0]				
Access	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	PORTEI2	EVACT2[1:0]		PID2[4:0]				
Access	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW	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
	PORTEI1	EVACT1[1:0]		PID1[4:0]				
Access	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW	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
	PORTEI0	EVACT0[1:0]		PID0[4:0]				
Access	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW	RW/-/RW
Reset	0	0	0	0	0	0	0	0

Bits 7, 15, 23, 31 – PORTEIx PORT Event Input Enable x [x = 3..0]

Value	Description
0	The event action x (EVACTx) will not be triggered on any incoming event.
1	The event action x (EVACTx) will be triggered on any incoming event.

Bits 5:6, 13:14, 21:22, 29:30 – EVACTx PORT Event Action x [x = 3..0]
These bits define the event action the PORT will perform on event input x. See also Table 31-5.

Bits 0:4, 8:12, 16:20, 24:28 – PIDx PORT Event Pin Identifier x [x = 3..0]
These bits define the I/O pin on which the event action will be performed, according to Table 31-6.

Table 31-5. PORT Event x Action (x = [3..0])

Value	Name	Description
0x0	OUT	Output register of pin will be set to level of event.
0x1	SET	Set output register of pin on event.
0x2	CLR	Clear output register of pin on event.

.....continued		
Value	**Name**	**Description**
0x3	TGL	Toggle output register of pin on event.

Table 31-6. PORT Event x Pin Identifier (x = [3..0])

Value	Name	Description
0x0	PIN0	Event action to be executed on PIN 0.
0x1	PIN1	Event action to be executed on PIN 1.
...	...	...
0x31	PIN31	Event action to be executed on PIN 31.

31.7.13 Peripheral Multiplexing n

Name: PMUX
Offset: 0x30 + n*0x01 [n=0..15]
Reset: 0x06 (PMUX15 for Group 0 only), 0x00 (others)
Property: PAC Write-Protection, Mix-Secure

Important: For **PIC32CM LS00/LS60 Non-Secure** accesses, read and write accesses (RW*) are allowed only if the security attribution for the corresponding I/O pin is set as Non-Secured in the NONSEC register.

Tip: The I/O pins are assembled in pin groups ("PORT groups") with up to 32 pins. Group 0 consists of the PA pins, group 1 is for the PB pins, etc. Each pin group has its own PORT registers, with a 0x80 address spacing. For example, the register address offset for the Data Direction (DIR) register for group 0 (PA00 to PA31) is 0x00, and the register address offset for the DIR register for group 1 (PB00 to PB31) is 0x80.

There are up to 16 Peripheral Multiplexing registers in each group, one for every set of two subsequent I/O lines. The n denotes the number of the set of I/O lines.

Bit	7	6	5	4	3	2	1	0
	PMUXO[3:0]				PMUXE[3:0]			
Access	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW
Reset	0	0	0	0	0	0	0	0

Bits 7:4 – PMUXO[3:0] Peripheral Multiplexing for Odd-Numbered Pin
These bits select the peripheral function for odd-numbered pins (2*n + 1) of a PORT group, if the corresponding PINCFGy.PMUXEN bit is '1'.
Not all possible values for this selection may be valid. For more details, refer to the *Pinout and Packaging*.

Value	Name	Description
0x0	A	Peripheral function A selected
0x1	B	Peripheral function B selected
0x2	C	Peripheral function C selected
0x3	D	Peripheral function D selected
0x4	E	Peripheral function E selected
0x5	-	Reserved
0x6	G	Peripheral function G selected
0x7	H	Peripheral function H selected
0x8	I	Peripheral function I selected
0x9	J	Peripheral function J selected
0xA	K	Peripheral function K selected
0xB-0xF	-	Reserved

Bits 3:0 – PMUXE[3:0] Peripheral Multiplexing for Even-Numbered Pin
These bits select the peripheral function for even-numbered pins (2*n) of a PORT group, if the corresponding PINCFGy.PMUXEN bit is '1'.
Not all possible values for this selection may be valid. For more details, refer to the *Pinout and Packaging*

Value	Name	Description
0x0	A	Peripheral function A selected
0x1	B	Peripheral function B selected
0x2	C	Peripheral function C selected

.....continued

Value	Name	Description
0x3	D	Peripheral function D selected
0x4	E	Peripheral function E selected
0x5	-	Reserved
0x6	G	Peripheral function G selected
0x7	H	Peripheral function H selected
0x8	I	Peripheral function I selected
0x9	J	Peripheral function J selected
0xA	K	Peripheral function K selected
0xB-0xF	-	Reserved

31.7.14 Pin Configuration n

Name: PINCFG
Offset: 0x40 + n*0x01 [n=0..31]
Reset: 0x00
Property: PAC Write-Protection, Mix-Secure

Important: For **PIC32CM LS00/LS60 Non-Secure** accesses, read and write accesses (RW*) are allowed only if the security attribution for the corresponding I/O pin is set as Non-Secured in the NONSEC register.

Tip: The I/O pins are assembled in pin groups ("PORT groups") with up to 32 pins. Group 0 consists of the PA pins, group 1 is for the PB pins, etc. Each pin group has its own PORT registers, with a 0x80 address spacing. For example, the register address offset for the Data Direction (DIR) register for group 0 (PA00 to PA31) is 0x00, and the register address offset for the DIR register for group 1 (PB00 to PB31) is 0x80.

There are up to 32 Pin Configuration registers in each PORT group, one for each I/O line.

Bit	7	6	5	4	3	2	1	0
		DRVSTR				PULLEN	INEN	PMUXEN
Access		RW/RW*/RW				RW/RW*/RW	RW/RW*/RW	RW/RW*/RW
Reset		0				0	0	0

Bit 6 – DRVSTR Output Driver Strength Selection
This bit controls the output driver strength of an I/O pin configured as an output.

Value	Description
0	Pin drive strength is set to normal drive strength.
1	Pin drive strength is set to stronger drive strength.

Bit 2 – PULLEN Pull Enable
This bit enables the internal pull-up or pull-down resistor of an I/O pin configured as an input.

Value	Description
0	Internal pull resistor is disabled, and the input is in a high-impedance configuration.
1	Internal pull resistor is enabled, and the input is driven to a defined logic level in the absence of external input.

Bit 1 – INEN Input Enable
This bit controls the input buffer of an I/O pin configured as either an input or output.
Writing a zero to this bit disables the input buffer completely, preventing read-back of the physical pin state when the pin is configured as either an input or output.

Value	Description
0	Input buffer for the I/O pin is disabled, and the input value will not be sampled.
1	Input buffer for the I/O pin is enabled, and the input value will be sampled when required.

Bit 0 – PMUXEN Peripheral Multiplexer Enable
This bit enables or disables the peripheral multiplexer selection set in the Peripheral Multiplexing register (PMUXn) to enable or disable alternative peripheral control over an I/O pin direction and output drive value.
Writing a zero to this bit allows the PORT to control the pad direction via the Data Direction register (DIR) and output drive value via the Data Output Value register (OUT). The peripheral multiplexer value in PMUXn is ignored. Writing '1' to this bit enables the peripheral selection in PMUXn to control the pad. In this configuration, the physical pin state may still be read from the Data Input Value register (IN) if PINCFGn.INEN is set.

Value	Description
0	The peripheral multiplexer selection is disabled, and the PORT registers control the direction and output drive value.
1	The peripheral multiplexer selection is enabled, and the selected peripheral function controls the direction and output drive value.

31.7.15 Interrupt Enable Clear

Name: INTENCLR
Offset: 0x60
Reset: 0x00000000
Property: PAC Write-Protection

Important: This register is only available for **PIC32CM LS00/LS60** and has no effect for **PIC32CM LE00**.

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).

Tip: The I/O pins are assembled in pin groups (”PORT groups”) with up to 32 pins. Group 0 consists of the PA pins, group 1 is for the PB pins, etc. Each pin group has its own PORT registers, with a 0x80 address spacing. For example, the register address offset for the Data Direction (DIR) register for group 0 (PA00 to PA31) is 0x00, and the register address offset for the DIR register for group 1 (PB00 to PB31) is 0x80.

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
								NSCHK
Access								RW
Reset								0

Bit 0 – NSCHK Non-Secure Check Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the Non-Secure Check Interrupt Enable bit, which disables the Non-Secure Check interrupt.

Value	Description
0	The Non-Secure Check interrupt is disabled.
1	The Non-Secure Check interrupt is enabled.

31.7.16 Interrupt Enable Set

Name: INTENSET
Offset: 0x64
Reset: 0x00000000
Property: PAC Write-Protection

Important: This register is only available for **PIC32CM LS00/LS60** and has no effect for **PIC32CM LE00**.

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).

Tip: The I/O pins are assembled in pin groups (”PORT groups”) with up to 32 pins. Group 0 consists of the PA pins, group 1 is for the PB pins, etc. Each pin group has its own PORT registers, with a 0x80 address spacing. For example, the register address offset for the Data Direction (DIR) register for group 0 (PA00 to PA31) is 0x00, and the register address offset for the DIR register for group 1 (PB00 to PB31) is 0x80.

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
								NSCHK
Access								RW
Reset								0

Bit 0 – NSCHK Non-Secure Check Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will set the Non-Secure Check Interrupt Enable bit, which enables the Non-Secure Check interrupt.

Value	Description
0	The Non-Secure Check interrupt is disabled.
1	The Non-Secure Check interrupt is enabled.

31.7.17 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x68
Reset: 0x00000000
Property: -

Important: This register is only available for **PIC32CM LS00/LS60** and has no effect for **PIC32CM LE00**.

Tip: The I/O pins are assembled in pin groups ("PORT groups") with up to 32 pins. Group 0 consists of the PA pins, group 1 is for the PB pins, etc. Each pin group has its own PORT registers, with a 0x80 address spacing. For example, the register address offset for the Data Direction (DIR) register for group 0 (PA00 to PA31) is 0x00, and the register address offset for the DIR register for group 1 (PB00 to PB31) is 0x80.

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
								NSCHK
Access								RW/RW/RW
Reset								0

Bit 0 – NSCHK Non-Secure Check
This flag is set on NONSEC write when a bit in NSCHK is 1 and the corresponding bit in NONSEC is cleared, or when a bit in NSCHK is 0 and the corresponding bit in NONSEC is set.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Non-Secure Check interrupt flag.

31.7.18 Security Attribution

Name: NONSEC
Offset: 0x6C
Reset: 0x00000000
Property: PAC Write-Protection, Write-Secure

Important: This register is only available for **PIC32CM LS00/LS60** and has no effect for **PIC32CM LE00**.

This register allows the user to configure one or more I/O pins as secured or non-secured.

Tip: The I/O pins are assembled in pin groups (”PORT groups”) with up to 32 pins. Group 0 consists of the PA pins, group 1 is for the PB pins, etc. Each pin group has its own PORT registers, with a 0x80 address spacing. For example, the register address offset for the Data Direction (DIR) register for group 0 (PA00 to PA31) is 0x00, and the register address offset for the DIR register for group 1 (PB00 to PB31) is 0x80.

Bit	31	30	29	28	27	26	25	24
	NONSEC[31:24]							
Access	RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	NONSEC[23:16]							
Access	RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	NONSEC[15:8]							
Access	RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW
Reset	0	0	0	0	0	0	0	0

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

Bits 31:0 – NONSEC[31:0] Port Security Attribution
These bits set the security attribution for the individual I/O pins in the PORT group.

Value	Description
0	The corresponding I/O pin in the PORT group is configured as secured. When module is PAC secured, the configuration for this pin is only available through the secure alias. Attempt to change the pin configuration through the non-secure alias will be silently ignored and reads will return 0.
1	The corresponding I/O pin in the PORT group is configured as non-secured. The I/O line configuration for this pin is available through the non-secure alias.

31.7.19 Security Attribution Check

Name: NSCHK
Offset: 0x70
Reset: 0x00000000
Property: PAC Write-Protection

Important: This register is only available for **PIC32CM LS00/LS60** and has no effect for **PIC32CM LE00**.

This register allows the user to select one or more pins to check their security attribution as non-secured.

Tip: The I/O pins are assembled in pin groups (”PORT groups”) with up to 32 pins. Group 0 consists of the PA pins, group 1 is for the PB pins, etc. Each pin group has its own PORT registers, with a 0x80 address spacing. For example, the register address offset for the Data Direction (DIR) register for group 0 (PA00 to PA31) is 0x00, and the register address offset for the DIR register for group 1 (PB00 to PB31) is 0x80.

Bit	31	30	29	28	27	26	25	24
	NSCHK[31:24]							
Access	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/RW/RW
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	NSCHK[23:16]							
Access	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/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
	NSCHK[15:8]							
Access	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/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
	NSCHK[7:0]							
Access	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/RW/RW
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – NSCHK[31:0] Port Security Attribution Check
These bits select the individual pins for security attribution check. If any pin selected in NSCHK has the corresponding bit in NONSEC set to the opposite value, then the NSCHK interrupt flag will be set.

Value	Description
0	0-to-1 transition will be detected on corresponding NONSEC bit.
1	1-to-0 transition will be detected on corresponding NONSEC bit.

32. Event System (EVSYS)

32.1 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. Channels and event users can be defined as secured or non-secured, where secured channels or event users can only be handled by secure code.

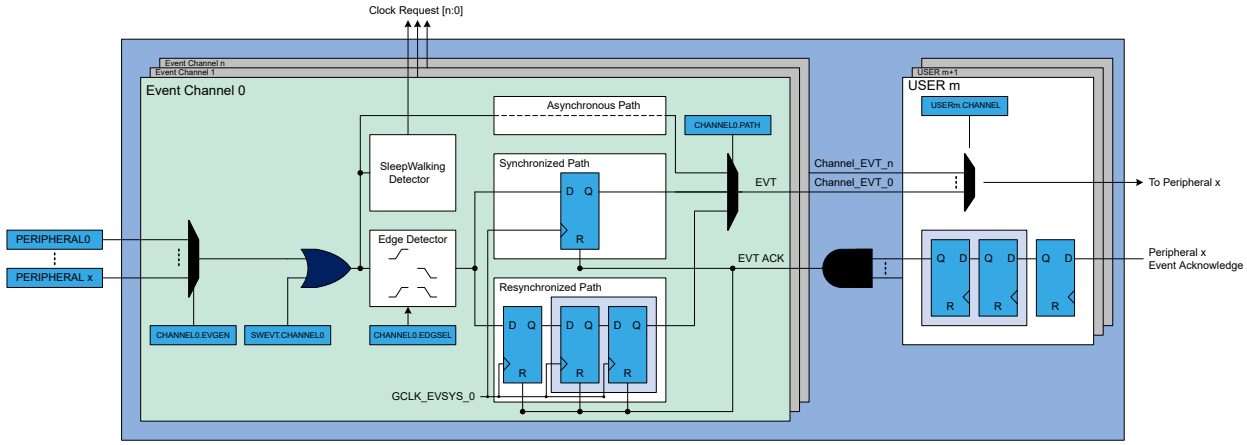
Communication is made without CPU intervention and without consuming system resources such as bus or RAM bandwidth. This reduces the load on the CPU and other system resources, compared to a traditional interrupt-based system.

32.2 Features

- 12 configurable event channels:
  - All channels can be connected to any event generator
  - All channels provide a pure asynchronous path
  - 8 channels (CHANNEL0 to CHANNEL7) provide a resynchronized or synchronous path using their dedicated generic clock (GCLK_EVSYS_CHANNEL_n)
- 91 event generators.
- 52 event users.
- Configurable edge detector.
- Peripherals can be event generators, event users, or both.
- SleepWalking 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.
- Each channel and each event user can be configured as secured or non-secured (**PIC32CM LS00/LS60**).

32.3 Block Diagram

Figure 32-1. Event System Block Diagram



32.4 Peripheral Dependencies

Table 32-1. EVSYS Configuration Summary

Peripheral name	Base address	NVIC IRQ Index	MCLK AHB/APB Clocks	GCLK Peripheral Channel Index (GCLK.PCHCTRL)	PAC Peripheral Identifier Index (PAC.WRCTRL)	Power Domain (PM.STDBYCFG)
EVSYS	0x42000000	21: EVD0, OVR0 22: EVD1, OVR1 23: EVD2, OVR2 24: EVD3, OVR3 25: EVD4, OVR4 26: EVD5, OVR5 27: EVD6, OVR6 28: EVD7, OVR7 29: NSCHK	CLK_EVSYS_APB	8: GCLK_EVSYS_CHANNEL_0 9: GCLK_EVSYS_CHANNEL_1 10: GCLK_EVSYS_CHANNEL_2 11: GCLK_EVSYS_CHANNEL_3 12: GCLK_EVSYS_CHANNEL_4 13: GCLK_EVSYS_CHANNEL_5 14: GCLK_EVSYS_CHANNEL_6 15: GCLK_EVSYS_CHANNEL_7	64	PDSW

32.5 Functional Description

32.5.1 Principle of Operation

The Event System consists of several channels which route the internal events from peripherals (generators) to other internal peripherals or I/O pins (users). 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.

Each channel which can be configured as synchronous or resynchronized has a dedicated generic clock (GCLK_EVSYS_CHANNEL_n). These are used for event detection and propagation for each channel. These clocks must be configured and enabled in the generic clock controller before using the EVSYS. Refer to GCLK - Generic Clock Controller for details.

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.

Important: Only EVSYS channel 0 to 7 can be configured as synchronous or resynchronized.

For further details, refer to the Channel Path section of this chapter.

32.5.2 Basic Operation

32.5.2.1 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 event generator peripheral, enable output of 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, RTC.EVCTRL.OVFEO.
2. Configure the EVSYS:
   a. Configure the Event User multiplexer by writing the respective EVSYS.USERm register, refer to 32.5.2.3. User Multiplexer Setup.

b. Configure the Event Channel by writing the respective EVSYS.CHANNELn register, refer to 32.5.2.4. Event System Channel.

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 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, ADC.EVCTRL.STARTEI.

32.5.2.2 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 will be reset to their initial state and all ongoing events will be canceled.

Refer to CTRLA.SWRST register for details.

32.5.2.3 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 user multiplexers is found in the User (USERm) register description.

32.5.2.4 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.

32.5.2.5 Event Generators

Each event channel can receive the events form 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 (ie, CHANNELn.EVGEN = 0)

32.5.2.6 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_CHANNEL_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 should 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.

32.5.2.7 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).

32.5.2.8 Event Latency

An event from an event generator is propagated to an event user with different latency, depending on event channel 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_CHANNEL_n clock cycle.
- Resynchronized Path: The maximum routing latency of an external event is three GCLK_EVSYS_CHANNEL_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.

32.5.2.9 The Overrun Channel n Interrupt

The Overrun Channel n interrupt flag in the Interrupt Flag Status and Clear register (CHINTFLAGn.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 CHINTFLAGn.OVR is always read as zero.

32.5.2.10 The Event Detected Channel n Interrupt

The Event Detected Channel n interrupt flag in the Interrupt Flag Status and Clear register (CHINTFLAGn.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 asynchronous path, the CHINTFLAGn.EVD is always zero.

32.5.2.11 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.

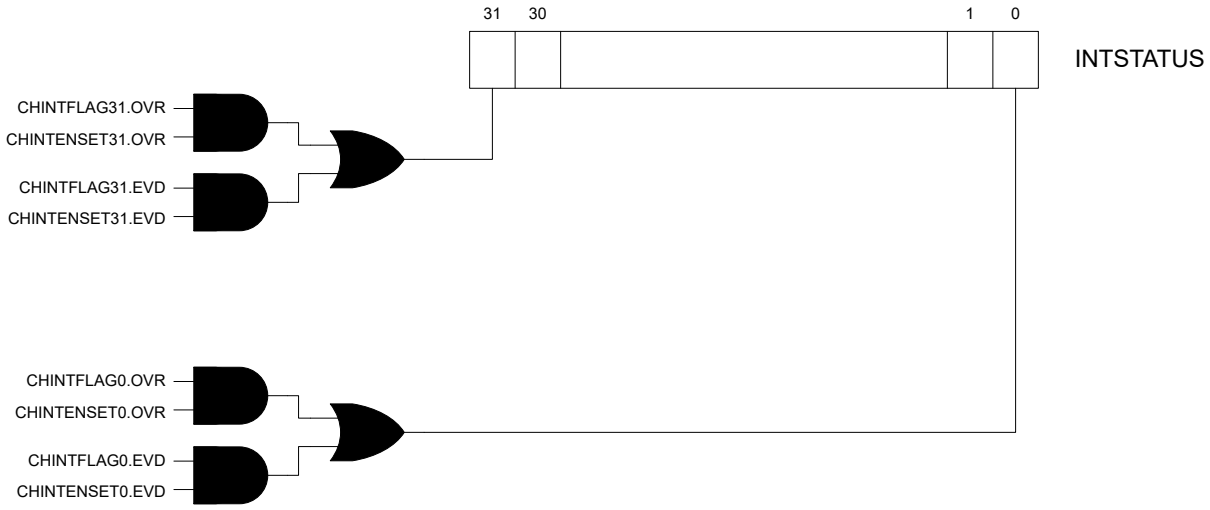
32.5.2.12 Software Event

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; i.e., when the bit is set to ‘1’, an event will be generated on the respective channel.

32.5.2.13 Interrupt Status and Interrupts Arbitration

The Interrupt Status register stores all channels with pending interrupts, as shown below.

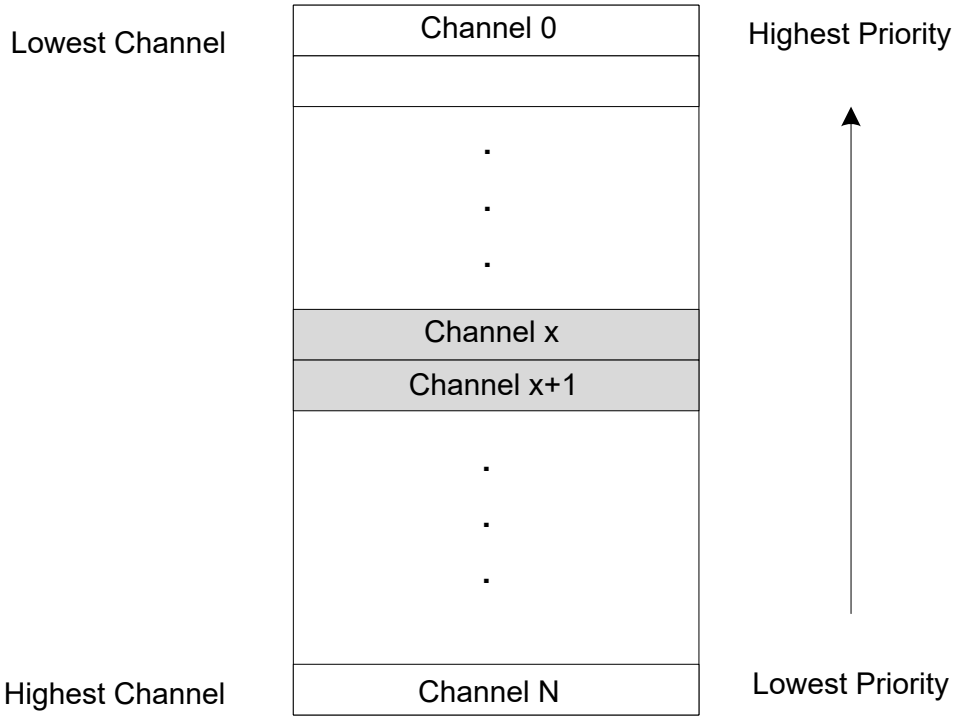
Figure 32-2. Interrupt Status Register



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 has been cleared. The Channel Pending Interrupt register (INTPEND) will provide the channel number with the highest interrupt priority, and the corresponding channel interrupt flags and status bits.

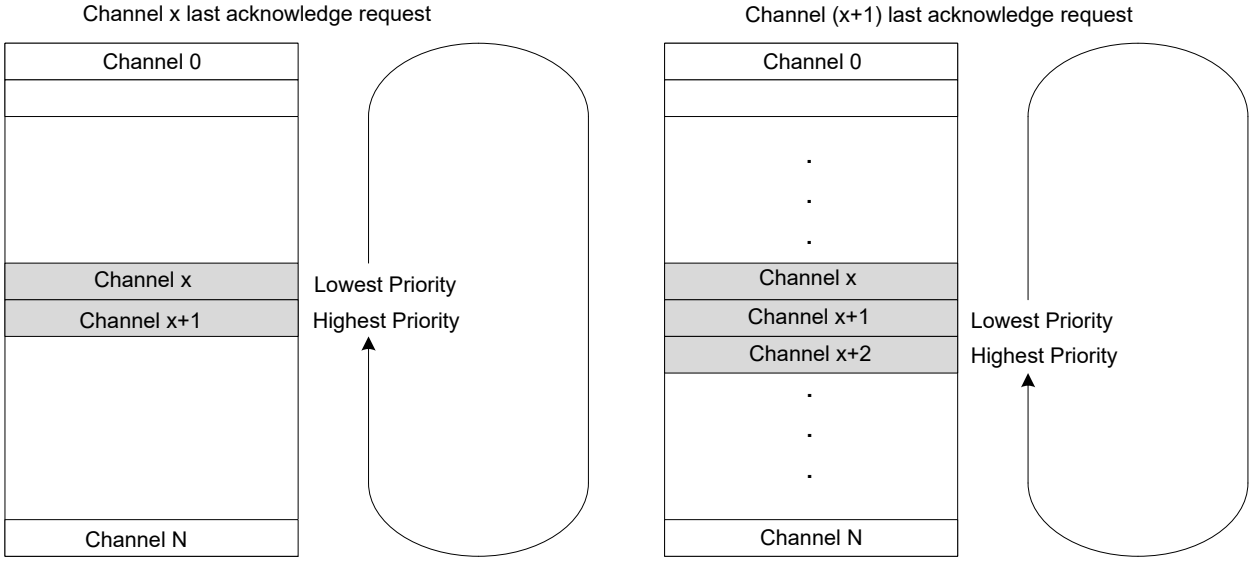
By default, static arbitration is enabled (PRICTRL.RRENx is '0'), the arbiter will prioritize a low channel number over a high channel number as shown below. When using the status scheme, there is a risk of high channel numbers never being granted access by the arbiter. This can be avoided using a dynamic arbitration scheme.

Figure 32-3. 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 shown below. 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 32-4. 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, will clear the corresponding interrupt flag of the channel specified by the INTPEND.ID bits.

32.5.3 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 pending interrupt and the respective interrupt flags.

32.5.4 Sleep Mode Operation

The Event System can generate interrupts to wake up the device from 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 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 will increase by two GCLK_EVSYS_CHANNEL_n clock (i.e., up to five GCLK_EVSYS_CHANNEL_n clock cycles).

A channel will behave differently in different sleep modes regarding to CHANNELn.RUNSTDBY and CHANNELn.ONDEMAND:

Table 32-2. Event Channel Sleep Behavior (1)

CHANNELn.PATH	CHANNELn. ONDEMAND	CHANNELn. RUNSTDBY	Sleep Behavior
ASYNC	0	0	Only run in IDLE sleep modes if an event must be propagated. Disabled in STANDBY sleep mode.
SYNC/RESYNC	0	0	Not Applicable. Works only in ACTIVE mode.
SYNC/RESYNC	0	1	Run in both IDLE and STANDBY sleep modes.
SYNC/RESYNC	1	0	Only run in IDLE sleep 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.

Note:

1. ONDEMAND=1 or RUNSTDBY=1 are not supported for channels when the asynchronous path is selected.

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_CHANNEL_n) will be restarted if that channel uses a synchronized path or a resynchronized path. It does not need to wake the system from sleep.

Important: This generic clock only applies to channels which can be configured as synchronous or resynchronized.

32.5.5 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.

32.6 Register Summary

Refer to the Registers Description section for more details on register properties and access permissions.

Important: (PIC32CM LS00 only)
The peripheral register map is automatically duplicated in a Secure and Non-Secure alias:

- The Non-Secure alias is at the peripheral base address
- The Secure alias is located at the peripheral base address + 0x200

Refer to Mix-Secure Peripherals for more information on register access rights.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRLA	7:0								SWRST
0x01 ... 0x03	Reserved									
0x04	SWEVT	31:24								
		23:16								
		15:8					CHANNEL11	CHANNEL10	CHANNEL9	CHANNEL8
		7:0	CHANNEL7	CHANNEL6	CHANNEL5	CHANNEL4	CHANNEL3	CHANNEL2	CHANNEL1	CHANNEL0
0x08	PRICTRL	7:0	RREN					PRI[2:0]		
0x09 ... 0x0F	Reserved									
0x10	INTPEND	15:8	BUSY	READY					EVD	OVR
		7:0						ID[2:0]		
0x12 ... 0x13	Reserved									
0x14	INTSTATUS	31:24								
		23:16								
		15:8								
		7:0	CHINT7	CHINT6	CHINT5	CHINT4	CHINT3	CHINT2	CHINT1	CHINT0
0x18	BUSYCH	31:24								
		23:16								
		15:8								
		7:0	BUSYCH7	BUSYCH6	BUSYCH5	BUSYCH4	BUSYCH3	BUSYCH2	BUSYCH1	BUSYCH0
0x1C	READYUSR	31:24								
		23:16								
		15:8								
		7:0	READYUSR7	READYUSR6	READYUSR5	READYUSR4	READYUSR3	READYUSR2	READYUSR1	READYUSR0
0x20	CHANNEL0	31:24								
		23:16								
		15:8	ONDEMAND	RUNSTDBY			EDGSEL[1:0]		PATH[1:0]	
		7:0		EVGEN[6:0]						
0x24	CHINTENCLR0	7:0							EVD	OVR
0x25	CHINTENSET0	7:0							EVD	OVR
0x26	CHINTFLAG0	7:0							EVD	OVR
0x27	CHSTATUS0	7:0							BUSYCH	RDYUSR
0x28	CHANNEL1	31:24								
		23:16								
		15:8	ONDEMAND	RUNSTDBY			EDGSEL[1:0]		PATH[1:0]	
		7:0		EVGEN[6:0]						
0x2C	CHINTENCLR1	7:0							EVD	OVR
0x2D	CHINTENSET1	7:0							EVD	OVR
0x2E	CHINTFLAG1	7:0							EVD	OVR
0x2F	CHSTATUS1	7:0							BUSYCH	RDYUSR

.....continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x30	CHANNEL2	31:24								
		23:16								
		15:8	ONDEMAND	RUNSTDBY			EDGSEL[1:0]		PATH[1:0]	
		7:0		EVGEN[6:0]						
0x34	CHINTENCLR2	7:0							EVD	OVR
0x35	CHINTENSET2	7:0							EVD	OVR
0x36	CHINTFLAG2	7:0							EVD	OVR
0x37	CHSTATUS2	7:0							BUSYCH	RDYUSR
0x38	CHANNEL3	31:24								
		23:16								
		15:8	ONDEMAND	RUNSTDBY			EDGSEL[1:0]		PATH[1:0]	
		7:0		EVGEN[6:0]						
0x3C	CHINTENCLR3	7:0							EVD	OVR
0x3D	CHINTENSET3	7:0							EVD	OVR
0x3E	CHINTFLAG3	7:0							EVD	OVR
0x3F	CHSTATUS3	7:0							BUSYCH	RDYUSR
0x40	CHANNEL4	31:24								
		23:16								
		15:8	ONDEMAND	RUNSTDBY			EDGSEL[1:0]		PATH[1:0]	
		7:0		EVGEN[6:0]						
0x44	CHINTENCLR4	7:0							EVD	OVR
0x45	CHINTENSET4	7:0							EVD	OVR
0x46	CHINTFLAG4	7:0							EVD	OVR
0x47	CHSTATUS4	7:0							BUSYCH	RDYUSR
0x48	CHANNEL5	31:24								
		23:16								
		15:8	ONDEMAND	RUNSTDBY			EDGSEL[1:0]		PATH[1:0]	
		7:0		EVGEN[6:0]						
0x4C	CHINTENCLR5	7:0							EVD	OVR
0x4D	CHINTENSET5	7:0							EVD	OVR
0x4E	CHINTFLAG5	7:0							EVD	OVR
0x4F	CHSTATUS5	7:0							BUSYCH	RDYUSR
0x50	CHANNEL6	31:24								
		23:16								
		15:8	ONDEMAND	RUNSTDBY			EDGSEL[1:0]		PATH[1:0]	
		7:0		EVGEN[6:0]						
0x54	CHINTENCLR6	7:0							EVD	OVR
0x55	CHINTENSET6	7:0							EVD	OVR
0x56	CHINTFLAG6	7:0							EVD	OVR
0x57	CHSTATUS6	7:0							BUSYCH	RDYUSR
0x58	CHANNEL7	31:24								
		23:16								
		15:8	ONDEMAND	RUNSTDBY			EDGSEL[1:0]		PATH[1:0]	
		7:0		EVGEN[6:0]						
0x5C	CHINTENCLR7	7:0							EVD	OVR
0x5D	CHINTENSET7	7:0							EVD	OVR
0x5E	CHINTFLAG7	7:0							EVD	OVR
0x5F	CHSTATUS7	7:0							BUSYCH	RDYUSR
0x60	CHANNEL8	31:24								
		23:16								
		15:8	ONDEMAND	RUNSTDBY			EDGSEL[1:0]		PATH[1:0]	
		7:0		EVGEN[6:0]						
0x64 ... 0x67	Reserved									
0x68	CHANNEL9	31:24								
		23:16								
		15:8	ONDEMAND	RUNSTDBY			EDGSEL[1:0]		PATH[1:0]	
		7:0		EVGEN[6:0]						

.....continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x6C ... 0x6F	Reserved									
0x70	CHANNEL10	31:24								
		23:16								
		15:8	ONDEMAND	RUNSTDBY			EDGSEL[1:0]		PATH[1:0]	
		7:0		EVGEN[6:0]						
0x74 ... 0x77	Reserved									
0x78	CHANNEL11	31:24								
		23:16								
		15:8	ONDEMAND	RUNSTDBY			EDGSEL[1:0]		PATH[1:0]	
		7:0		EVGEN[6:0]						
0x7C ... 0x011F	Reserved									
0x0120	USER0	7:0					CHANNEL[3:0]			
...										
0x0153	USER51	7:0					CHANNEL[3:0]			
0x0154 ... 0x01D3	Reserved									
0x01D4	INTENCLR	7:0								NSCHK
0x01D5	INTENSET	7:0								NSCHK
0x01D6	INTFLAG	7:0								NSCHK
0x01D7	Reserved									
0x01D8	NONSECCHAN	31:24								
		23:16								
		15:8					CHANNEL11	CHANNEL10	CHANNEL9	CHANNEL8
		7:0	CHANNEL7	CHANNEL6	CHANNEL5	CHANNEL4	CHANNEL3	CHANNEL2	CHANNEL1	CHANNEL0
0x01DC	NSCHKCHAN	31:24								
		23:16								
		15:8					CHANNEL11	CHANNEL10	CHANNEL9	CHANNEL8
		7:0	CHANNEL7	CHANNEL6	CHANNEL5	CHANNEL4	CHANNEL3	CHANNEL2	CHANNEL1	CHANNEL0
0x01E0	NONSECUSER0	31:24	USER31	USER30	USER29	USER28	USER27	USER26	USER25	USER24
		23:16	USER23	USER22	USER21	USER20	USER19	USER18	USER17	USER16
		15:8	USER15	USER14	USER13	USER12	USER11	USER10	USER9	USER8
		7:0	USER7	USER6	USER5	USER4	USER3	USER2	USER1	USER0
0x01E4	NONSECUSER1	31:24								
		23:16					USER51	USER50	USER49	USER48
		15:8	USER47	USER46	USER45	USER44	USER43	USER42	USER41	USER40
		7:0	USER39	USER38	USER37	USER36	USER35	USER34	USER33	USER32
0x01E8 ... 0x01EF	Reserved									
0x01F0	NSCHKUSER0	31:24	USER31	USER30	USER29	USER28	USER27	USER26	USER25	USER24
		23:16	USER23	USER22	USER21	USER20	USER19	USER18	USER17	USER16
		15:8	USER15	USER14	USER13	USER12	USER11	USER10	USER9	USER8
		7:0	USER7	USER6	USER5	USER4	USER3	USER2	USER1	USER0
0x01F4	NSCHKUSER1	31:24								
		23:16					USER51	USER50	USER49	USER48
		15:8	USER47	USER46	USER45	USER44	USER43	USER42	USER41	USER40
		7:0	USER39	USER38	USER37	USER36	USER35	USER34	USER33	USER32

32.6.1 Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00
Property: PAC Write-Protection , Secure

Bit	7	6	5	4	3	2	1	0
								SWRST
Access								W/-/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 it is recommended to disable the event generators.

32.6.2 Software Event

Name: SWEVT
Offset: 0x04
Reset: 0x00000000
Property: PAC Write-Protection, Mix-Secure

Important: For **PIC32CM LS00/LS60 Non-Secure** accesses, write accesses (W*) are allowed only if the security attribution for the corresponding channel (CHANNELx) is set as Non-Secured in the NONSECCHAN 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
					CHANNEL11	CHANNEL10	CHANNEL9	CHANNEL8
Access					W/W*/W	W/W*/W	W/W*/W	W/W*/W
Reset					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*/W	W/W*/W	W/W*/W	W/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 – CHANNELx Channel x Software Selection [x=0..7]
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will trigger a software event for channel x.
These bits always return '0' when read.

32.6.3 Priority Control

Name: PRICTRL
Offset: 0x08
Reset: 0x00
Property: PAC Write-Protection, Secure

Bit	7	6	5	4	3	2	1	0
	RREN					PRI[2:0]		
Access	RW/RW/RW					RW/-/RW	RW/-/RW	RW/-/RW
Reset	0					0	0	0

Bit 7 – RREN Round-Robin Scheduling Enable
For details on scheduling schemes, refer to Interrupt Status and Interrupts Arbitration

Value	Description
0	Static scheduling scheme for channels with level priority
1	Round-robin scheduling scheme for channels with level priority

Bits 2:0 – PRI[2:0] Channel Priority Number
When round-robin arbitration is enabled (PRICTRL.RREN=1) for 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 priority level, and the value of this bit group is nonzero, it will not affect the static priority scheme.
This bit group is not reset when round-robin scheduling gets disabled (PRICTRL.RREN written to zero).

32.6.4 Channel Pending Interrupt

Name: INTPEND
Offset: 0x10
Reset: 0x4000
Property: Secure

An interrupt that handles several channels should 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	R/-/R					RW/-/RW	RW/-/RW
Reset	0	1					0	0

Bit	7	6	5	4	3	2	1	0
						ID[2:0]		
Access						RW/-/RW	RW/-/RW	RW/-/RW
Reset						0	0	0

Bit 15 – BUSY Busy
This bit is read '1' when the event on a channel selected by Channel ID field (ID) has not been handled by all the event users connected to this channel.

Bit 14 – READY Ready
This bit is read '1' when all event users connected to the channel selected by 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 will be 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 will clear it. It will also clear 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 will be generated if CHINTENCLR/SET.OVRx is '1'.
There are two possible overrun channel conditions:

- One or more of the event users on channel selected by Channel ID field (ID) are not ready when a new event occurs
- An event happens when the previous event on channel selected by Channel ID field (ID) 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 it. It will also clear 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 2:0 – ID[2:0] 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.

32.6.5 Interrupt Status

Name: INTSTATUS
Offset: 0x14
Reset: 0x00000000
Property: Mix-Secure

Important: For **PIC32CM LS00/LS60 Non-Secure** accesses, read accesses (R*) are allowed only if the security attribution for the corresponding channel (CHANNELx) is set as Non-Secured in the NONSECCHAN 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
	CHINT7	CHINT6	CHINT5	CHINT4	CHINT3	CHINT2	CHINT1	CHINT0
Access	R/R*/R	R/R*/R	R/R*/R	R/R*/R	R/R*/R	R/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 – 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.

32.6.6 Busy Channels

Name: BUSYCH
Offset: 0x18
Reset: 0x00000000
Property: Mix-Secure

Important: For **PIC32CM LS00/LS60 Non-Secure** accesses, read accesses (R*) are allowed only if the security attribution for the corresponding channel (CHANNELx) is set as Non-Secured in the NONSECCHAN 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
	BUSYCH7	BUSYCH6	BUSYCH5	BUSYCH4	BUSYCH3	BUSYCH2	BUSYCH1	BUSYCH0
Access	R/R*/R	R/R*/R	R/R*/R	R/R*/R	R/R*/R	R/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 – BUSYCHx Busy Channel x
This bit is set if an event occurs on channel x has not been 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 '0'.

32.6.7 Ready Users

Name: READYUSR
Offset: 0x1C
Reset: 0x000000FF
Property: Mix-Secure

Important: For **PIC32CM LS00/LS60 Non-Secure** accesses, read accesses (R*) are allowed only if the security attribution for the corresponding channel (CHANNELx) is set as Non-Secured in the NONSECCHAN 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
	READYUSR7	READYUSR6	READYUSR5	READYUSR4	READYUSR3	READYUSR2	READYUSR1	READYUSR0
Access	R/R*/R	R/R*/R	R/R*/R	R/R*/R	R/R*/R	R/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 – 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.

32.6.8 Channel n Control

Name: CHANNELn
Offset: 0x20 + n*0x08 [n=0..11]
Reset: 0x00008000 (CHANNEL0-7), 0x00000000 (CHANNEL8-11)
Property: PAC Write-Protection, Mix-Secure

Important: For **PIC32CM LS00/LS60 Non-Secure** accesses, read and write accesses (RW*) are allowed only if the security attribution for the corresponding channel (CHANNELn) is set as Non-Secured in the NONSECCHAN register.

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			RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW
Reset	x	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/RW*/RW	RW/RW*/RW	RW/RW*/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.
This bit has no effect for channels when asynchronous path is selected or for channels with asynchronous support only.
This bit is always read zero for channels with asynchronous support only.

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, for a resynchronized channel.
This bit has no effect for channels when asynchronous path is selected or for channels with asynchronous support only.
This bit is always read zero for channels with asynchronous support only.

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 will be used by the selected channel.

Note: The path choice can be limited by the channel source, see the table in 32.6.13. USERm.

Important: Only EVSYS channel 0 to 7 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 32-3. Event Generators

Value	Event Generator	Description
0 (0x00)	NONE	No event generator selected
1 (0x01)	CFD	XOSC clock failure detection (OSCCTRL)
2 (0x02)	CFD	XOSC32K clock failure detection (OSC32KCTRL)
3 (0x03)	BOD33DET	SUPC BOD33 detection
4-11 (0x04-0x0B)	PER0-7	RTC period
4 (0x04)	ALARM0	RTC alarm (Clock/Calendar mode)
12-13 (0x0C-0x0D)	CMP0-1	RTC comparison (COUNT16 and COUNT32 modes)
14 (0x0E)	TAMPER	RTC tamper detection
15 (0x0F)	OVF	RTC overflow
16 (0x10)	PERD	RTC periodic interval daily
17-32 (0x11-0x20)	EXTINT0-15	EIC external interrupt
33- 40 (0x21-0x28)	CH0-7	DMAC channel
41 (0x29)	OVF	TC0 overflow
42-43 (0x2A-0x2B)	MC0-1	TC0 match/compare
44 (0x2C)	OVF	TC1 overflow
45-46 (0x2D-0x2E)	MC0-1	TC1 match/compare
47 (0x2F)	OVF	TC2 overflow
48-49 (0x30-0x31)	MC0-1	TC2 match/compare
50 (0x32)	TRG	TCC0 trigger
51 (0x33)	CNT	TCC0 counter
52-55 (0x34-0x37)	MC0-3	TCC0 match/compare
56 (0x38)	OVF	TCC0 overflow/underflow
57 (0x39)	TRG	TCC1 trigger
58 (0x3A)	CNT	TCC1 counter
59-50 (0x3B-0x3C)	MC0-1	TCC1 match/compare
61 (0x3D)	OVF	TCC1 overflow/underflow

.....continued

Value	Event Generator	Description
62 (0x3E)	TRG	TCC2 trigger
63 (0x3F)	CNT	TCC2 counter
64-65 (0x40-0x41)	MC0-1	TCC2 match/compare
66 (0x42)	OVF	TCC2 overflow/underflow
67 (0x43)	TRG	TCC3 trigger
68 (0x44)	CNT	TCC3 counter
69-72 (0x45-0x48)	MC0-3	TCC3 match/compare
73 (0x49)	OVF	TCC3 overflow/underflow
74 (0x4A)	RESRDY	ADC resolution ready
75 (0x4B)	WINMON	ADC window monitor
76-79 (0x4C-0x4F)	COMP0-3	AC comparator
80-81 (0x50-0x51)	WIN0-1	AC window
82-83 (0x52-0x53)	EMPTY0-1	DAC empty
84 (0x54)	EOC	PTC end of conversion
85 (0x55)	WCOMP	PTC window comparator
86 (0x56)	DATARDY	TRNG Data Ready
87-90 (0x57-0x5A)	LUT0-3	CCL LUT output
91 (0x5B)	ERR	PAC access error

32.6.9 Channel n Interrupt Enable Clear

Name: CHINTENCLR
Offset: 0x24 + n*0x08 [n=0..7]
Reset: 0x00
Property: PAC Write-Protection, Mix-Secure

Important: For **PIC32CM LS00/LS60 Non-Secure** accesses, read and write accesses (RW*) are allowed only if the security attribution for the corresponding channel (CHANNELx) is set as Non-Secured in the NONSECCHAN register.

Bit	7	6	5	4	3	2	1	0
							EVD	OVR
Access							RW/RW*/RW	RW/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 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 Enable
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.

32.6.10 Channel n Interrupt Enable Set

Name: CHINTENSET
Offset: 0x25 + n*0x08 [n=0..7]
Reset: 0x00
Property: PAC Write-Protection, Mix-Secure

Important: For **PIC32CM LS00/LS60 Non-Secure** accesses, read and write accesses (RW*) are allowed only if the security attribution for the corresponding channel (CHANNELx) is set as Non-Secured in the NONSECCHAN register.

Bit	7	6	5	4	3	2	1	0
							EVD	OVR
Access							RW/RW*/RW	RW/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.

32.6.11 Channel n Interrupt Flag Status and Clear

Name: CHINTFLAG
Offset: 0x26 + n*0x08 [n=0..7]
Reset: 0x00
Property: Mix-Secure

Important: For **PIC32CM LS00/LS60 Non-Secure** accesses, read and write accesses (RW*) are allowed only if the security attribution for the corresponding channel (CHANNELx) is set as Non-Secured in the NONSECCHAN register.

Bit	7	6	5	4	3	2	1	0
							EVD	OVR
Access							RW/RW*/RW	RW/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.OVRx 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.

32.6.12 Channel n Status

Name: CHSTATUS
Offset: 0x27 + n*0x08 [n=0..7]
Reset: 0x01
Property: Mix-Secure

Important: For **PIC32CM LS00/LS60 Non-Secure** accesses, read accesses (R*) are allowed only if the security attribution for the corresponding channel (CHANNELx) is set as Non-Secured in the NONSECCHAN register.

Bit	7	6	5	4	3	2	1	0
							BUSYCH	RDYUSR
Access							R/R*/R	R/R*/R
Reset							0	0

Bit 1 – BUSYCH Busy Channel
This bit is cleared when channel is idle.
This bit is set if an event on channel has not been handled by all event users connected to channel.
When the event channel path is asynchronous, this bit is always read '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 channel are ready to handle incoming events on the channel.
When the event channel path is asynchronous, this bit is always read zero.

32.6.13 Event User m

Name: USERm
Offset: 0x0120 + m*0x01 [m=0..51]
Reset: 0x0
Property: PAC Write-Protection, Mix-Secure

Table 32-4. User Multiplexer Number m

USERm	User Multiplexer	Description	Path Type
m=0	TUNE	DFLLULP Tune (OSCCTRL)	A
m=1	TAMPEVT	RTC Tamper	A,S,R
m=2	AUTOW	NVMCTRL Auto-Write	A
m=3..6	EVU0-3	PORT Event	A
m=7..14	CH0-7	DMA Channel Event	A,S,R
m=15	EVU	TC0 EVU	A
m=16	EVU	TC1 EVU	A
m=17	EVU	TC2 EVU	A
m=18..19	EV0-1	TCC0 EVU	A,S,R
m=20..23	MC0-3	TCC0 MC	A,S,R
m=24..25	EV0-1	TCC1 EVU	A,S,R
m=26..27	MC0-1	TCC1 MC	A,S,R
m=28..29	EV0-1	TCC2 EVU	A,S,R
m=30..31	MC0-1	TCC2 MC	A,S,R
m=32..33	EV0-1	TCC3 EVU	A,S,R
m=34..37	MC0-3	TCC3 MC	A,S,R
m=38	START	ADC Start Conversion	A,S,R
m=39	FLUSH	Flush ADC	A,S,R
m=40..43	COMP0-3	AC Start Comparator	A
m=44..45	START0-1	DAC Start Conversion	A
m=46	STCONV	PTC Start Conversion	A,S,R
m=47	DSEQR	PTC Sequencing	A,S,R
m=48..51	LUT0-3	CCL LUT Input	A

Note:
1. A = Asynchronous path, S = Synchronous path, R = Resynchronized path.

Important: For **PIC32CM LS00/LS60 Non-Secure** accesses, read and write accesses (RW*) are allowed only if the security attribution for the corresponding event user (USERx) is set as Non-Secured in the NONSECUSER register.

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

Bits 3:0 – CHANNEL[3:0] 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.

Value	Description
0x00	No channel selected
0x01	Channel 0 selected
0x02	Channel 1 selected
0x03	Channel 2 selected
0x04	Channel 3 selected
0x05	Channel 4 selected
0x06	Channel 5 selected
0x07	Channel 6 selected
0x08	Channel 7 selected
0x09	Channel 8 selected
0x0A	Channel 9 selected
0x0B	Channel 10 selected
0x0C	Channel 11 selected
0x0D-0x0F	Reserved

32.6.14 Interrupt Enable Clear

Name: INTENCLR
Offset: 0x1D4
Reset: 0x0
Property: PAC Write-Protection

Important: This register is only available for **PIC32CM LS00/LS60** and has no effect for **PIC32CM LE00**.

Bit	7	6	5	4	3	2	1	0
								NSCHK
Access								RW/RW/RW
Reset								0

Bit 0 – NSCHK Non-Secure Check Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the Non-Secure Check Interrupt Enable bit, which disables the Non-Secure Check interrupt.

Value	Description
0	The Non-Secure Check interrupt is disabled.
1	The Non-Secure Check interrupt is enabled.

32.6.15 Interrupt Enable Set

Name: INTENSET
Offset: 0x1D5
Reset: 0x0
Property: PAC Write-Protection

Important: This register is only available for **PIC32CM LS00/LS60** and has no effect for **PIC32CM LE00**.

Bit	7	6	5	4	3	2	1	0
								NSCHK
Access								RW/RW/RW
Reset								0

Bit 0 – NSCHK Non-Secure Check Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will set the Non-Secure Check Interrupt Enable bit, which disables the Non-Secure Check interrupt.

Value	Description
0	The Non-Secure Check interrupt is disabled.
1	The Non-Secure Check interrupt is enabled.

32.6.16 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x1D6
Reset: 0x0
Property: -

Important: This register is only available for **PIC32CM LS00/LS60** and has no effect for **PIC32CM LE00**.

Bit	7	6	5	4	3	2	1	0
								NSCHK
Access								RW/RW/RW
Reset								0

Bit 0 – NSCHK Non-Secure Check
This flag is set when a bit in NSCHKCHAN is 1 and the corresponding bit in NONSECCHAN is cleared, or when a bit in NSCHKCHAN is 0 and the corresponding bit in NONSECCHAN is set, or when a bit in NSCHKUSER is 1 and the corresponding bit in NONSECUSER is cleared, or when a bit in NSCHKUSER is 0 and the corresponding bit in NONSECUSER is set.
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the Non-Secure Check interrupt flag.

32.6.17 Channel Security Attribution

Name: NONSECCHAN
Offset: 0x1D8
Reset: 0x00000000
Property: PAC Write-Protection, Write-Secure

This register allows the user to configure one or more channels as secured or non-secured.

Important: This register is only available for **PIC32CM LS00/LS60** and has no effect for **PIC32CM LE00**.

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
					CHANNEL11	CHANNEL10	CHANNEL9	CHANNEL8
Access					RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CHANNEL7	CHANNEL6	CHANNEL5	CHANNEL4	CHANNEL3	CHANNEL2	CHANNEL1	CHANNEL0
Access	RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11 – CHANNELn Channel n Security Attribution [n=0..7]
The bit n of CHANNEL enables the non-secure mode of CHANNELn. The registers whose CHANNEL bit or bitfield n is set in non-secure mode by NONSECCHAN.CHANNELn are CHANNELn, CHINTENCLRn, CHINTENSETn, CHINTFLAGn and CHSTATUSn registers.
These bits set the security attribution for the individual channels.

Value	Description
0	The corresponding channel is secured. When the module is PAC secured, the configuration and status bits for this channel are only available through the secure alias. Attempts to change the channel configuration through the non-secure alias will be silently ignored and reads will return 0.
1	The corresponding channel is non-secured. The configuration and status bits for this channel are available through the non-secure alias.

32.6.18 Channel Security Attribution Check

Name: NSCHKCHAN
Offset: 0x1DC
Reset: 0x00000000
Property: PAC Write-Protection

This register allows the user to select one or more channels to check their security attribution as non-secured.

Important: This register is only available for **PIC32CM LS00/LS60** and has no effect for **PIC32CM LE00**.

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
					CHANNEL11	CHANNEL10	CHANNEL9	CHANNEL8
Access					RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/RW/RW
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CHANNEL7	CHANNEL6	CHANNEL5	CHANNEL4	CHANNEL3	CHANNEL2	CHANNEL1	CHANNEL0
Access	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/RW/RW
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11 – CHANNELn Channel n Selection [n=0..7]
These bits selects the individual channels for security attribution check. If any channel selected in NSCHKCHAN has the corresponding bit in NONSECCHAN set to the opposite value, then the NSCHK interrupt flag will be set.

Value	Description
0	0-to-1 transition will be detected on corresponding NONSECCHAN bit.
1	1-to-0 transition will be detected on corresponding NONSECCHAN bit.

32.6.19 Event User Security Attribution

Name: NONSECUSER0
Offset: 0x1E0
Reset: 0x00000000
Property: PAC Write-Protection, Write-Secure

This register allows the user to configure one or more event users as secured or non-secured.

Important: This register is only available for **PIC32CM LS00/LS60** and has no effect for **PIC32CM LE00**.

Bit	31	30	29	28	27	26	25	24
	USER31	USER30	USER29	USER28	USER27	USER26	USER25	USER24
Access	RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	USER23	USER22	USER21	USER20	USER19	USER18	USER17	USER16
Access	RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	USER15	USER14	USER13	USER12	USER11	USER10	USER9	USER8
Access	RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	USER7	USER6	USER5	USER4	USER3	USER2	USER1	USER0
Access	RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW
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 – USER Event User n Security Attribution [n=0..31]

The bit n of USER enables the non-secure mode of USERn. The registers whose USER bit or bitfield n is set in non-secure mode by NONSECUSER.USERn are USERn registers.
These bits set the security attribution for the individual event users.

Value	Description
0	The corresponding event user is secured. When the module is PAC secured, the configuration and status bits for this event user are only available through the secure alias. Attempts to change the event user configuration through the non-secure alias will be silently ignored and reads will return 0.
1	The corresponding event user is non-secured. The configuration and status bits for this event user are available through the non-secure alias.

32.6.20 Event User Security Attribution

Name: NONSECUSER1
Offset: 0x1E4
Reset: 0x00000000
Property: PAC Write-Protection, Write-Secure

This register allows the user to configure one or more event users as secured or non-secured.

Important: This register is only available for **PIC32CM LS00/LS60** and has no effect for **PIC32CM LE00**.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					USER51	USER50	USER49	USER48
Access					RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
	USER47	USER46	USER45	USER44	USER43	USER42	USER41	USER40
Access	RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	USER39	USER38	USER37	USER36	USER35	USER34	USER33	USER32
Access	RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW	RW/R/RW
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 – USER Event User n Security Attribution [n=32..51]
The bit n of USER enables the non-secure mode of USERn. The registers whose USER bit or bitfield n is set in non-secure mode by NONSECUSER.USERn are USERn registers.
These bits set the security attribution for the individual event users.

Value	Description
0	The corresponding event user is secured. When the module is PAC secured, the configuration and status bits for this event user are only available through the secure alias. Attempts to change the event user configuration through the non-secure alias will be silently ignored and reads will return 0.
1	The corresponding event user is non-secured. The configuration and status bits for this event user are available through the non-secure alias.

32.6.21 Event User Security Attribution Check

Name: NSCHKUSER0
Offset: 0x1F0
Reset: 0x00000000
Property: PAC Write-Protection

This register allows the user to select one or more event users to check their security attribution as non-secured.

Important: This register is only available for **PIC32CM LS00/LS60** and has no effect for **PIC32CM LE00**.

Bit	31	30	29	28	27	26	25	24
	USER31	USER30	USER29	USER28	USER27	USER26	USER25	USER24
Access	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/RW/RW
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	USER23	USER22	USER21	USER20	USER19	USER18	USER17	USER16
Access	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/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
	USER15	USER14	USER13	USER12	USER11	USER10	USER9	USER8
Access	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/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
	USER7	USER6	USER5	USER4	USER3	USER2	USER1	USER0
Access	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/RW/RW
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 – USER Event User n Selection [n=0..31]

These bits selects the individual event users for security attribution check. If any event user selected in NSCHKUSER has the corresponding bit in NONSECUSER set to the opposite value, then the NSCHK interrupt flag will be set.

Value	Description
0	0-to-1 transition will be detected on corresponding NONSECUSER bit.
1	1-to-0 transition will be detected on corresponding NONSECUSER bit.

32.6.22 Event User Security Attribution Check

Name: NSCHKUSER1
Offset: 0x1F4
Reset: 0x00000000
Property: PAC Write-Protection

This register allows the user to select one or more event users to check their security attribution as non-secured.

Important: This register is only available for **PIC32CM LS00/LS60** and has no effect for **PIC32CM LE00**.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					USER51	USER50	USER49	USER48
Access					RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/RW/RW
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
	USER47	USER46	USER45	USER44	USER43	USER42	USER41	USER40
Access	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/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
	USER39	USER38	USER37	USER36	USER35	USER34	USER33	USER32
Access	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/RW/RW	RW/RW/RW
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 – USER Event User n Selection [n=32..51]
These bits selects the individual event users for security attribution check. If any event user selected in NSCHKUSER has the corresponding bit in NONSECUSER set to the opposite value, then the NSCHK interrupt flag will be set.

Value	Description
0	0-to-1 transition will be detected on corresponding NONSECUSER bit.
1	1-to-0 transition will be detected on corresponding NONSECUSER bit.

33. Serial Communication Interface (SERCOM)

33.1 Overview

There are up to six instances of the serial communication interface (SERCOM) peripheral.

A SERCOM can be configured to support a number of modes: I^2C, SPI, and USART. When an instance of SERCOM is configured and enabled, all of the resources of that SERCOM instance will be dedicated to the selected mode.

The SERCOM serial engine consists of a transmitter and receiver, baud-rate generator and address matching functionality. It can use the internal generic clock or an external clock. Using an external clock allows the SERCOM to be operated in all Sleep modes.

33.2 Features

- Interface for configuring into one of the following:
  - Inter-Integrated Circuit (I^2C) Two-wire Serial Interface
  - System Management Bus (SMBus™) compatible
  - Serial Peripheral Interface (SPI)
  - Universal Synchronous/Asynchronous Receiver/Transmitter (USART)
- Baud-rate generator
- Address match/mask logic
- Operational in all Sleep modes with an external clock source
- Can be used with DMA
- Receive buffer: 8-bytes FIFO
- Transmit buffer: 8-bytes FIFO
- 32-bit extension for better system bus utilization
- Secure pin multiplexing to isolate on dedicated I/O pins a secured communication with external devices from the non-secure application (PIC32CM LS00 only)

The following table lists the supported features for each SERCOM instance:

Table 33-1. SERCOM Features Summary (PIC32CM LE00 and PIC32CM LS00)

	SERCOM Instance					
Protocol	**SERCOM0**	**SERCOM1**	**SERCOM2**	**SERCOM3**	**SERCOM4**	**SERCOM5**
SPI	Yes					
USART	Yes					
I2C	Yes	Yes	Yes	Yes(1)	Yes	Yes(1)
Secure Pin Multiplexing (**PIC32CM LS00 only**)	No	Yes	No	No	No	No

Note:

1. I^2C is not supported on SERCOM3 and SERCOM5 for 48-pin packages. Refer to the Pinout for more information.

Table 33-2. SERCOM Features Summary (PIC32CM LS60)

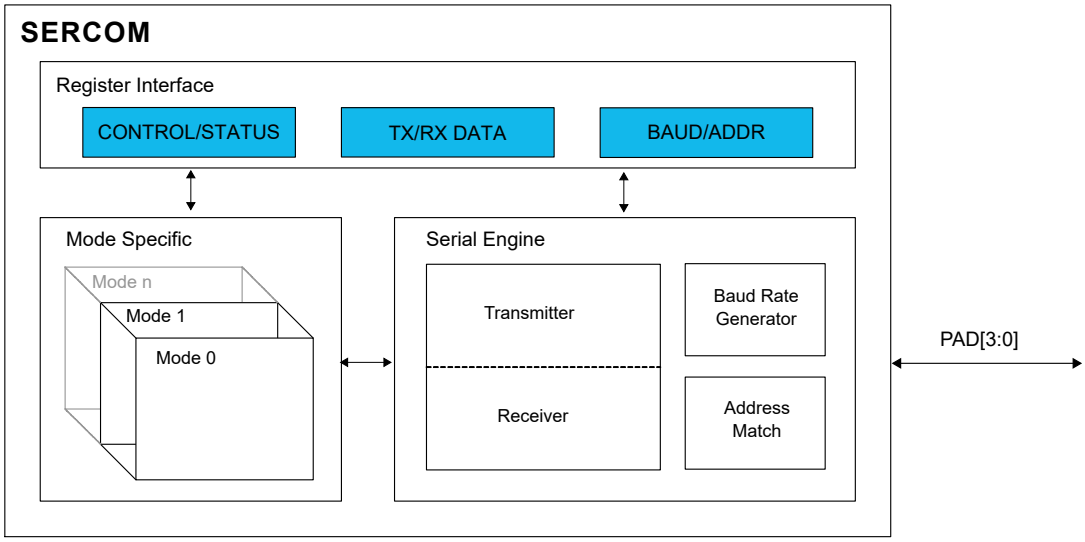
	SERCOM Instance					
Protocol	**SERCOM0**	**SERCOM1**	**SERCOM2**	**SERCOM3**	**SERCOM4**	**SERCOM5**
SPI	Yes	Reserved for ATECC608B (I^2C)	Yes			
USART	Yes		Yes			
I2C	Yes		Yes	Yes(1)	Yes	Yes(1)
Secure Pin Multiplexing	No		No	No	No	No

Note:

1. I^2C is not supported on SERCOM3 and SERCOM5 for 48-pin packages. Refer to the Pinout for more information.

33.3 Block Diagram

Figure 33-1. SERCOM Block Diagram



33.4 Signal Description

See the respective SERCOM mode chapters for details:

- SERCOM USART
- SERCOM SPI
- SERCOM I^2C

⚠ CAUTION I/Os for SERCOM peripherals are grouped into IO sets, listed in their 'IOSET' column in the pinout tables. For these peripherals, it is mandatory to use I/Os that belong to the same IO set. The timings are not guaranteed when IOs from different IO sets are mixed. Refer to the Pinout and Packaging chapter to get IOSET definitions.

33.5 Peripheral Dependencies

Table 33-3. SERCOM Configuration Summary

Peripheral name	Base address	NVIC IRQ Index	MCLK AHB/APB Clocks	GCLK Peripheral Channel Index (GCLK.PCHCTRL)	PAC Peripheral Identifier Index (PAC.WRCTRL)	DMA Trigger Source Index (DMAC.CHCTRLB)	Power Domain (PM.STDBYCFG)
SERCOM0	0x42000400	31: bit 0 32: bit 1 33: bit 2 34: bit 3-7	CLK_SERCOM0_APB	17: GCLK_SERCOM0_CORE 16: GCLK_SERCOM0_SLOW	1	4: RX 5: TX	PDSW
SERCOM1	0x42000800	35: bit 0 36: bit 1 37: bit 2 38: bit 3-7	CLK_SERCOM1_APB	18: GCLK_SERCOM1_CORE 16: GCLK_SERCOM1_SLOW	2	6: RX 7: TX	PDSW
SERCOM2	0x42000C00	39: bit 0 40: bit 1 41: bit 2 42: bit 3-7	CLK_SERCOM2_APB	19: GCLK_SERCOM2_CORE 16: GCLK_SERCOM2_SLOW	3	8: RX 9: TX	PDSW
SERCOM3	0x42001000	43: bit 0 44: bit 1 45: bit 2 46: bit 3-7	CLK_SERCOM3_APB	20: GCLK_SERCOM3_CORE 16: GCLK_SERCOM3_SLOW	4	10: RX 11: TX	PDSW
SERCOM4	0x42001400	47: bit 0 48: bit 1 49: bit 2 50: bit 3-7	CLK_SERCOM4_APB	21: GCLK_SERCOM4_CORE 16: GCLK_SERCOM4_SLOW	5	12: RX 13: TX	PDSW
SERCOM5	0x42001800	51: bit 0 52: bit 1 53: bit 2 54: bit 3-7	CLK_SERCOM5_APB	22: GCLK_SERCOM5_CORE 16: GCLK_SERCOM5_SLOW	6	14: RX 15: TX	PDSW

33.6 Functional Description

33.6.1 Principle of Operation

The SERCOM has four internal pads, PAD[3:0], and the signals from I2C, SPI and USART are routed through these SERCOM pads via a multiplexer. The configuration of the multiplexer is available from the different SERCOM modes. Refer to the mode specific chapters for details:

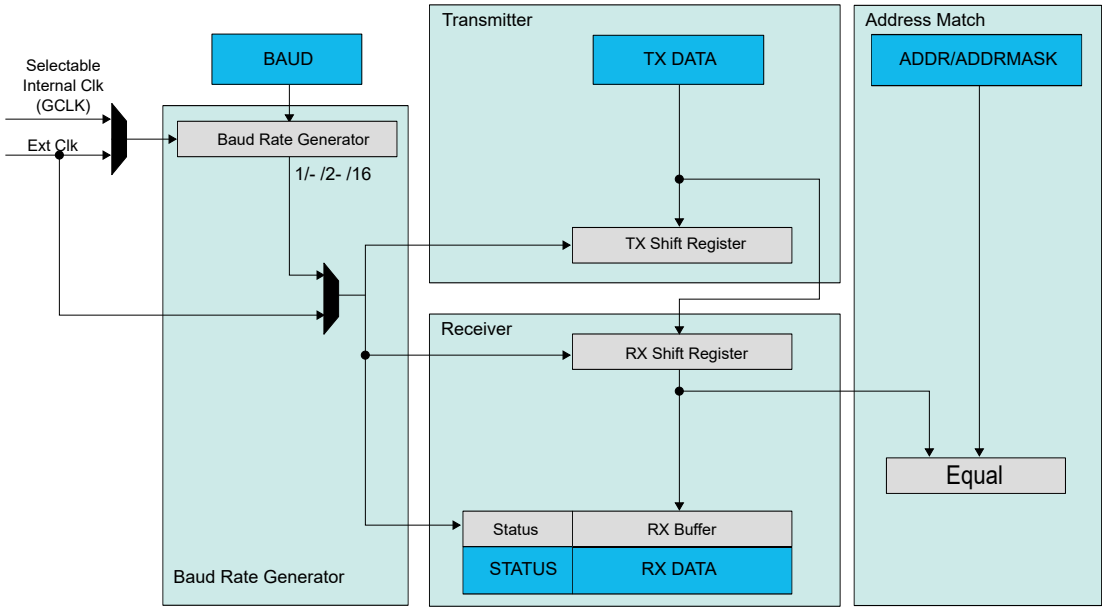
- SERCOM USART
- SERCOM SPI
- SERCOM I^2C

The SERCOM uses up to two generic clocks: GCLK_SERCOMx_CORE and GCLK_SERCOMx_SLOW. The core clock (GCLK_SERCOMx_CORE) is required to clock the SERCOM while working as a host. The slow clock (GCLK_SERCOMx_SLOW) is only required for certain functions. See specific mode chapters for details.

These clocks must be configured and enabled in the Generic Clock Controller (GCLK) before using the SERCOM.

The basic structure of the SERCOM serial engine is shown in the following figure. 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 33-2. SERCOM Serial Engine



The transmitter consists of a minimum of 8-bytes write buffer and a Shift register.

The receiver consists of a minimum of 8-bytes 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.

33.6.2 Basic Operation

33.6.2.1 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 33-4. 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:

- SERCOM USART
- SERCOM SPI
- SERCOM I^2C

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) will reset all registers of this peripheral to their initial states, except the DBGCTRL register, and the peripheral is disabled.

Refer to the CTRLA register description for details.

33.6.2.3 Clock Generation – Baud-Rate Generator

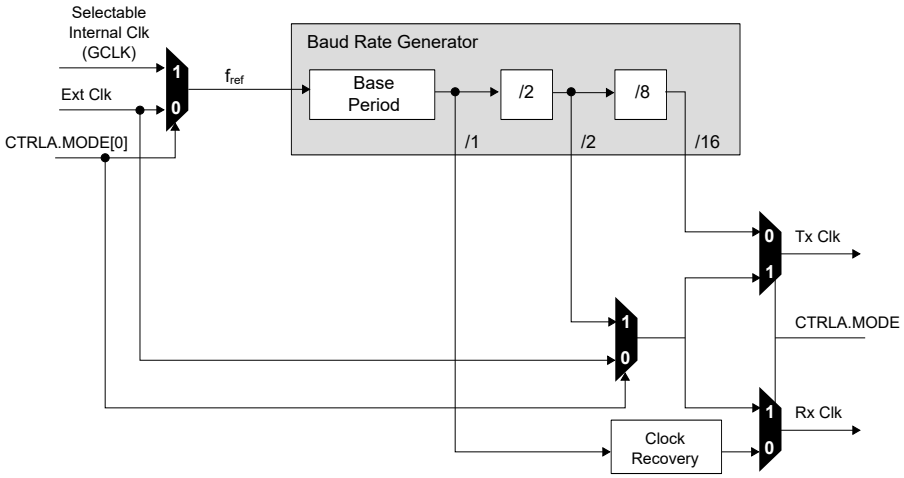
The baud-rate generator, as shown in the following figure, generates internal clocks for asynchronous and synchronous communication. The output frequency (f_{BAUD}) is determined by the Baud register (BAUD) setting and the baud reference frequency (f_{ref}). The baud reference clock is the serial engine 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 33-3. 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 33-5. Baud Rate Equations

Operating Mode	Condition	Baud Rate (Bits Per Second)	BAUD Register Value Calculation
Asynchronous Arithmetic	$f_{BAUD} \leq \frac{f_{ref}}{S}$	$f_{BAUD} = \frac{f_{ref}}{S}\left(1 - \frac{BAUD}{65536}\right)$	$BAUD = 65536 \cdot \left(1 - S \cdot \frac{f_{BAUD}}{f_{ref}}\right)$
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:

$$\text{Error} = 1 - \left(\frac{\text{ExpectedBaudRate}}{\text{ActualBaudRate}}\right)$$

33.6.2.3.1 Asynchronous Arithmetic Mode BAUD Value Selection

The formula given for f_{BAUD} calculates the average frequency over 65536 f_{ref} cycles. Although the BAUD register can be set to any value between 0 and 65536, the actual average frequency of f_{BAUD} over a single frame is more granular. The BAUD register values that will affect the average frequency over a single frame lead to an integer increase in the cycles per frame (CPF)

$$CPF = \frac{f_{ref}}{f_{BAUD}}(D + S)$$

where

- *D* represent the data bits per frame
- *S* represent the sum of start and first stop bits, if present.

Table 33-6 shows the BAUD register value versus baud frequency f_{BAUD} at a serial engine frequency of 48MHz. This assumes a *D* value of 8 bits and an *S* value of 2 bits (10 bits, including start and stop bits).

Table 33-6. BAUD Register Value vs. Baud Frequency

BAUD Register Value	Serial Engine CPF	f_{BAUD} at 48MHz Serial Engine Frequency (f_{REF})
0 – 406	160	3MHz
407 – 808	161	2.981MHz
809 – 1205	162	2.963MHz
...	...	...
65206	31775	15.11kHz
65207	31871	15.06kHz
65208	31969	15.01kHz

33.6.3 Additional Features

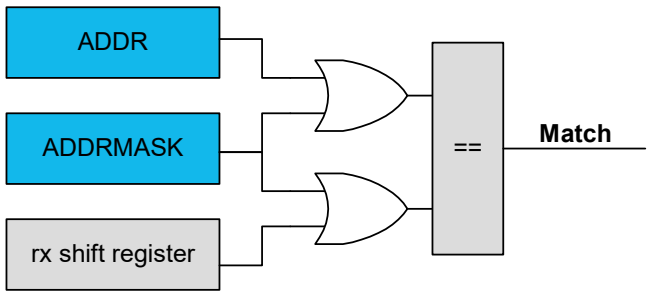
33.6.3.1 Address Match and Mask

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.

33.6.3.1.1 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.

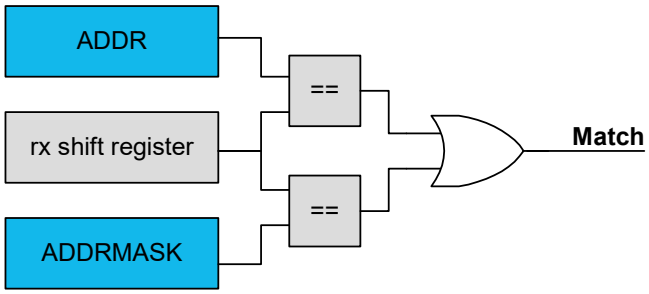
Figure 33-4. Address With Mask



33.6.3.1.2 Two Unique Addresses

The two addresses written to ADDR and ADDRMASK will cause a match.

Figure 33-5. Two Unique Addresses



33.6.3.1.3 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 33-6. Address Range

33.6.3.2 Secure Pin Multiplexing (PIC32CM LS00 only)

The Secure Pin Multiplexing feature can be used on dedicated SERCOM1 I/O pins to isolate a secure communication with external devices from the non-secure application.

Secure Pin Multiplexing is automatically enabled as soon as SERCOM1 is configured as a secure peripheral (PAC Security attribution).

The following table lists the SERCOM1 pins that support the Secure Pin Multiplexing feature:

Table 33-7. Secure Pin Multiplexing on SERCOM1 Pins (PIC32CM LS00 only)

Pin Name	Secure Pin Multiplexing Pad Name
PA16	SERCOM1/PAD[0]
PA17	SERCOM1/PAD[1]
PA18	SERCOM1/PAD[2]
PA19	SERCOM1/PAD[3]

⚠ CAUTION When Secure Pin Multiplexing is enabled, only the SERCOM1 pins from the above table become mapped. All of the alternate SERCOM1 pins are kept in a Hi-Z configuration and cannot be selected.

33.6.4 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:

- SERCOM USART
- SERCOM SPI
- SERCOM I^2C

33.6.5 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. See the 11.2. Nested Vector Interrupt Controller for more information.

33.6.6 Sleep Mode Operation

The peripheral can operate in any sleep mode 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:

- SERCOM USART
- SERCOM SPI
- SERCOM I^2C

33.6.7 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 - refer to the Debug Control (DBGCTRL) register for details.

33.6.8 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.

34. Universal Synchronous and Asynchronous Receiver-Transmitter (SERCOM USART)

34.1 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 34.3. Block Diagram. Labels in uppercase letters are synchronous to CLK_SERCOMx_APB and accessible for CPU. Labels in lowercase letters can be programmed to run on the internal generic clock or an external clock.

The transmitter consists of a 8-bytes write FIFO buffer, a Shift register, and control logic for different frame formats. The write buffer support data transmission without any delay between frames.

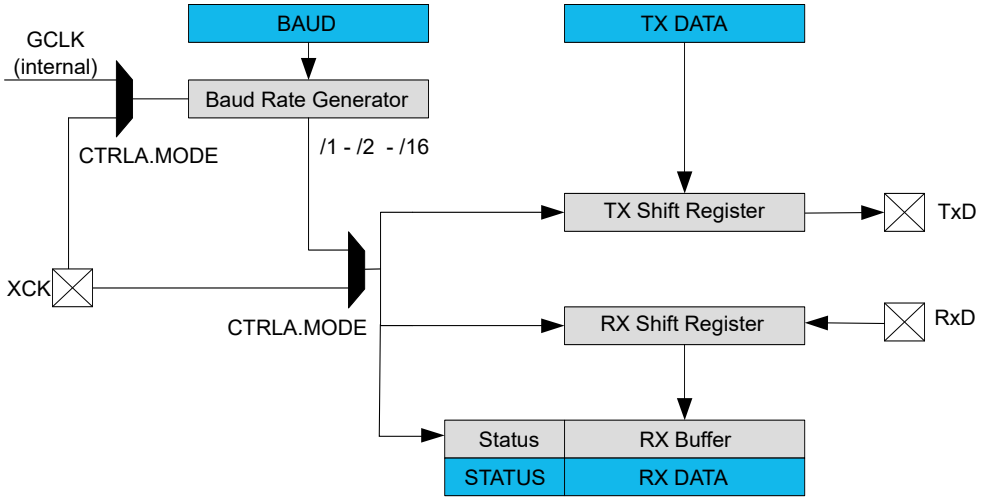
The receiver consists of a 8-bytes receive FIFO buffer and a Shift register. Status information of the received data is available for error checking. Data and clock recovery units ensure robust synchronization and noise filtering during asynchronous data reception.

34.2 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
- Sleep modes operation supported
- RTS and CTS flow control
- IrDA modulation and demodulation up to 115.2kbps
- LIN Host support
- LIN Client support
  - Auto-baud and break character detection
- ISO7816 T=0 or T=1 protocols for Smart Card interfacing
- RS485 Support
- Start-of-frame detection
- Receive buffer: 8-bytes FIFO
- Transmit buffer: 8-bytes FIFO
- DMA operations supported
- 32-bit extension for better system bus utilization

34.3 Block Diagram

Figure 34-1. USART Block Diagram



34.4 Signal Description

Table 34-1. SERCOM USART Signals

Signal Name	Type	Description
PAD[3:0]	Digital I/O	General SERCOM pins

⚠ CAUTION I/Os for SERCOM peripherals are grouped into IO sets, listed in their 'IOSET' column in the pinout tables. For these peripherals, it is mandatory to use I/Os that belong to the same IO set. The timings are not guaranteed when IOs from different IO sets are mixed. Refer to the Pinout and Packaging chapter to get IOSET definitions.

34.5 Peripheral Dependencies

Table 34-2. SERCOM Configuration Summary

Peripheral name	Base address	NVIC IRQ Index	MCLK AHB/APB Clocks	GCLK Peripheral Channel Index (GCLK.PCHCTRL)	PAC Peripheral Identifier Index (PAC.WRCTRL)	DMA Trigger Source Index (DMAC.CHCTRLB)	Power Domain (PM.STDBYCFG)
SERCOM0	0x42000400	31: bit 0 32: bit 1 33: bit 2 34: bit 3-7	CLK_SERCOM0_APB	17: GCLK_SERCOM0_CORE	65	4: RX 5: TX	PDSW
SERCOM1	0x42000800	35: bit 0 36: bit 1 37: bit 2 38: bit 3-7	CLK_SERCOM1_APB	18: GCLK_SERCOM1_CORE	66	6: RX 7: TX	PDSW

.....continued

Peripheral name	Base address	NVIC IRQ Index	MCLK AHB/APB Clocks	GCLK Peripheral Channel Index (GCLK.PCHCTRL)	PAC Peripheral Identifier Index (PAC.WRCTRL)	DMA Trigger Source Index (DMAC.CHCTRLB)	Power Domain (PM.STDBYCFG)
SERCOM2	0x42000C00	39: bit 0 40: bit 1 41: bit 2 42: bit 3-7	CLK_SERCOM2_APB	19: GCLK_SERCOM2_CORE	67	8: RX 9: TX	PDSW
SERCOM3	0x42001000	43: bit 0 44: bit 1 45: bit 2 46: bit 3-7	CLK_SERCOM3_APB	20: GCLK_SERCOM3_CORE	68	10: RX 11: TX	PDSW
SERCOM4	0x42001400	47: bit 0 48: bit 1 49: bit 2 50: bit 3-7	CLK_SERCOM4_APB	21: GCLK_SERCOM4_CORE	69	12: RX 13: TX	PDSW
SERCOM5	0x42001800	51: bit 0 52: bit 1 53: bit 2 54: bit 3-7	CLK_SERCOM5_APB	22: GCLK_SERCOM5_CORE	70	14: RX 15: TX	PDSW

34.6 Functional Description

34.6.1 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

When the SERCOM is used in USART mode, the SERCOM controls the direction and value of the I/O pins according to the table below.

Table 34-3. USART Pin Configuration

Pin	Pin Configuration
TxD	Output
RxD	Input
XCK	Output or input

PORT Control bit PINCFGn.DRVSTR is still effective for the SERCOM output pins.

PORT Control bit PINCFGn.PULLEN is still effective on the SERCOM input pins, but is limited to the enabling/disabling of a pull down only (it is not possible to enable/disable a pull up).

If the receiver or transmitter is disabled, these pins can be used for other purposes.

The combined configuration of PORT and the Transmit Data Pinout and Receive Data Pinout bit fields in the Control A register (CTRLA.TXPO and CTRLA.RXPO, respectively) will define the physical position of the USART signals in the Pin Configuration Summary.

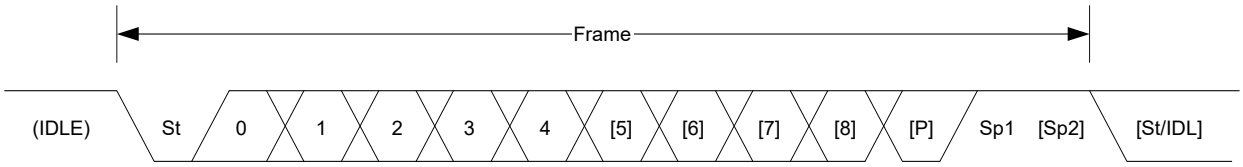
USART data transfer is frame based. A serial frame consists of:

- 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 figure below illustrates the possible frame formats. Brackets denote optional bits.

Figure 34-2. Frame Formats



St Start bit. Signal is always low.

n, [n] Data bits. 0 to [5..9]

[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.

34.6.2 Basic Operation

34.6.2.1 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)
- Control C register (CTRLC)
- Receive Pulse Length register (RXPL)

When the USART is enabled or is being enabled (CTRLA.ENABLE=1), any writing attempt to these registers will be discarded. If the peripheral is being disabled, writing to these registers will be 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 external (0x0) or internal clock (0x1) by writing the Operating Mode value in the CTRLA register (CTRLA.MODE).
2. Select either asynchronous (0) or or synchronous (1) communication mode by writing the Communication Mode bit in the CTRLA register (CTRLA.CMODE).
3. Select 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 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).

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) will reset all registers of this peripheral to their initial states, except the DBGCTRL register, and the peripheral is disabled.

Refer to the CTRLA register description for details.

34.6.2.3 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 a zero to CTRLA.CMODE.

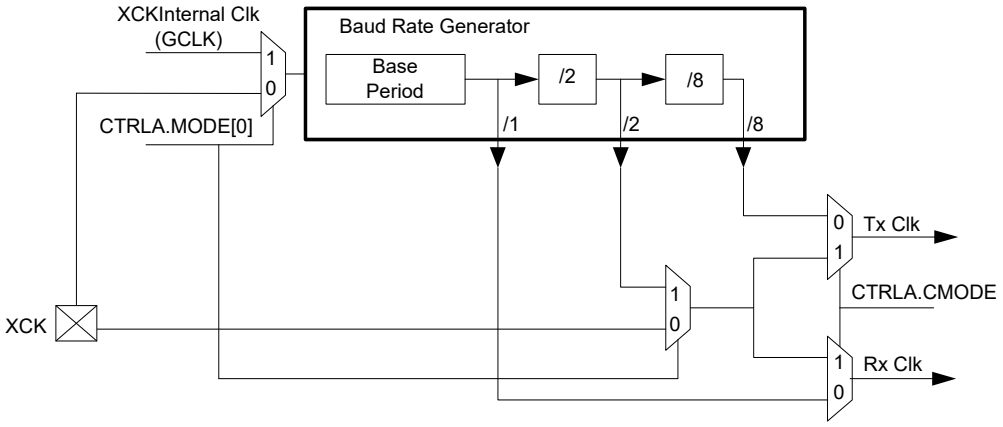
The internal clock source is selected by writing 0x1 to the Operation Mode bit field in the Control A register (CTRLA.MODE), the external clock source is selected by writing 0x0 to CTRLA.MODE.

The SERCOM baud-rate generator is configured as in the figure below.

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. Refer to Clock Generation – Baud-Rate Generator for details on configuring the baud rate.

Figure 34-3. Clock Generation



34.6.2.3.1 Synchronous Clock Operation

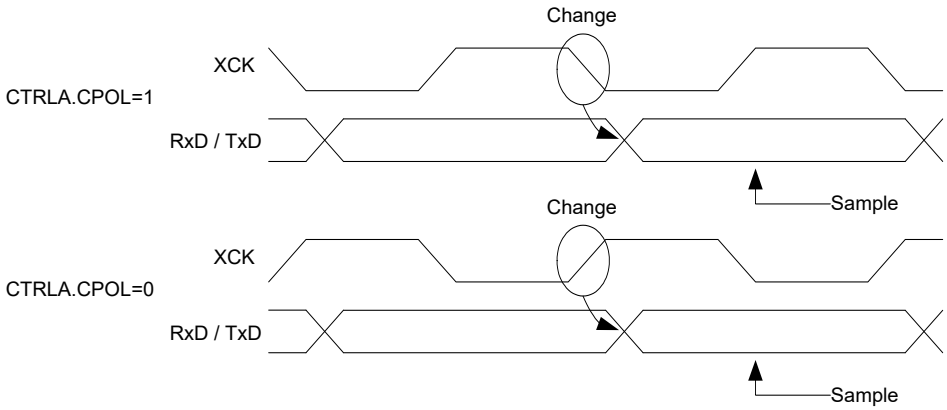
In synchronous mode, the CTRLA.MODE bit field determines whether the transmission clock line (XCK) serves either as input or output. The dependency between clock edges, data sampling, and data change is the same for internal and external clocks. Data input on the RxD pin is sampled at the opposite XCK 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 will be changed on the rising edge of XCK, and sampled on the falling edge of XCK.

When CTRLA.CPOL is '1', the data will be changed on the falling edge of XCK, and sampled on the rising edge of XCK.

Figure 34-4. 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.

34.6.2.4 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.

34.6.2.5 Data Transmission

Data transmission is initiated by writing the data to be sent into the Tx buffer by accessing the DATA register. Then, the data in Tx buffer 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(s) has been transmitted and the Tx buffer is empty, 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 at least FIFO threshold (CTRLC.TXTRHOLD) locations are empty and ready for new data. The DATA register should only be written to when INTFLAG.DRE is set.

34.6.2.5.1 Disabling the Transmitter

The transmitter is disabled by writing '0' to the Transmitter Enable bit in the CTRLB register (CTRLB.TXEN).

Disabling the transmitter will complete only after any ongoing and pending transmissions are completed, i.e., there is no data in the Transmit Shift register and Tx buffer to transmit.

34.6.2.6 Data Reception

The receiver accepts data when a valid Start bit is detected. Each bit following the Start bit will be 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 will be 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 will be moved into the receive buffer.

The Receive Complete Interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.RXC) will be set when the number of bytes present in the FIFO equals or is higher than the threshold value defined by the CTRLC.RXTRHOLD setting. An optional interrupt will be generated.

The received data can be read from the DATA register when the Receive Complete Interrupt flag is set.

34.6.2.6.1 Disabling the Receiver

Writing '0' to the Receiver Enable bit in the CTRLB register (CTRLB.RXEN) will disable the receiver, flush the two-level receive buffer, and data from ongoing receptions will be lost.

34.6.2.6.2 Error Bits

The USART receiver has three error bits in the Status (STATUS) register: Frame Error (FERR), Buffer Overflow (BUFOVF), and Parity Error (PERR). Once an error happens, the corresponding error bit will be 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. After the received data is read, STATUS.BUFOVF will be set along with INTFLAG.RXC.

34.6.2.6.3 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.

34.6.2.6.4 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 is depending 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 will not be able to synchronize the frames to the start bit.

There are two possible sources for a mismatch in baud rate: First, the reference clock will always have 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 should be set to give the lowest possible error. Refer to Clock Generation – Baud-Rate Generator for details.

Recommended maximum receiver baud-rate errors for various character sizes are shown in the table below.

Table 34-4. 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
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

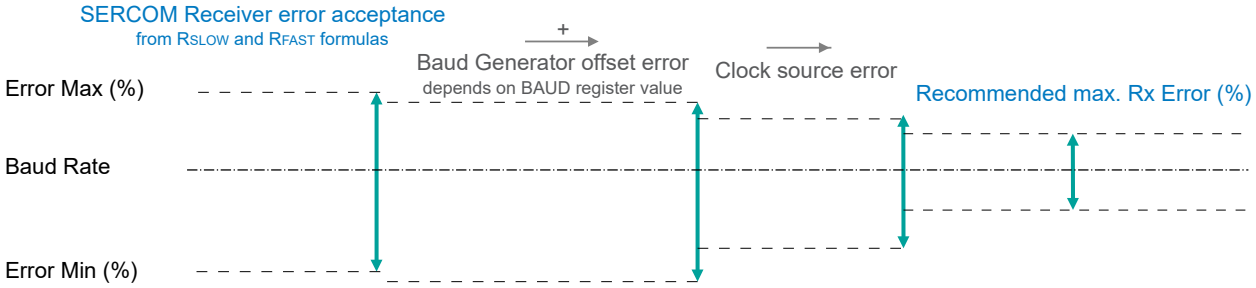
The following equations calculate the ratio of the incoming data rate and internal receiver baud rate:

$$R_{\mathrm{SLOW}} = \frac{(D+1)S}{S-1+D\cdot S+S_F} \quad , \qquad R_{\mathrm{FAST}} = \frac{(D+2)S}{(D+1)S+S_M}$$

- 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 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.

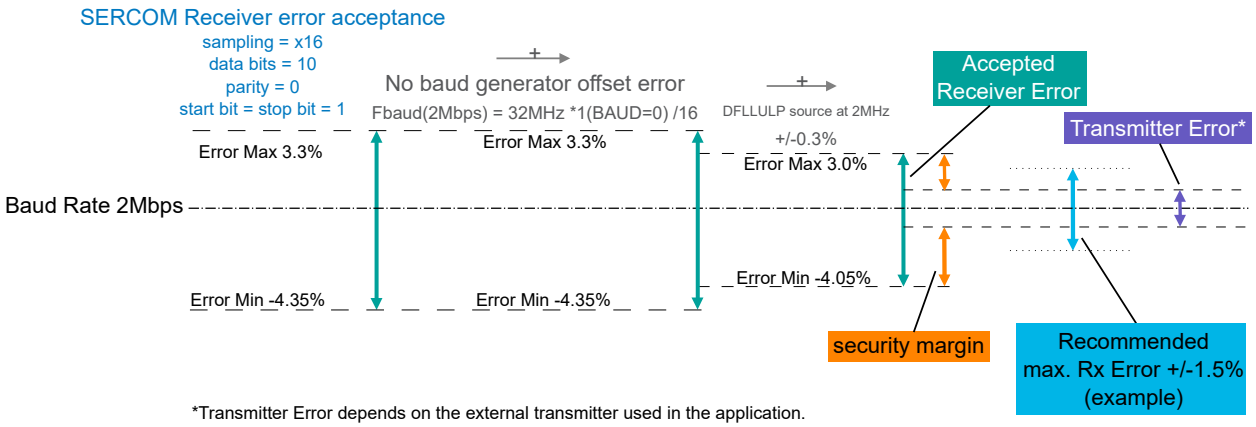
The recommended maximum Rx Error assumes that the receiver and transmitter equally divide the maximum total error. Its connection to the SERCOM Receiver error acceptance is depicted in this figure:

Figure 34-5. USART Rx Error Calculation



The recommendation values in the table above accommodate errors of the clock source and the baud generator. The following figure gives an example for a baud rate of 3Mbps:

Figure 34-6. USART Rx Error Calculation Example



*Transmitter Error depends on the external transmitter used in the application.
It is advised that it is within the Recommended max. Rx Error (+/-1.5% in this example).
Larger Transmitter Errors are acceptable but must lie within the Accepted Receiver Error.

34.6.3 Additional Features

34.6.3.1 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.

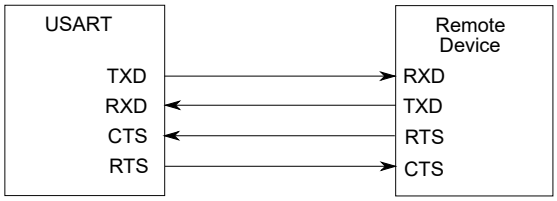
If *odd parity* is selected (CTRLB.PMODE=1), the parity bit of an outgoing frame is '1' if the data contains an even number of bits that are '0', making the total number of '1' odd.

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.

34.6.3.2 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 shown in the figure below.

Figure 34-7. Connection with a Remote Device for Hardware Handshaking

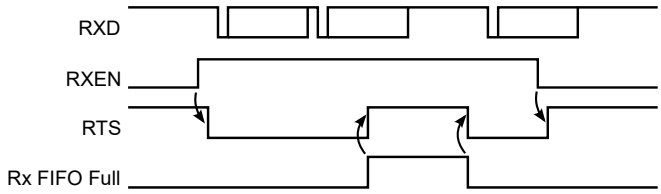


Hardware handshaking is only available in the following configuration:

- USART with internal clock (CTRLA.MODE=1),
- Asynchronous mode (CTRLA.CMODE=0),
- and Flow control pinout (CTRLA.TXPO=2).

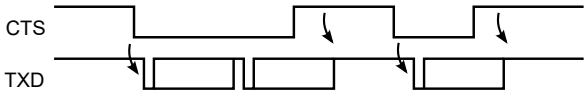
When the receiver is disabled or the receive FIFO is full, the receiver will drive the RTS pin high. This notifies the remote device to stop transfer after the ongoing transmission. Enabling and disabling the receiver by writing to CTRLB.RXEN will set/clear the RTS pin after a synchronization delay. When the receive FIFO goes full, RTS will be set immediately and the frame being received will be stored in the shift register until the receive FIFO is no longer full.

Figure 34-8. Receiver Behavior when Operating with Hardware Handshaking



The current CTS Status is in the STATUS register (STATUS.CTS). Character transmission will start only if STATUS.CTS=0. When CTS is set, the transmitter will complete the ongoing transmission and stop transmitting.

Figure 34-9. Transmitter Behavior when Operating with Hardware Handshaking



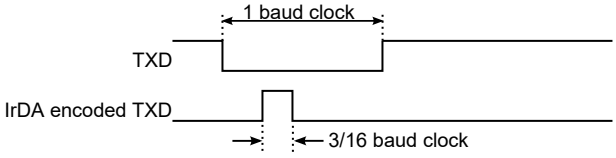
34.6.3.3 IrDA Modulation and Demodulation

Transmission and reception can be encoded IrDA compliant up to 115.2 kb/s. IrDA modulation and demodulation work in the following configuration:

- IrDA encoding enabled (CTRLB.ENC=1),
- Asynchronous mode (CTRLA.CMODE=0),
- and 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 figure below.

Figure 34-10. IrDA Transmit Encoding



The reception decoder has two main functions.

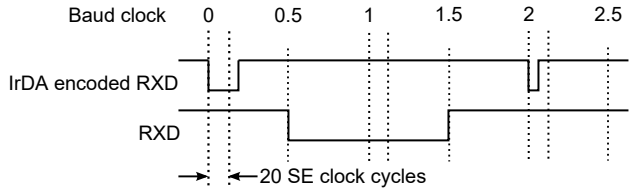
The first is to synchronize the incoming data to the IrDA baud rate counter. Synchronization is performed at the start of each zero pulse.

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.

Note: Note that 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 figure below illustrates reception where RXPL.RXPL is set to 19. This indicates that the pulse width should 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 since it does not meet the minimum requirement of 2/16 baud clock.

Figure 34-11. IrDA Receive Decoding



34.6.3.4 Break Character Detection and Auto-Baud/LIN Client

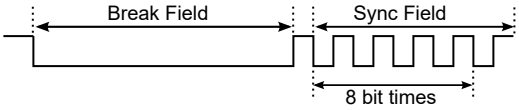
Break character detection and auto-baud are available in this configuration:

- Auto-baud frame format (CTRLA.FORM = 0x04 or 0x05),
- Asynchronous mode (CTRLA.CMODE = 0),
- and 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 has been 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 in order 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 34-12. 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 most significant bits of the counter (value divided by 8) give the new clock divider (BAUD.BAUD), and the 3 least significant bits of this value (the remainder) give the new Fractional Part (BAUD.FP).

When the Sync Field has been 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.

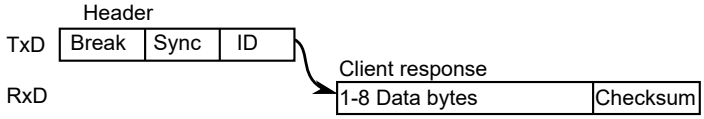
34.6.3.5 LIN Host

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)
- LSB is transmitted first (CTRLA.DORD = 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 will respond with 1-8 bytes of data plus checksum.

Figure 34-13. 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.

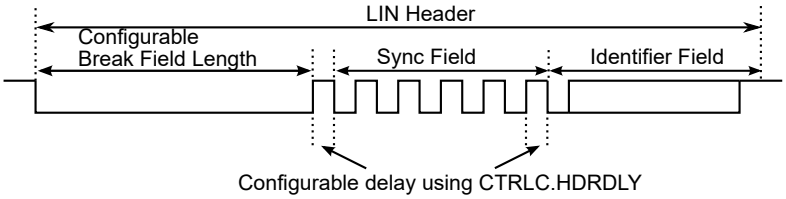
- CTRLB.LINCMD is written to 0x1.
- DATA register written to 0x00. This triggers transmission of the break field by hardware. Note that writing the DATA register with any other value will also result in the transmission of the break field by hardware.
- DATA register written to 0x55. The 0x55 value (sync) is transmitted.
- DATA register 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 written to the identifier. This triggers 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 34-14. LIN Header Generation



After header transmission is complete, the client responds with 1-8 data bytes plus checksum.

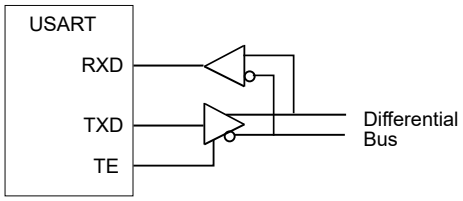
34.6.3.6 RS485

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 shown in the figure below. While operating in RS485 mode, the transmit enable pin (TE) is driven high when the transmitter is active.

Figure 34-15. RS485 Bus Connection



The TE pin will remain high for the complete frame including stop bit(s). If a Guard Time is programmed in the Control C register (CTRLC.GTIME), the line will remain driven after the last character completion. The following figure shows a transfer with one stop bit and CTRLC.GTIME=3.

Figure 34-16. Example of TE Drive with Guard Time

The Transmit Complete interrupt flag (INTFLAG.TXC) will be raised after the guard time is complete and TE goes low.

34.6.3.7 ISO 7816 for Smart Card Interfacing

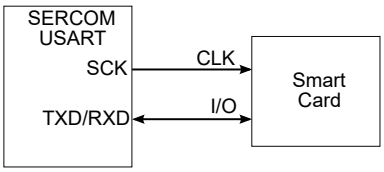
The SERCOM USART features an ISO/IEC 7816-compatible operating mode. This mode permits interfacing with smart cards and Security Access Modules (SAM) communicating through an ISO 7816 link. Both T=0 and T=1 protocols defined by the ISO 7816 specification are supported.

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 as the host of the communication as it generates the clock.

Figure 34-17. 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

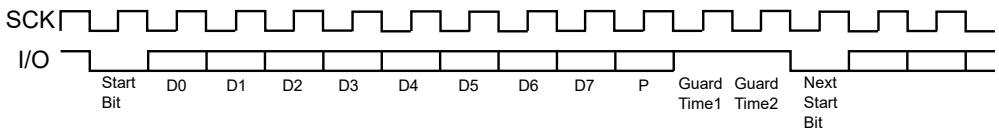
In T=0 protocol, a character is made up of:

- one start bit,
- eight data bits,
- one parity bit
- and 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).

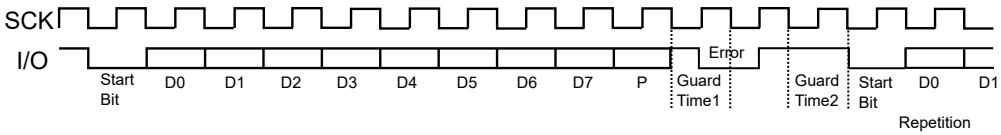
If no parity error is detected, the I/O line remains during the guard time and the transmitter can continue with the transmission of the next character, as shown in the figure below.

Figure 34-18. 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 shown in the next 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 34-19. 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 as 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.

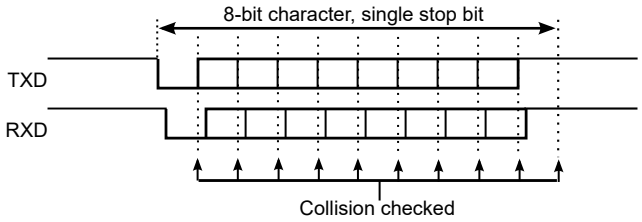
Parity is generated when transmitting and checked when receiving. Parity error detection sets the STATUS.PERR bit, and the erroneous character is written to the receive FIFO. When using T=1 protocol, the receiver does not signal errors on the I/O line and the transmitter does not retransmit.

34.6.3.8 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).

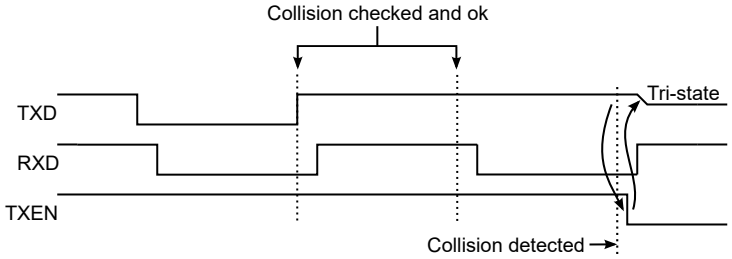
Collision detection is performed for each bit transmitted by comparing the received value with the transmit value, as shown in the figure below. While the transmitter is idle (no transmission in progress), characters can be received on RxD without triggering a collision.

Figure 34-20. Collision Checking



The next figure shows 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 34-21. 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) will be set until this is complete.
   - After disabling, the TxD pin will be 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), since the transmit buffer no longer contains data.

After a collision, software must manually enable the transmitter again before continuing, after assuring that the CTRLB Synchronization Busy bit (SYNCBUSY.CTRLB) is not set.

34.6.3.9 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.

34.6.3.10 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, an internal fast start-up oscillator must be selected as the GCLK_SERCOMx_CORE source. If the fast startup oscillator is not selected, there may be corruption until the oscillator is stable.

When a 1-to-0 transition is detected on RxD, an internal fast start-up 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. 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.

34.6.3.11 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 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.

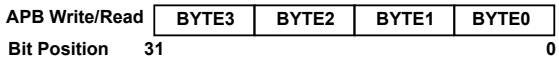
34.6.3.12 32-bit Extension

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 data transfer begins, LENGTH.LEN must be enabled only when CTRLC.DATA32B is enabled.

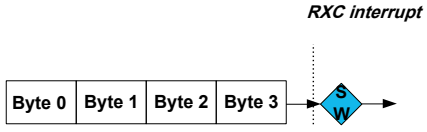
The figure below shows the order of transmit and receive when using 32-bit extension. Bytes are transmitted or received, and stored in order from 0 to 3. Only 8-bit and smaller character sizes are supported. If the character size is less than 8 bits, characters will still be 8-bit aligned within the 32-bit APB write or read. The unused bits within each byte will be zero for received data and unused for transmit data.

Figure 34-22. 32-bit Extension Ordering



A receive transaction using 32-bit extension is in the next 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 34-23. 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 will be 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 will produce 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 will produce unpredictable results.

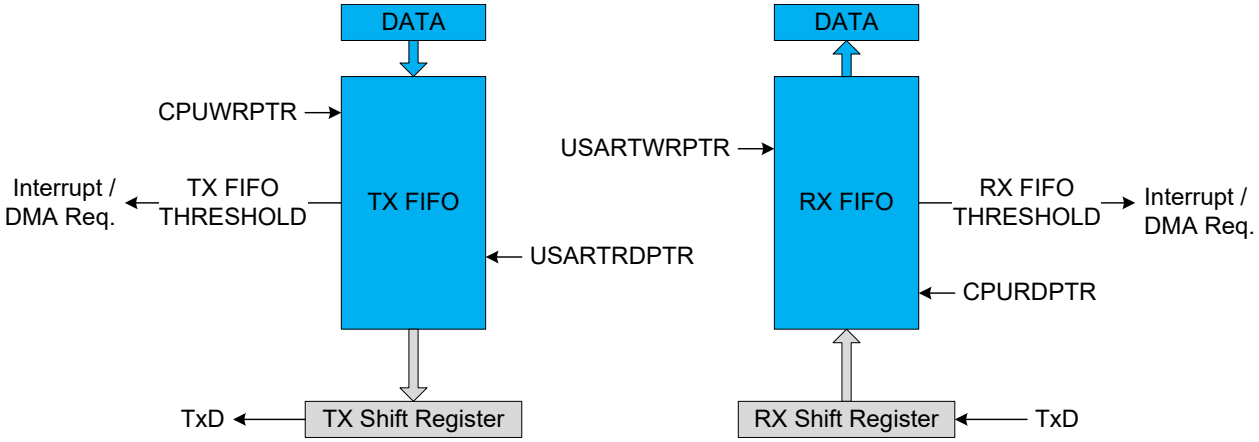
34.6.3.13 FIFO Operation

The USART embeds up to 16-bytes FIFO capability. The receive / transmit buffer is considered to have the FIFO mode enabled when the FIFOEN bit in CTRLC register is set to a ‘1’ (CTRLC.FIFOEN = 1). By default, the FIFO can act as a 16-by-8-bit array, or as a 4-by-32-bit array, depending on the setting of the CTRLC.DATA32B bit.

The hardware around this array implements four pointers, called the CPU Write Pointer (CPUWRPTR), the CPU Read Pointer (CPURDPTR), the USART Write pointer (USARTWRPTR) and the USART Read pointer (USARTRDPTR). All of these pointers reset to ‘0’. The CPUWRPTR and CPURDPTR pointers are native to the CPU clock domain, while the USARTWRPTR and USARTRDPTR are native to the USART domain. The location pointed to by the CPUWRPTR is the current TX FIFO. The location pointed to by the CPURDPTR becomes the

current RX FIFO. Writes to DATA register by the CPU will point to TX FIFO. Reads to DATA register by the CPU will point to RX FIFO. The location pointed to by the USARTWRPTR / USARTRDPTR is logically the current RX/TX shift registers.

Figure 34-24. FIFO Overview



The interrupts and DMA triggers are generated according to FIFO threshold settings in Control C register (CTRLC.TXTRHOLD, CTRLC.RXTRHOLD).

The Data Register Empty interrupt flag, and the DMA TX trigger respectivly, are generated when the available place in the TX FIFO is equal or higher than the threshold value defined by the CTRLC.TXTRHOLD settings. The Transfer complete interrupt is generated when the TX FIFO is empty and the entire data (including the stop bits) has been transmitted.

The Receive Complete interrupt flag, and the DMA RX trigger respectivly, are generated when the number of bytes present in the RX FIFO equals or is higher than the threshold value defined by the CTRLC.RXTRHOLD settings. The ERROR interrupt flag is generated when both RX shifter and the RX FIFO are full.

The FIFO is fully accessible if the SERCOM is halted, by writting the corresponding CPU FIFO pointer in the FIFOPTR register. The RX or TX FIFO can be individually cleared, by setting the respective FIFO Clear bit in the Control B register (CTRLB.FIFOCLR). The FIFO Clear must be written before data transfer begins. Writing the FIFO Clear bits while a frame is in progress will produce unpredicatable results.

34.6.3.13.1 Pointer Operation when DATA Transmission

As in normal operation, data transmission is initiated by writing the data to be sent into the TX FIFO, by accessing the DATA register. CPUWRPTR is incremented by 1 every time the CPU writes a word to the memory array. Then, the data in TX FIFO will be moved to the shift register when the shift register is empty and ready to send a new frame, and the USARTRDPTR is incremented by 1. After the shift register is loaded with data, the data frame will be transmitted.

As long as data are present in TX FIFO (FIFOSPACE.TXSPACE != 0), a new data will be automatically loaded in the TX shift register when the previous data transmission is completed. All pointers increment to their maximum value, dictated by CTRLC.DATA32B bit, and then rolls over to ‘0’.

Depending the TX FIFO Threshold settings (CTRLC.TXTRHOLD), Interrupt Flag Status and Clear register (INTFLAG.DRE) indicates that the register is empty and ready for new data.

If the USART is halted when debugging, the CPUWRPTR pointer can be accessed by writing the CPUWRPTR bits in FIFOPTR register (FIFOPTR.CPUWRPTR). These bits will not increment if a new data is written into the TX FIFO memory.

34.6.3.13.2 Pointer Operation when DATA Reception

As in normal operation, 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 will be moved into the RX FIFO, and the USARTWRPTR is incremented by one. Depending the RX FIFO Threshold settings (CTRLC.RXTRHOLD), the Receive Complete interrupt flag (INTFLAG.RXC) is set, and the DATA can be read from RX FIFO. When a DATA is read, the CPURDPTR is incremented. As long as data are present in RX FIFO (FIFOSPACE.RXSPACE != 0), the CPU can read these data

by accessing the DATA register. All pointers increment to their maximum value, dictated by CTRLC.DATA32B bit, and then rolls over to '0'.

When both RX shifter and RX FIFO are full, the Buffer Overflow status bit is set (STATUS.BUFOVF) and optional ERROR interrupt is generated. The data will not be stored while BUFOVF is '1', effectively disabling the module until software reads RX FIFO.

If the USART is halted when debugging, the RX FIFO CPU read pointer can be accessed by writting the CPURDPTR bits in FIFOPTR register (FIFOPTR.CPURDPTR). These bits will not increment if a new data is read from the RX FIFO memory.

34.6.4 DMA, Interrupts and Events

Table 34-5. 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	

34.6.4.1 DMA Operation

The USART generates the following DMA requests:

- Data received (RX): The request is set when data is available in the receive FIFO 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.

34.6.4.2 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) 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 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, refer to the INTFLAG register description.

The value of INTFLAG indicates which interrupt is executed. Note that interrupts must be globally enabled for interrupt requests. Refer to Nested Vector Interrupt Controller for details.

34.6.5 Sleep Mode Operation

The behavior in sleep mode is depending 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: Internal clock will be disabled, after any ongoing transfer was completed. The Receive Complete interrupt(s) can wake up the device.
- External clocking, CTRLA.RUNSTDBY=0: External clock will be disconnected, after any ongoing transfer was completed. All reception will be dropped.

34.6.6 Synchronization

Some registers (or bit fields within a register) require synchronization when read and/or written.

Synchronization is denoted by the "Read-Synchronized" (or "Read-Synchronized Bits”) and/or "Write-Synchronized" (or "Write-Synchronized Bits”) property in each individual register description.

For more details, refer to Register Synchronization.

34.6.7 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. Refer to the Debug Control (DBGCTRL) register for details.

34.7 Register Summary

Refer to the Registers Description section for more details on register properties and access permissions.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRLA	31:24		DORD	CPOL	CMODE	FORM[3:0]			
		23:16	SAMPA[1:0]		RXPO[1:0]				TXPO[1:0]	
		15:8	SAMPR[2:0]					RXINV	TXINV	IBON
		7:0	RUNSTDBY			MODE[2:0]			ENABLE	SWRST
0x04	CTRLB	31:24							LINCMD[1:0]	
		23:16	FIFOCLR[1:0]						RXEN	TXEN
		15:8			PMODE			ENC	SFDE	COLDEN
		7:0		SBMODE				CHSIZE[2:0]		
0x08	CTRLC	31:24	TXTRHOLD[1:0]		RXTRHOLD[1:0]		FIFOEN		DATA32B[1:0]	
		23:16		MAXITER[2:0]					DSNACK	INACK
		15:8					HDRDLY[1:0]		BRKLEN[1:0]	
		7:0						GTIME[2:0]		
0x0C	BAUD	15:8	BAUD[15:8]							
		7:0	BAUD[7:0]							
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	15:8								
		7:0	ITER	TXE	COLL	ISF	CTS	BUFOVF	FERR	PERR
0x1C	SYNCBUSY	31:24								
		23:16								
		15:8								
		7:0				LENGTH	RXERRCNT	CTRLB	ENABLE	SWRST
0x20	RXERRCNT	7:0	RXERRCNT[7:0]							
0x21	Reserved									
0x22	LENGTH	15:8							LENEN[1:0]	
		7:0	LEN[7:0]							
0x24 ... 0x27	Reserved									
0x28	DATA	31:24	DATA[31:24]							
		23:16	DATA[23:16]							
		15:8	DATA[15:8]							
		7:0	DATA[7:0]							
0x2C ... 0x2F	Reserved									
0x30	DBGCTRL	7:0								DBGSTOP
0x31 ... 0x33	Reserved									
0x34	FIFOSPACE	15:8				RXSPACE[4:0]				
		7:0				TXSPACE[4:0]				
0x36	FIFOPTR	15:8					CPURDPTR[3:0]			
		7:0					CPUWRPTR[3:0]			

34.7.1 Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected Bits, Write-Synchronized Bits

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.
Note: This bit is enable-protected. 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 synchronous mode.
Note: This bit is enable-protected. This bit is not synchronized.

Value	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.
Note: This bit is enable-protected. 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.
Note: This bit field is enable-protected. This bit field is not synchronized.

Value	Description
0x0	USART frame

.....continued	
Value	**Description**
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.

Note: This bit field is enable-protected. This bit field is not synchronized.

Value	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.

Note: This bit field is enable-protected. This bit field is not synchronized.

Value	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.

Note: This bit field is enable-protected. This bit field is not synchronized.

Value	TxD Pin Location	XCK Pin Location (When Applicable)	RTS/TE	CTS
0x0	SERCOM PAD[0]	SERCOM PAD[1]	N/A	N/A
0x1	Reserved			
0x2	SERCOM PAD[0]	N/A	SERCOM PAD[2]	SERCOM PAD[3]
0x3	SERCOM_PAD[0]	SERCOM_PAD[1]	SERCOM_PAD[2]	N/A

Bits 15:13 – SAMPR[2:0] Sample Rate

These bits select the sample rate.

Note: This bit field is enable-protected. This bit is not synchronized.

Value	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.

Note: This bit is enable-protected. 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.

Note: This bit is enable-protected. 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.

Note: This bit is enable-protected. 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.

Note: This bit is enable-protected. This bit is not synchronized.

Value	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.

Note: This bit is enable-protected. This bit is not synchronized.

Value	Description
0x0	USART with external clock
0x1	USART with internal clock

Bit 1 – ENABLE Enable

Notes:

1. This bit is write-synchronized: SYNCBUSY.ENABLE must be checked to ensure the CTRLA.ENABLE synchronization is complete.
2. 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 will be disabled.

Writing '1' to CTRLA.SWRST will always take precedence, meaning that all other writes in the same write-operation will be discarded. Any register write access during the ongoing reset will result in an APB error. Reading any register will return the reset value of the register.

Notes:

1. When the CTRLA.SWRST is written, the user must poll SYNCBUSY.SWRST bit to know when the reset operation is complete.
2. During a SWRST, access to registers/bits without SWRST are disallowed until SYNCBUSY.SWRST is cleared by hardware.
3. This bit is not enable protected.

Value	Description
0	There is no reset operation ongoing.
1	The reset operation is ongoing.

34.7.2 Control B

Name: CTRLB
Offset: 0x04
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected Bits, Write-Synchronized Bits

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
	FIFOCLR[1:0]						RXEN	TXEN
Access	R/W	R/W					R/W	R/W
Reset	0	0					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 will always read back as zero.
Note: This bit field is write-synchronized: SYNCBUSY.CTRLB must be checked to ensure the CTRLB.LINCMD synchronization is complete.

Note: This bit field is 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

Bits 23:22 – FIFOCLR[1:0] FIFO Clear
When these bits are set, the corresponding FIFO will be cleared. The bits will automatically clear when SYNCBUSY.CTRLB = 0.
Note: This bit field is write-synchronized: SYNCBUSY.CTRLB must be checked to ensure the CTRLB.FIFOCLR synchronization is complete.

Note: This bit field is not enable-protected.

Value	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 will disable the USART receiver. Disabling the receiver will flush the receive buffer and clear the FERR, PERR and BUFOVF bits in the STATUS register.
Writing '1' to this bit when the USART is disabled will set CTRLB.RXEN immediately. When the USART is enabled, CTRLB.RXEN is cleared and the receiver enable is only effective at the end of the SYNCBUSY.CTRLB synchronization.
Writing '1' to this bit when the USART is enabled requires to wait the end of the SYNCBUSY.CTRLB synchronization to ensure the receiver is enabled.
Note: This bit is write-synchronized: SYNCBUSY.CTRLB must be checked to ensure the CTRLB.RXEN synchronization is complete.

Note: This bit is not enable-protected.

Value	Description
0	The receiver is disabled or being enabled.
1	The receiver is enabled or will be enabled when the USART is enabled.

Bit 16 – TXEN Transmitter Enable

Writing '0' to this bit will disable the USART transmitter. Disabling the transmitter will not become effective until ongoing and pending transmissions are completed.
Writing '1' to this bit when the USART is disabled will set CTRLB.TXEN immediately. When the USART is enabled, CTRLB.TXEN is cleared and the transmitter enable is only effective at the end of the SYNCBUSY.CTRLB synchronization.
Writing '1' to this bit when the USART is enabled requires to wait the end of the SYNCBUSY.CTRLB synchronization to ensure the transmitter is enabled.
Note: This bit is write-synchronized: SYNCBUSY.CTRLB must be checked to ensure the CTRLB.TXEN synchronization is complete.

Note: This bit is not enable-protected.

Value	Description
0	The transmitter is disabled or being enabled.
1	The transmitter is enabled or will be 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 will automatically generate and send the parity of the transmitted data bits within each frame. The receiver will generate a parity value for the incoming data and parity bit, compare it to the parity mode and, if a mismatch is detected, STATUS.PERR will be set.
Note: This bit is enable-protected. 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.
Note: This bit is enable-protected. 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 will wake up the device when a start bit is detected on the RxD line.
Note: This bit is enable-protected. This bit is not synchronized.

Value	INTENSET.RXS	INTENSET.RXC	Description
0	X	X	Start-of-frame detection disabled.
1	0	0	Reserved

.....continued			
Value	**INTENSET.RXS**	**INTENSET.RXC**	**Description**
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.

Note: This bit is enable-protected. 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.

Note: This bit is enable-protected. 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.

Note: This bit field is enable-protected. This bit field is not synchronized.

Value	Description
0x0	8 bits
0x1	9 bits
0x2-0x4	Reserved
0x5	5 bits
0x6	6 bits
0x7	7 bits

34.7.3 Control C

Name: CTRLC
Offset: 0x08
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
	TXTRHOLD[1:0]		RXTRHOLD[1:0]		FIFOEN		DATA32B[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	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 31:30 – TXTRHOLD[1:0] Transmit FIFO Threshold
These bits define the threshold for generating the Data Register Empty interrupt and DMA TX trigger.

Value	Name	Description
0	DEFAULT	Interrupt and DMA triggers can be generated as long as the FIFO is not full.
1	HALF	Interrupt and DMA triggers are generated when half FIFO space is free.
2	EMPTY	Interrupt and DMA triggers are generated when the FIFO is empty.
3	-	Reserved

Bits 29:28 – RXTRHOLD[1:0] Receive FIFO Threshold
These bits define the threshold for generating the RX Complete interrupt and DMA RX trigger.

Value	Name	Description
0	DEFAULT	Interrupt and DMA triggers can be generated when a DATA is present in the FIFO.
1	HALF	Interrupt and DMA triggers can be generated only when the FIFO is half-full.
2	FULL	Interrupt and DMA triggers can be generated only when the FIFO is full.
3	-	Reserved

Bit 27 – FIFOEN FIFO Enable
This bit enables the FIFO operation.

Value	Description
0	FIFO operation is disabled
1	FIFO operation is enabled

Bits 25:24 – DATA32B[1:0] Data 32 Bit
These bits configure 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.

Value	Description
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).

Bit 17 – DSNACK Disable Successive Not Acknowledge
This bit controls how many times NACK will be 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 4 bit time.
0x3	Delay between break and sync transmission is 14 bit time. Delay between sync and ID transmission is 4 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.

34.7.4 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[0]=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, as described in the 33.6.2.3. Clock Generation – Baud-Rate Generator section.

- **Bits 12:0 - BAUD[12:0]: Baud Value**

  These bits control the clock generation, as described in the 33.6.2.3. Clock Generation – Baud-Rate Generator section.

34.7.5 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}.

$$PULSE \geq \left(\mathrm{RXPL} + 1\right) \cdot SE_{per}$$

34.7.6 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 will also be 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 will clear 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 will clear 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 will clear 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 will clear 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 will clear 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 will clear the Transmit Complete Interrupt Enable bit, which disables the Receive 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 will clear 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.

34.7.7 Interrupt Enable Set

Name: INTENSET
Offset: 0x16
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 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 will set 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 will set 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 will set 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 will set 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 will set 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 will set 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 will set 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.

34.7.8 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 will set this flag have corresponding status flags in the STATUS register. Errors that will set this flag are COLL, ISF, BUFOVF, FERR, and PERR.Writing '0' to this bit has no effect.
Writing '1' to this bit will clear 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 will clear the flag.

Bit 4 – CTSIC Clear to Send Input Change
This flag is cleared by writing a '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 will clear 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 will clear 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 when the entire frame in the transmit shift register has been shifted out and there are no new data in DATA.
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear 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.

34.7.9 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 ISO7816 T=0 mode.
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear 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 zero.
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear 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 will clear 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 will clear 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 will indicate 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 will clear it.

Bit 1 – FERR Frame Error

Reading this bit before reading the Data register will indicate 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, i.e., when the first stop bit is zero.
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear it.

Bit 0 – PERR Parity Error

Reading this bit before reading the Data register will indicate 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 will clear it.

34.7.10 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 will be set until synchronization is complete. If the LENGTH register is written to 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 will be 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 bit will be set until synchronization is complete. If CTRLB is written while SYNCBUSY.CTRLB is asserted, an APB error will be 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 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.

Value	Description
0	SWRST synchronization is not busy.
1	SWRST synchronization is busy.

34.7.11 Receive Error Count

Name: RXERRCNT
Offset: 0x20
Reset: 0x00
Property: Read-Synchronized

Note: This register is read-synchronized: SYNCBUSY.RXERRCNT must be checked to ensure the RXERRCNT register synchronization is complete.

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.

34.7.12 Length

Name: LENGTH
Offset: 0x22
Reset: 0x00
Property: Write-Synchronized

Note: This register is write-synchronized: SYNCBUSY.LENGTH must be checked to ensure the LENGTH register synchronization is complete.

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

34.7.13 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 will return the contents of the Receive Data register. The register should 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 should be read before reading the DATA value in order to get any corresponding error.
Writing these bits will write the Transmit Data register. This register should 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.

34.7.14 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.

34.7.15 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).

34.7.16 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.

35. Serial Peripheral Interface (SERCOM SPI)

35.1 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 shown in 35.3. Block Diagram. Each side, host and client, depicts a separate SPI containing a shift register, a transmit buffer and a receive buffer.

The transmitter consists of a 8-bytes write FIFO buffer, a Shift register, and control logic for different frame formats. The write buffer support data transmission without any delay between frames.

The receiver consists of a 8-bytes receive FIFO 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.

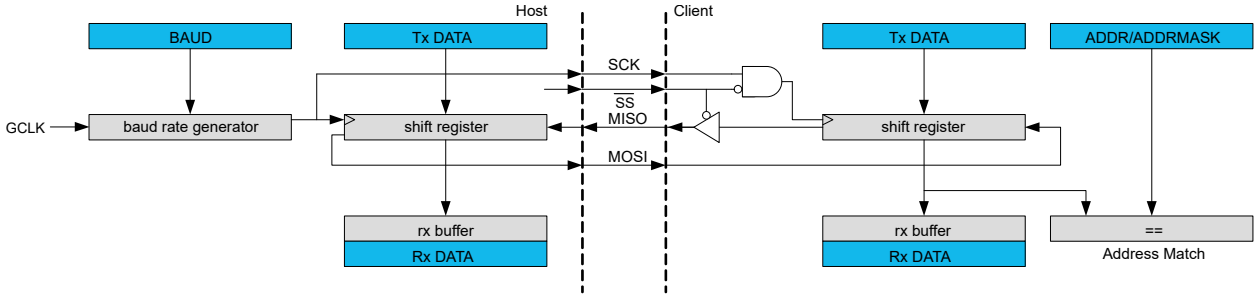
In addition, the SPI host uses the SERCOM baud-rate generator, while the SPI client can use the SERCOM address match logic. Labels in capital letters are synchronous to CLK_SERCOMx_APB and accessible by the CPU, while labels in lowercase letters are synchronous to the SCK clock.

35.2 Features

- Full-duplex, four-wire interface (MISO, MOSI, SCK, $\overline{SS}$)
- Receive buffer: 8-bytes FIFO
- Transmit buffer: 8-bytes FIFO
- Supports all four SPI modes of operation
- Single data direction operation allows alternate function on MISO or MOSI pin
- Selectable LSB-first data transfer or MSB-first data transfer
- DMA operations supported
- 32-bit Extension for better system bus utilization
- Host operation:
  - Serial clock speed, $f_{SCK}=1/t_{SCK}$(1)
  - 8-bit clock generator
  - Hardware controlled $\overline{SS}$
- Client operation:
  - Serial clock speed, $f_{SCK} = 1/t_{SSCK}$(1)
  - Optional 8-bit address match operation
  - Operation in all sleep modes
  - Wake on $\overline{SS}$ transition

35.3 Block Diagram

Figure 35-1. Full-Duplex SPI Host Client Interconnection



35.4 Signal Description

Table 35-1. SERCOM SPI Signals

Signal Name	Type	Description
PAD[3:0]	Digital I/O	General SERCOM pins

⚠ CAUTION I/Os for SERCOM peripherals are grouped into I/O sets, listed in their 'IOSET' column in the pinout tables. For these peripherals, it is mandatory to use I/Os that belong to the same I/O set. The timings are not guaranteed when I/Os from different I/O sets are mixed. Refer to the Pinout and Packaging chapter to get IOSET definitions.

35.5 Peripheral Dependencies

Table 35-2. SERCOM Configuration Summary

Peripheral name	Base address	NVIC IRQ Index	MCLK AHB/APB Clocks	GCLK Peripheral Channel Index (GCLK.PCHCTRL)	PAC Peripheral Identifier Index (PAC.WRCTRL)	DMA Trigger Source Index (DMAC.CHCTRLB)	Power Domain (PM.STDBYCFG)
SERCOM0	0x42000400	31: bit 0 32: bit 1 33: bit 2 34: bit 3-7	CLK_SERCOM0_APB	17: GCLK_SERCOM0_CORE	65	4: RX 5: TX	PDSW
SERCOM1	0x42000800	35: bit 0 36: bit 1 37: bit 2 38: bit 3-7	CLK_SERCOM1_APB	18: GCLK_SERCOM1_CORE	66	6: RX 7: TX	PDSW
SERCOM2	0x42000C00	39: bit 0 40: bit 1 41: bit 2 42: bit 3-7	CLK_SERCOM2_APB	19: GCLK_SERCOM2_CORE	67	8: RX 9: TX	PDSW
SERCOM3	0x42001000	43: bit 0 44: bit 1 45: bit 2 46: bit 3-7	CLK_SERCOM3_APB	20: GCLK_SERCOM3_CORE	68	10: RX 11: TX	PDSW
SERCOM4	0x42001400	47: bit 0 48: bit 1 49: bit 2 50: bit 3-7	CLK_SERCOM4_APB	21: GCLK_SERCOM4_CORE	69	12: RX 13: TX	PDSW
SERCOM5	0x42001800	51: bit 0 52: bit 1 53: bit 2 54: bit 3-7	CLK_SERCOM5_APB	22: GCLK_SERCOM5_CORE	70	14: RX 15: TX	PDSW

35.6 Functional Description

35.6.1 Principle of Operation

The SPI is a high-speed synchronous data transfer interface It allows high-speed communication between the device and peripheral devices.

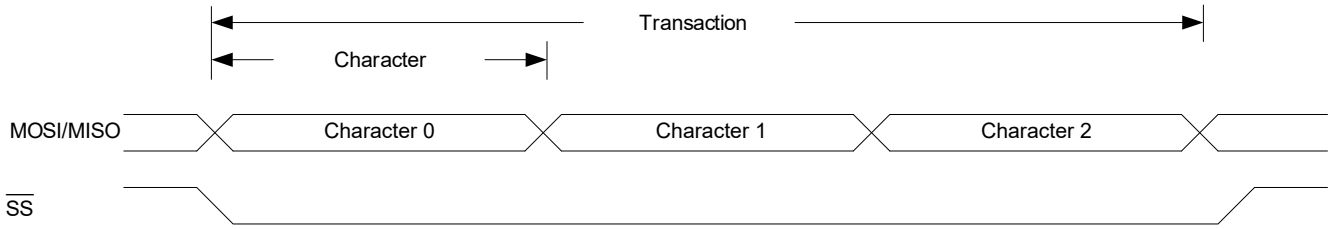
The SPI can operate as host or client. As host, the SPI initiates and controls all data transactions.

When transmitting data, the Tx Buffer 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 35-2. SPI Transaction Format



The SPI host must pull the SPI Select line ($\overline{SS}$) of the desired client low to initiate a transaction. 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 are 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

When the SERCOM is configured for SPI operation, the SERCOM controls the direction and value of the I/O pins according to the table below.

PORT Control bit PINCFGn.DRVSTR is still effective for the SERCOM output pins.

PORT Control bit PINCFGn.PULLEN is still effective on the SERCOM input pins, but is limited to the enabling/disabling of a pull down only (it is not possible to enable/disable a pull up).

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 35-3. SPI Pin Configuration

Pin	Host SPI	Client SPI
MOSI	Output	Input
MISO	Input	Output
SCK	Output	Input

The combined configuration of PORT, 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 above.

35.6.2 Basic Operation

35.6.2.1 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 will be discarded.

When the SPI is being disabled, writing to these registers will be 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 CTRLB register (CTRLB.MSSEN).
9. Enable the receiver by writing the Receiver Enable bit in the CTRLB register (CTRLB.RXEN=1).

35.6.2.2 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) will reset all registers of this peripheral to their initial states, except the DBGCTRL register, and the peripheral is disabled.

Refer to the CTRLA register description for details.

35.6.2.3 Clock Generation

In SPI host operation (CTRLA.MODE=0x3), the serial clock (SCK) is generated internally by the SERCOM baud-rate generator.

In SPI mode, the baud-rate generator is set to synchronous mode. The 8-bit Baud register (BAUD) value is used for generating SCK and clocking the shift register. Refer to Clock Generation – Baud-Rate Generator for more details.

In SPI client operation (CTRLA.MODE is 0x2), the clock is provided by an external host on the SCK pin. This clock is used to directly clock the SPI shift register.

35.6.2.4 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.

35.6.2.5 SPI Transfer Modes

There are four combinations of SCK phase and polarity to transfer serial data. The SPI data transfer modes are shown in SPI Transfer Modes (Table) and SPI Transfer Modes (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.

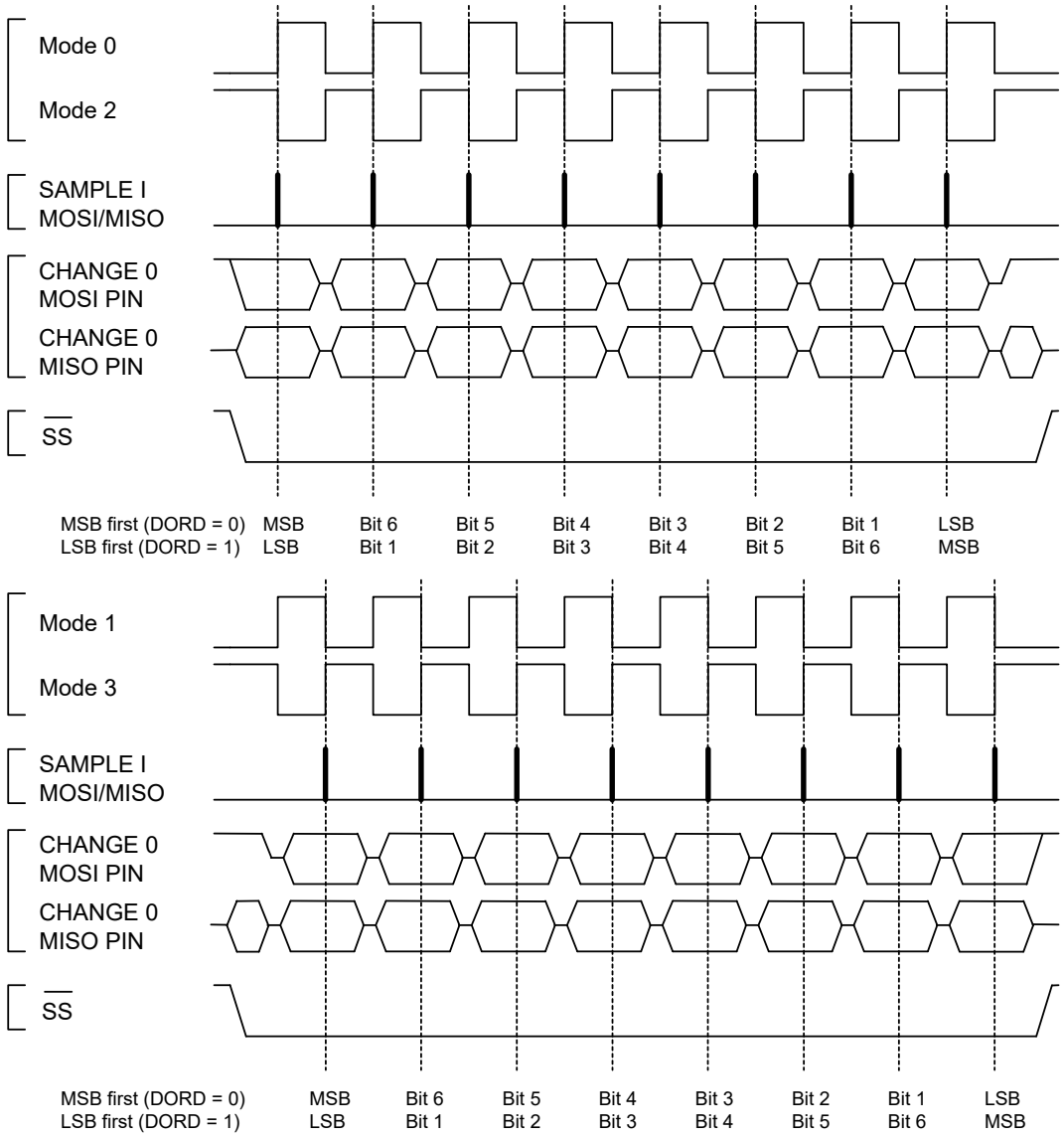
Table 35-4. SPI Transfer Modes

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

Note:
Leading edge is the first clock edge in a clock cycle.

Trailing edge is the second clock edge in a clock cycle.

Figure 35-3. SPI Transfer Modes



35.6.2.6 Transferring Data

35.6.2.6.1 Host

In host mode (CTRLA.MODE=0x3), when Host SPI Enable Select (CTRLB.MSSEN) is '1', hardware will control the $\overline{SS}$ line.

When the 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 will be transferred to the Shift register. Once the content of TxDATA has been transferred to the Shift register, the Data Register Empty flag in the Interrupt Flag Status and Clear register (INTFLAG.DRE) will be set. And a new character can be written to DATA. If the FIFO is enabled, INTFLAG.DRE will be set when at least FIFO threshold (CTRLC.TXTRHOLD) locations in TX FIFO are empty. Characters can be written as long as the FIFO is not full.

Each time one character is shifted out from the host, another character will be shifted in from the client simultaneously. If the receiver is enabled (CTRLA.RXEN=1), the contents of the shift register will be 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) will be set. The received data can be retrieved by reading DATA.

When the last character has been transmitted and there is no valid data in DATA, the Transmit Complete Interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.TXC) will be set. TXC is set when the transmit is complete and the TX FIFO is empty. 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.

35.6.2.6.2 Client

In Client mode (CTRLA.MODE=0x2), the SPI interface will remain inactive with the MISO line tri-stated as long as the $\overline{SS}$ pin is pulled high. Software may update the contents of Tx Buffer 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.

If FIFO is disabled, when the content of DATA has been loaded into the Shift register, INTFLAG.DRE will be set, and new data can be written to DATA. If the FIFO is enabled, INTFLAG.DRE will be set when at least FIFO threshold (CTRLC.TXTRHOLD) locations in TX FIFO are empty. When DATA or FIFO threshold locations are empty respectively, it takes three CLK_SERCOM_APB cycles for INTFLAG.DRE to be set.

Similar to the host, the client will receive one character for each character transmitted. A character will be transferred into the receive buffer within the same clock cycle its last data bit is received. The received character can be retrieved from DATA when the Receive Complete interrupt flag (INTFLAG.RXC) is set. If FIFO is disabled, the RXC is set when a character reception is complete. If the FIFO is enabled, the RXC is set when at least RX threshold (CTRLC.RXTRHOLD) data are received.

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) will be 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. Refer to Preloading the Client Shift Register.

When transmitting several characters in one SPI transaction, the data has to be written into 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 will be transmitted.

35.6.2.7 Receiver Error Bit

The SPI receiver has one error bit: the Buffer Overflow bit (BUFOVF), which can be read from the Status register (STATUS). Once an error happens, the bit will stay 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 is read, STATUS.BUFOVF and INTFLAG.ERROR will be set along with INTFLAG.RXC, and RxDATA will be zero.

35.6.3 Additional Features

35.6.3.1 Address Recognition

When the SPI is configured for client operation (CTRLA.MODE=0x2) with address recognition (CTRLA.FORM is 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 in order 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) in order to use this mode.

For further information, refer to 33.6.3.2. Secure Pin Multiplexing (PIC32CM LS00 only).

35.6.3.2 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.

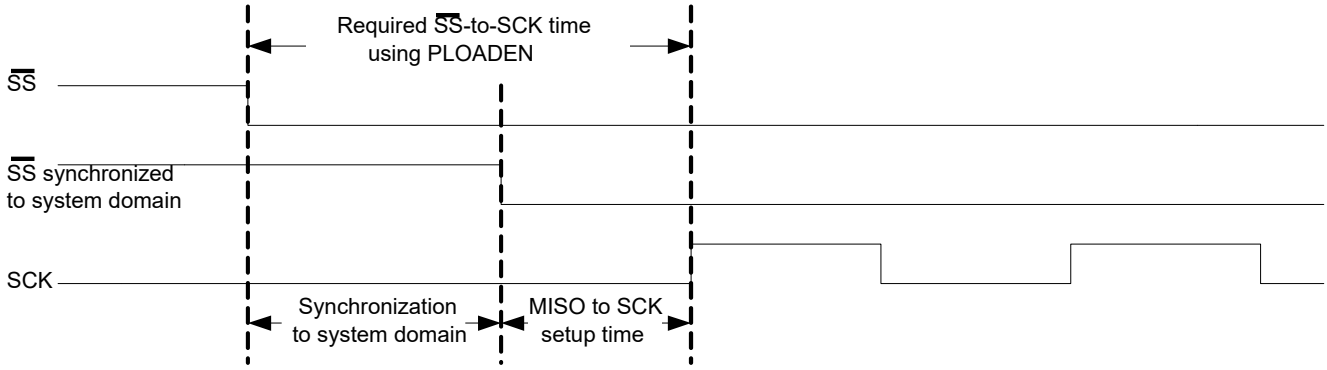
Only one data character will be preloaded into the shift register while the synchronized $\overline{SS}$ signal is high. If the next character is written to DATA before $\overline{SS}$ is pulled low, the second character will be stored in DATA until transfer begins.

For proper preloading, sufficient time must elapse between $\overline{SS}$ going low and the first SCK sampling edge, as in Timing Using Preloading.

Preloading is enabled by writing '1' to the Client Data Preload Enable bit in the CTRLB register (CTRLB.PLOADEN).

Note: In that mode, the SERCOM APB Clock remains active when going to Standby Sleep mode.

Figure 35-4. Timing Using Preloading

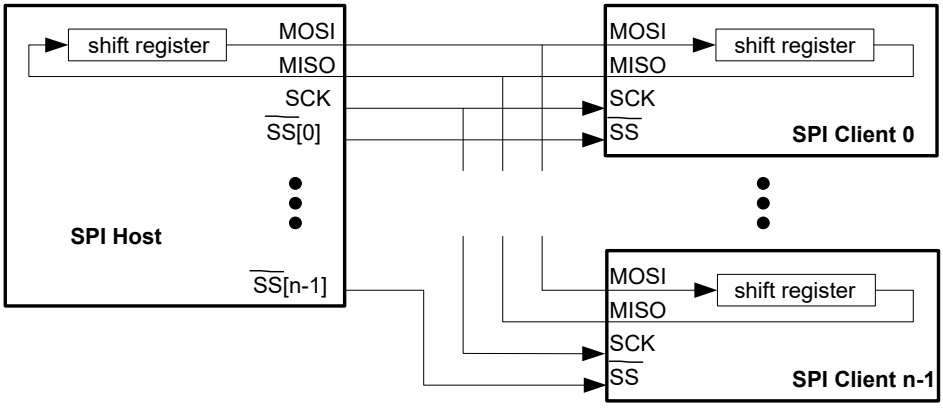


35.6.3.3 Host with Several Clients

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

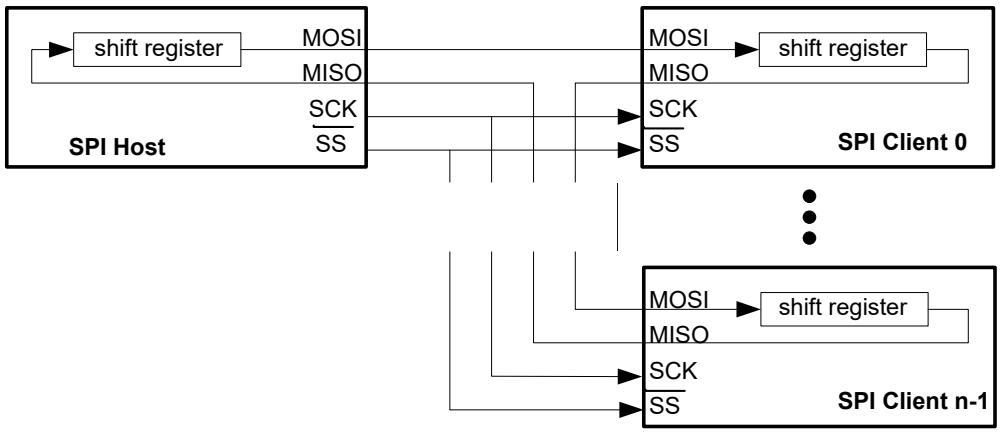
I/O pins to control the $\overline{SS}$ line to each of the clients on the bus, as shown in Multiple Clients in Parallel. In this configuration, the single selected SPI client will drive the tri-state MISO line.

Figure 35-5. Multiple Clients in Parallel



Another configuration is multiple clients in series, as in Multiple Clients in Series. 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.

Figure 35-6. Multiple Clients in Series



35.6.3.4 Loop-Back Mode

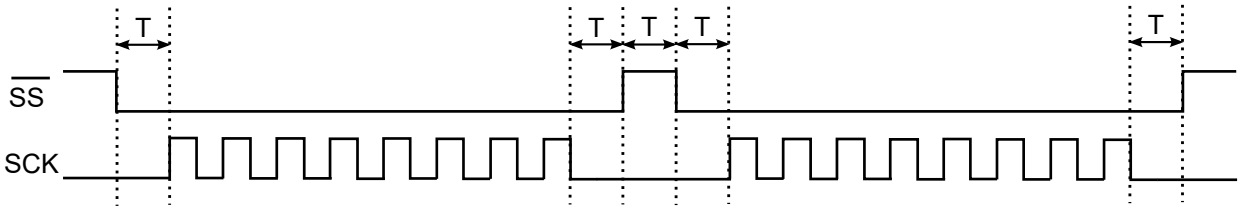
For loop-back mode, configure the Data In Pinout (CTRLA.DIPO) and Data Out Pinout (CTRLA.DOPO) to use the same data pins for transmit and receive. The loop-back is through the pad, so the signal is also available externally.

35.6.3.5 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 Hardware Controlled $\overline{SS}$, the time T is between one and two baud cycles depending on the SPI transfer mode.

Figure 35-7. Hardware Controlled $\overline{SS}$



T = 1 to 2 baud cycles

When CTRLB.MSSEN=0, the $\overline{SS}$ pin is controlled by user software and normal GPIO.

35.6.3.6 SPI Select ($\overline{SS}$) Low Detection

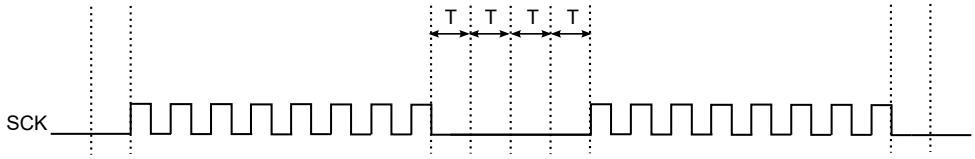
In client mode, the SPI can wake the CPU when the SPI Select ($\overline{SS}$) goes low. When the SPI Select Low Detect is enabled (CTRLB.SSDE=1), a high-to-low transition will set the SPI Select Low interrupt flag (INTFLAG.SSL) and the device will wake up if applicable.

35.6.3.7 Host Inter-Character Spacing

When configured as 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 gives an example for CTRLC.ICSPACE=4; In this case, the SCK is inactive for 4 baud cycles.

Figure 35-8. Four Cycle Inter-Character Spacing Example



T = 1 baud cycle

35.6.3.8 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 transaction to/from 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 data transfer begins. LENGTH.LEN must be enabled only when CTRLC.DATA32B is enabled.

The figure below shows 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.

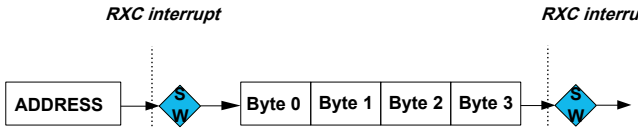
Figure 35-9. 32-bit Extension Byte Ordering

APB Write/Read	BYTE3	BYTE2	BYTE1	BYTE0
Bit Position	31			0

32-bit Extension Client Operation

The figure below shows 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. Once DATA has been registered in the core domain, INTFLAG.DRE will be raised, so that the next 32 bits can be written to the DATA register.

Figure 35-10. 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 will be 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 to for transmit before SCK starts, transmitted data will be 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 will be transmitted at the beginning of the next frame. If this is not desired, the SERCOM must be disabled and re-enabled in order to flush the Tx and Rx pipelines.

Writing the LENGTH register while a frame is in progress will produce 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 will be transmitted in the next frame.

32-bit Extension Host Operation

When using the SPI configured as 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. Once DATA has been registered in the core domain, INTFLAG.DRE will be raised so that the next 32 bits can be written to the DATA register.

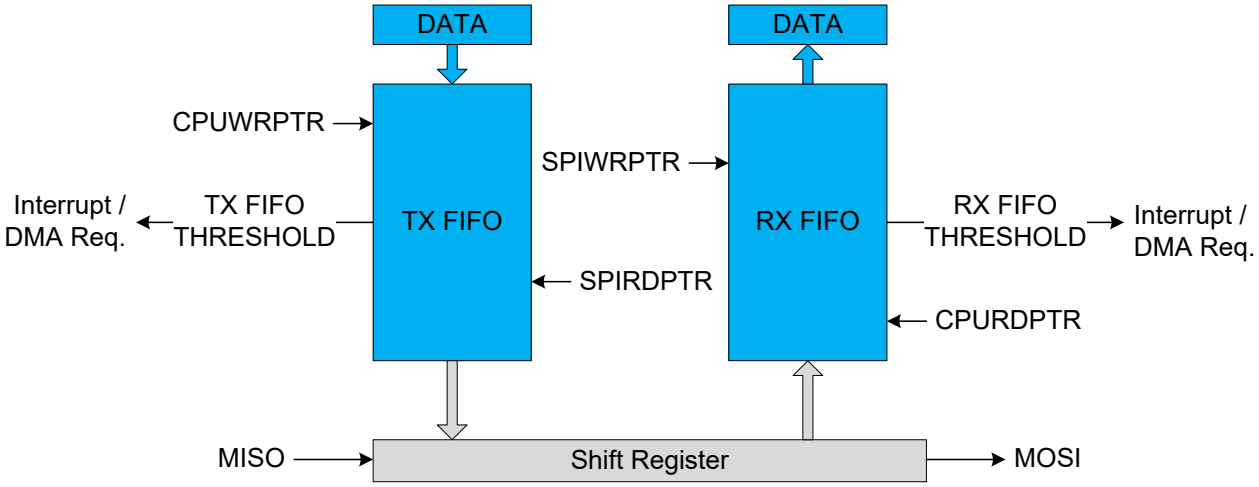
If multiple lengths are to be transmitted, INTFLAG.TXC must go high before writing DATA for subsequent lengths.

35.6.3.9 FIFO Operation

The SPI embeds up to 16-bytes FIFO capability. The receive / transmit buffer is considered to have the FIFO mode enabled when the FIFOEN bit in CTRLC register is set to a ‘1’ (CTRLC.FIFOEN = 1). By default, the FIFO can act as a 16-by-8-bit array, or as a 4-by-32-bit array, depending on the setting of the CTRLC.DATA32B bit.

The hardware around this array implements four pointers, called the CPU Write Pointer (CPUWRPTR), the CPU Read Pointer (CPURDPTR), the SPI Write pointer (SPIWRPTR) and the SPI Read pointer (SPIRDPTR). All of these pointers reset to ‘0’. The CPUWRPTR and CPURDPTR pointers are native to the CPU clock domain, while the SPIWRPTR and SPIRDPTR are native to the SPI domain. The location pointed to by the CPUWRPTR is the current TX FIFO. The location pointed to by the CPURDPTR becomes the current RX FIFO. Writes to DATA register by the CPU will point to TX FIFO. Reads to DATA register by the CPU will point to RX FIFO. The location pointed to by the SPIWRPTR / SPIRDPTR is logically the current shift register. Physically, the receive (shift-in) portion of the shift register is a single register located in the SCK clock domain, and the transmit (shift-out) portion is the buffer location pointed to by the SPIRDPTR. When a full word / byte is clocked into the SPI shift register, it is copied into the location pointed by SPIWRPTR.

Figure 35-11. FIFO Overview



The interrupts and DMA triggers are generated according to FIFO threshold settings in Control C register (CTRLC.TXTRHOLD, CTRLC.RXTRHOLD).

The Data Register Empty interrupt flag, and the DMA TX trigger respectivly, are generated when the available place in the TX FIFO is equal or higher than the threshold value defined by the CTRLC.TXTRHOLD settings. The Transfer complete interrupt is generated when the TX FIFO is empty and the last bit is shifted out.

The Receive Complete interrupt flag, and the DMA RX trigger respectivly, are generated when the number of bytes present in the RX FIFO equals or is higher than the threshold value defined by the CTRLC.RXTRHOLD settings. The ERROR interrupt flag is generated when both RX shifter and the RX FIFO are full.

The FIFO is fully accessible if the SERCOM is halted, by writting the corresponding CPU FIFO pointer in the FIFOPTR register. The RX or TX FIFO can be individually cleared, by setting the respective FIFO Clear bit in the Control B register (CTRLB.FIFOCLR). The FIFO Clear must be written before data transfer begins. Writing the FIFO Clear bits while a frame is in progress will produce unpredicatable results.

35.6.3.9.1 Pointer Operation in Host Mode

In Host mode, the transmit / receive sequence is started by the CPU writing one or more transmit words into the TX FIFO. The CPUWRPTR is incremented by 1 every time the CPU writes a word to the memory array. As soon as the CPUWRPTR becomes not-equal to SPIRDPTR (FIFOSPACE.TXSPACE != 0), the SPI transmits the data pointed to by the SPIRDPTR through MOSI. When a complete word is shifted in, the SPIRDPTR is compared to CPUWRPTR. If they are not equal, SPIRDPTR is incremented, another byte / word is shifted in/out, and so on. When the CPU completes a read from the RX FIFO location (FIFOSPACE.RXSPACE != 0), the CPURDPTR pointer is incremented. When both RX shifter and RX FIFO are full, the Buffer Overflow status bit is set (STATUS.BUFOVF) and optional ERROR interrupt is generated. The module will not respond to SCK transitions while BUFOVF is ‘1’, effectively disabling the module until software reads DATA register.

All pointers increment to their maximum value, dictated by CTRLC.DATA32B bit, and then rolls over to ‘0’. CPURDPTR will not increment past SPIWRPTR. In other words, if CPURDPTR = SPIWRPTR, and the CPU attempts another read, the pointer will stay at the value of SPIWRPTR.

35.6.3.9.2 Pointer Operation in Client Mode

In Client mode, the transmit / receive sequence is started by the SPI receiving an SCK clock pulse. As soon as an SCK pulse is received, the SPI transmits the data pointed to by the SPIRDPTR. When a complete data is shifted in, the SPIRDPTR / SPIWRPTR are incremented, another data are shifted in/out, and so on. The newly received data is written to the RX FIFO location pointed to by SPIWRPTR. The CPUWRPTR is incremented by one every time the CPU writes a new data to the TX FIFO memory array. If the CPUWRPTR is pointing to location n, and SPIWRPTR is pointing to location n, and a wrap is not detected, the CPUWRPTR will auto-increment to SPIWRPTR, to keep up with SPIWRPTR. This is so that the next data to be transmitted will be placed in the correct position in the storage element. The CPU can read data from RX FIFO as long as FIFOSPACE.RXSPACE != 0. When both RX shifter and RX FIFO are full, the Buffer Overflow status bit is set (STATUS.BUFOVF) and optional ERROR interrupt is generated. The module will not respond to SCK transitions while BUFOVF is ‘1’, effectively disabling the module until software reads RX FIFO. All pointers increment to their maximum value, dictated by CTRLC.DATA32B bit, and then roll over

to '0'. CPURDPTR will not increment past SPIWRPTR. In other words, if CPURDPTR = SPIWRPTR, and the CPU attempts another read, the pointer will stay at the value of SPIWRPTR.

35.6.4 DMA, Interrupts, and Events

Table 35-5. 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 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	
SPI Select Low (SSL)	NA	Yes	
Error (ERROR)	NA	Yes	

35.6.4.1 DMA Operation

The SPI generates the following DMA requests:

- Data received (RX): The request is set when data is 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.

35.6.4.2 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).

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, refer to the INTFLAG register description.

The value of INTFLAG indicates which interrupt is executed. Note that interrupts must be globally enabled for interrupt requests. Refer to Nested Vector Interrupt Controller for details.

35.6.5 Sleep Mode Operation

The behavior in Sleep mode is depending 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_SERCOM_CORE will continue to run in idle sleep mode and in Standby Sleep mode. Any interrupt can wake up the device.

- Host operation, CTRLA.RUNSTDBY=0: GLK_SERCOMx_CORE will be 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 will be dropped, including the ongoing transaction.

35.6.6 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 - refer to the Debug Control (DBGCTRL) register for details.

35.6.7 Synchronization

Some registers (or bit fields within a register) require synchronization when read and/or written.

Synchronization is denoted by the "Read-Synchronized" (or "Read-Synchronized Bits”) and/or "Write-Synchronized" (or "Write-Synchronized Bits”) property in each individual register description.

For more details, refer to Register Synchronization.

35.7 Register Summary

Refer to the Registers Description section for more details on register properties and access permissions.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRLA	31:24		DORD	CPOL	CPHA	FORM[3:0]			
		23:16			DIPO[1:0]				DOPO[1:0]	
		15:8								IBON
		7:0	RUNSTDBY			MODE[2:0]			ENABLE	SWRST
0x04	CTRLB	31:24								
		23:16	FIFOCLR[1:0]						RXEN	
		15:8	AMODE[1:0]		MSSEN				SSDE	
		7:0		PLOADEN				CHSIZE[2:0]		
0x08	CTRLC	31:24	TXTRHOLD[1:0]		RXTRHOLD[1:0]		FIFOEN			DATA32B
		23:16								
		15:8								
		7:0			ICSPACE[5:0]					
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	15:8					LENERR			
		7:0						BUFOVF		
0x1C	SYNCBUSY	31:24								
		23:16								
		15:8								
		7:0				LENGTH		CTRLB	ENABLE	SWRST
0x20 ... 0x21	Reserved									
0x22	LENGTH	15:8								LENEN
		7:0	LEN[7:0]							
0x24	ADDR	31:24								
		23:16	ADDRMASK[7:0]							
		15:8								
		7:0	ADDR[7:0]							
0x28	DATA	31:24	DATA[31:24]							
		23:16	DATA[23:16]							
		15:8	DATA[15:8]							
		7:0	DATA[7:0]							
0x2C ... 0x2F	Reserved									
0x30	DBGCTRL	7:0								DBGSTOP
0x31 ... 0x33	Reserved									
0x34	FIFOSPACE	15:8				RXSPACE[4:0]				
		7:0				TXSPACE[4:0]				
0x36	FIFOPTR	15:8					CPURDPTR[3:0]			
		7:0					CPUWRPTR[3:0]			

35.7.1 Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected Bits, Write-Synchronized Bits

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.
Note: This bit is enable-protected. 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.
Note: This bit is enable-protected. 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.
Note: This bit is enable-protected. 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 is sampled on a leading SCK edge and changed on a trailing SCK edge.

Value	Description
1	The data is 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.

Note: This bit field is enable-protected. This bit field is not synchronized.

Value	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.

Note: This bit field is enable-protected. This bit field is not synchronized.

Value	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

This bit defines the available pad configurations for Data Out (DO) and the Serial Clock (SCK).

In client operation, the SPI Select ($\overline{SS}$) line is controlled by DOPO.

In host operation, the SPI Select ($\overline{SS}$) line 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.

Note: This bit field is enable-protected. This bit field is not synchronized.

Value	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	-	-	-	-	-
0x2	PAD[3]	PAD[1]	PAD[2]	PAD[2]	Any GPIO configured by the application
0x3	-	-	-	-	-

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.

Note: This bit is enable-protected. 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.

These bits are not synchronized.

Value	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.

.....continued		
Value	**Client**	**Host**
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 of the SERCOM.
0x2: SPI client operation
0x3: SPI host operation
Note: This bit field is enable-protected. This bit field is not synchronized.

Bit 1 – ENABLE Enable

Notes:

1. This bit is write-synchronized: SYNCBUSY.ENABLE must be checked to ensure the CTRLA.ENABLE synchronization is complete.
2. 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 will be disabled.
Writing "1' to CTRL.SWRST will always take precedence, meaning that all other writes in the same write-operation will be discarded. Any register write access during the ongoing Reset will result in an APB error. Reading any register will return the Reset value of the register.

Notes:

1. When the CTRLA.SWRST is written, the user must poll SYNCBUSY.SWRST bit to know when the reset operation is complete.
2. During a SWRST, access to registers/bits without SWRST are disallowed until SYNCBUSY.SWRST is cleared by hardware.
3. This bit is not enable protected.

Value	Description
0	There is no Reset operation ongoing.
1	The Reset operation is ongoing.

35.7.2 Control B

Name: CTRLB
Offset: 0x04
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected Bits, Write-Synchronized Bits

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

Note: This bit field is write-synchronized: SYNCBUSY.CTRLB must be checked to ensure the CTRLB.FIFOCLR synchronization is complete.

Note: This bit field is not enable-protected.

Value	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 will disable the SPI receiver immediately. The receive buffer will be flushed, data from ongoing receptions will be lost and STATUS.BUFOVF will be cleared.
Writing '1' to this bit when the SPI is disabled will set CTRLB.RXEN immediately. When the SPI is enabled, CTRLB.RXEN is cleared and the receiver enable is only effective at the end of the SYNCBUSY.CTRLB synchronization.
Writing '1' to this bit when the SPI is enabled requires to wait the end of the SYNCBUSY.CTRLB synchronization to ensure the receiver is enabled.

Note: This bit is write-synchronized: SYNCBUSY.CTRLB must be checked to ensure the CTRLB.RXEN synchronization is complete.

Note: This bit is not enable-protected.

Value	Description
0	The receiver is disabled or being enabled.
1	The receiver is enabled or it will be 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.

Note: This bit field is enable-protected. This bit field is not synchronized.

Value	Name	Description
0x0	MASK	ADDRMASK is used as a mask to the ADDR register
0x1	2ADDRS	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.

Note: This bit is enable-protected. 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.

Note: This bit is enable-protected. 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 will enable 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 will be transferred immediately to the Shift register.

Note: This bit is enable-protected. This bit is not synchronized.

Bits 2:0 – CHSIZE[2:0] Character Size

Note: This bit field is enable-protected. This bit field is not synchronized.

Value	Name	Description
0x0	8BIT	8 bits
0x1	9BIT	9 bits
0x2-0x7	-	Reserved

35.7.3 Control C

Name: CTRLC
Offset: 0x08
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
	TXTRHOLD[1:0]		RXTRHOLD[1:0]		FIFOEN			DATA32B
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
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

Bits 31:30 – TXTRHOLD[1:0] Transmit FIFO Threshold
These bits define the threshold for generating the Data Register Empty interrupt and DMA TX trigger.

Value	Name	Description
0	DEFAULT	Interrupt and DMA triggers can be generated as long as the FIFO is not full.
1	HALF	Interrupt and DMA triggers are generated when half FIFO space is free.
2	EMPTY	Interrupt and DMA triggers are generated when the FIFO is empty.
3	-	Reserved

Bits 29:28 – RXTRHOLD[1:0] Receive FIFO Threshold
These bits define the threshold for generating the RX Complete interrupt and DMA RX trigger.

Value	Name	Description
0	DEFAULT	Interrupt and DMA triggers can be generated when a DATA is present in the FIFO.
1	HALF	Interrupt and DMA triggers can be generated only when the FIFO is half-full.
2	FULL	Interrupt and DMA triggers can be generated only when the FIFO is full.
3	-	Reserved

Bit 27 – FIFOEN FIFO Enable
This bit enables the FIFO operation.

Value	Description
0	FIFO operation is disabled
1	FIFO operation is enabled

Bit 24 – DATA32B Data 32 Bit
This bit enables 32-bit Extension for read and write transactions to the DATA register.
When disabled, access is according to CTRLB.CHSIZE.

Value	Description
0	Transactions from and to DATA register are 8-bit

Value	Description
1	Transactions from and to 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

35.7.4 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, as described in the SERCOM Clock Generation – Baud-Rate Generator.

35.7.5 Interrupt Enable Clear

Name: INTENCLR
Offset: 0x14
Reset: 0x00
Property: PAC Write-Protection

This register allows the user to disable an interrupt without 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 will clear 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 will clear 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 will clear 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 will clear the Transmit Complete Interrupt Enable bit, which disable 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 will clear 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.

35.7.6 Interrupt Enable Set

Name: INTENSET
Offset: 0x16
Reset: 0x00
Property: PAC Write-Protection

This register allows the user to disable 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 will set 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 will set 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 will set 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 will set 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 will set 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.

35.7.7 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 will set this flag have corresponding status flags in the STATUS register. The BUFOVF error and the LENERR error will set this interrupt flag.
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear 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 will clear 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 will be 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 have been 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 will clear 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.

35.7.8 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 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 will clear 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 will indicate 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. See also 35.7.1. CTRLA.IBON for overflow handling.
When set, the corresponding RxDATA will be zero.
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear it.

Value	Description
0	No Buffer Overflow has occurred.
1	A Buffer Overflow has occurred.

35.7.9 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 will be set until synchronization is complete. If the LENGTH register is written to while SYNCBUSY.LENGTH is asserted, an APB error is 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 – 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 will be 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.

35.7.10 Length

Name: LENGTH
Offset: 0x22
Reset: 0x0000
Property: Write-Synchronized

Note: This register is write-synchronized: SYNCBUSY.LENGTH must be checked to ensure the LENGTH register synchronization is complete.

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

35.7.11 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).

35.7.12 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 will return the contents of the receive data buffer. The register should 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 will write the transmit data buffer. This register should 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.

35.7.13 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.

35.7.14 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 Filled Space
These bits return the number of available locations in the TX FIFO (bytes or words, depending on CTRLC.DATA32B setting).

35.7.15 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.

36. Inter-Integrated Circuit (SERCOM I²C)

36.1 Overview

The inter-integrated circuit (I²C) interface is one of the available modes in the serial communication interface (SERCOM).

The I²C interface uses the SERCOM transmitter and receiver configured as shown in the Block Diagram. 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 I²C host or an I²C client. Both host and client have an interface containing a shift register, a transmit buffer and a receive buffer.

The transmit buffer consists of a 8-bytes write FIFO buffer.

The receive buffer consists of a 8-bytes write FIFO buffer.

In addition, the I²C host uses the SERCOM baud-rate generator, while the I²C client uses the SERCOM address match logic.

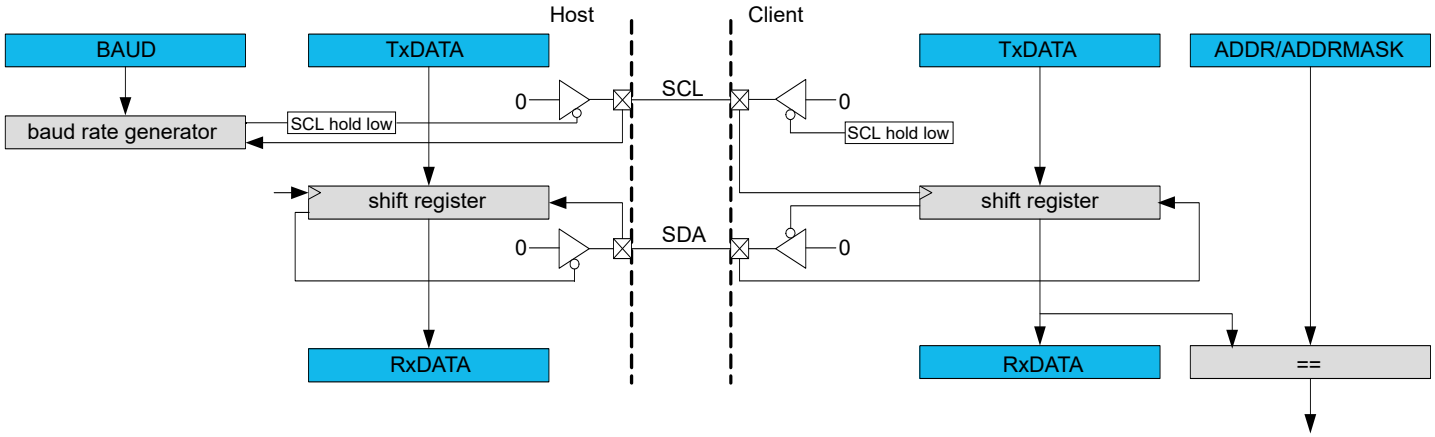
36.2 Features

SERCOM I²C includes the following features:

- Host or client operation
- DMA operations supported
- Philips I²C compatible
- SMBus™ compatible
- PMBus compatible
- Support of 100kHz and 400kHz, 1MHz and 3.4MHz I²C mode
- 32-bit Data Extension for better system bus utilization
- Receive buffer: 8-bytes FIFO
- Transmit buffer: 8-bytes FIFO
- 4-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

36.3 Block Diagram

Figure 36-1. I²C Single-Host Single-Client Interconnection



36.4 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 (4-wire operation)
PAD[3]	Digital I/O	SCL_OUT (4-wire operation)

⚠ CAUTION When the SERCOM is used in I²C mode, the SERCOM controls the direction and value of the I/O pins. The PORT Control bit, PINCFGn.DRVSTR, is still effective for the SERCOM output pins. The PORT Control bit, PINCFGn.PULLEN, is still effective on the SERCOM input pins, but is limited to the enabling or disabling of a pull down only (it is not possible to enable or disable a pull up). If the receiver or transmitter is disabled, these pins can be used for other purposes.

Important: I²C is not supported on all SERCOM pins.

The following table lists the SERCOM pins which support I²C:

Table 36-1. SERCOM I²C Pins

Pin Name	Peripheral C SERCOM	Peripheral D SERCOM ALT
PA08	SERCOM1/PAD[0]	SERCOM2/PAD[0]
PA09	SERCOM1/PAD[1]	SERCOM2/PAD[1]
PA12	SERCOM2/PAD[0]	SERCOM4/PAD[0]
PA13	SERCOM2/PAD[1]	SERCOM4/PAD[1]
PA16	SERCOM1/PAD[0]	SERCOM0/PAD[0]
PA17	SERCOM1/PAD[1]	SERCOM0/PAD[1]

.....continued

Pin Name	Peripheral C SERCOM	Peripheral D SERCOM ALT
PA22	SERCOM0/PAD[0]	SERCOM2/PAD[0]
PA23	SERCOM0/PAD[1]	SERCOM2/PAD[1]
PB12	SERCOM3/PAD[0]	-
PB13	SERCOM3/PAD[1]	-
PB16	SERCOM5/PAD[0]	-
PB17	SERCOM5/PAD[1]	-
PB30	SERCOM1/PAD[0]	SERCOM5/PAD[0]
PB31	SERCOM1/PAD[1]	SERCOM5/PAD[1]

⚠ CAUTION

- I^2C is not supported on SERCOM3 and SERCOM5 for 48-pin packages. Refer to the Pinout for more information.
- I/Os for SERCOM peripherals are grouped into IO sets, listed in their 'IOSET' column in the pinout tables. For these peripherals, it is mandatory to use I/Os that belong to the same IO set. The timings are not guaranteed when IOs from different IO sets are mixed. Refer to the Pinout and Packaging chapter to get IOSET definitions.

36.5 Peripheral Dependencies

Table 36-2. SERCOM Configuration Summary

Peripheral name	Base address	NVIC IRQ Index	MCLK AHB/APB Clocks	GCLK Peripheral Channel Index (GCLK.PCHCTRL)	PAC Peripheral Identifier Index (PAC.WRCTRL)	DMA Trigger Source Index (DMAC.CHCTRLB)	Power Domain (PM.STDBYCFG)
SERCOM0	0x42000400	31: bit 0 32: bit 1 33: bit 2 34: bit 3-7	CLK_SERCOM0_APB	17: GCLK_SERCOM0_CORE 16: GCLK_SERCOM0_SLOW	65	4: RX 5: TX	PDSW
SERCOM1	0x42000800	35: bit 0 36: bit 1 37: bit 2 38: bit 3-7	CLK_SERCOM1_APB	18: GCLK_SERCOM1_CORE 16: GCLK_SERCOM1_SLOW	66	6: RX 7: TX	PDSW
SERCOM2	0x42000C00	39: bit 0 40: bit 1 41: bit 2 42: bit 3-7	CLK_SERCOM2_APB	19: GCLK_SERCOM2_CORE 16: GCLK_SERCOM2_SLOW	67	8: RX 9: TX	PDSW
SERCOM3	0x42001000	43: bit 0 44: bit 1 45: bit 2 46: bit 3-7	CLK_SERCOM3_APB	20: GCLK_SERCOM3_CORE 16: GCLK_SERCOM3_SLOW	68	10: RX 11: TX	PDSW

.....continued

Peripheral name	Base address	NVIC IRQ Index	MCLK AHB/APB Clocks	GCLK Peripheral Channel Index (GCLK.PCHCTRL)	PAC Peripheral Identifier Index (PAC.WRCTRL)	DMA Trigger Source Index (DMAC.CHCTRLB)	Power Domain (PM.STDBYCFG)
SERCOM4	0x42001400	47: bit 0 48: bit 1 49: bit 2 50: bit 3-7	CLK_SERCOM4_APB	21: GCLK_SERCOM4_CORE 16: GCLK_SERCOM4_SLOW	69	12: RX 13: TX	PDSW
SERCOM5	0x42001800	51: bit 0 52: bit 1 53: bit 2 54: bit 3-7	CLK_SERCOM5_APB	22: GCLK_SERCOM5_CORE 16: GCLK_SERCOM5_SLOW	70	14: RX 15: TX	PDSW

36.6 Functional Description

36.6.1 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 36-2. Transaction Diagram Symbols

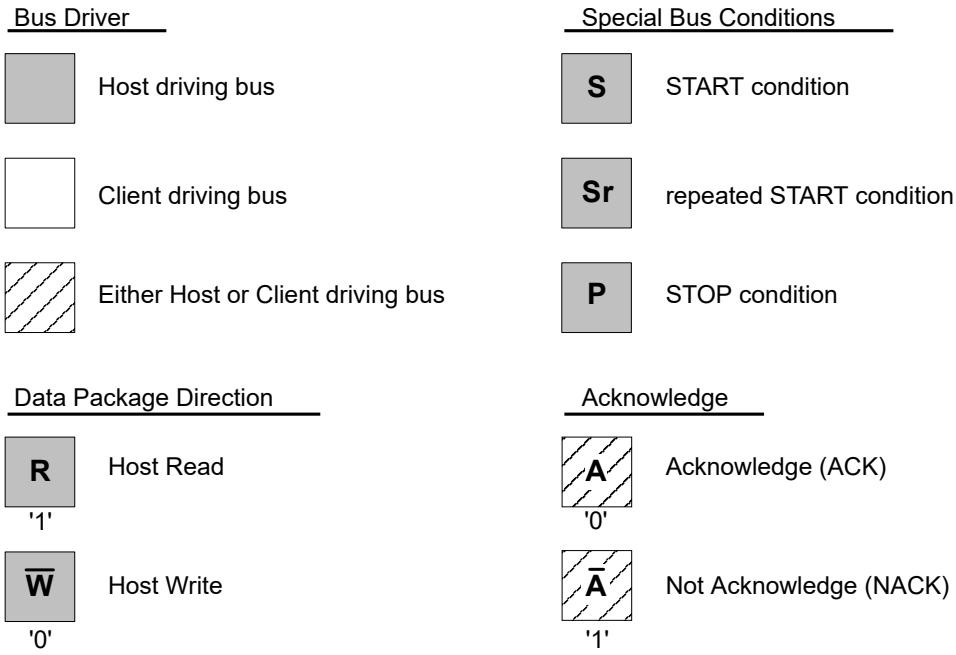
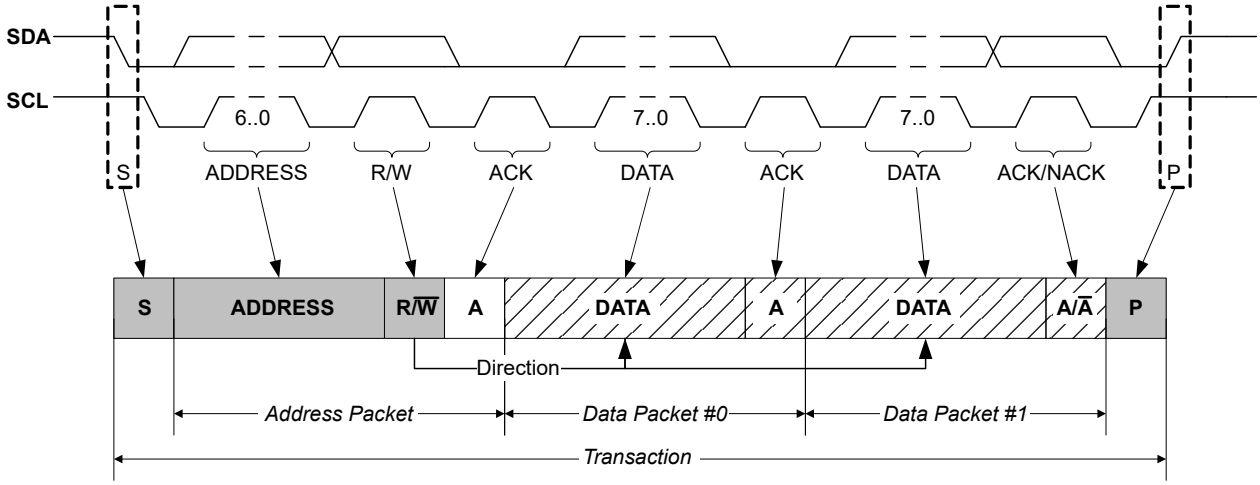


Figure 36-3. Basic I²C Transaction Diagram



36.6.2 Basic Operation

36.6.2.1 Initialization

The following registers are enable-protected, meaning they can be written only when the I²C interface is disabled (CTRLA.ENABLE is '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) in host operation
- Address register (ADDR) in client operation

When the I²C is enabled or is being enabled (CTRLA.ENABLE=1), writing to these registers will be discarded. If the I²C is being disabled, writing to these registers will be 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 value in the CTRLA register (CTRLA.SDAHOLD).
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.LOWTOUTEN).
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.

36.6.2.2 Enabling, Disabling, and Resetting

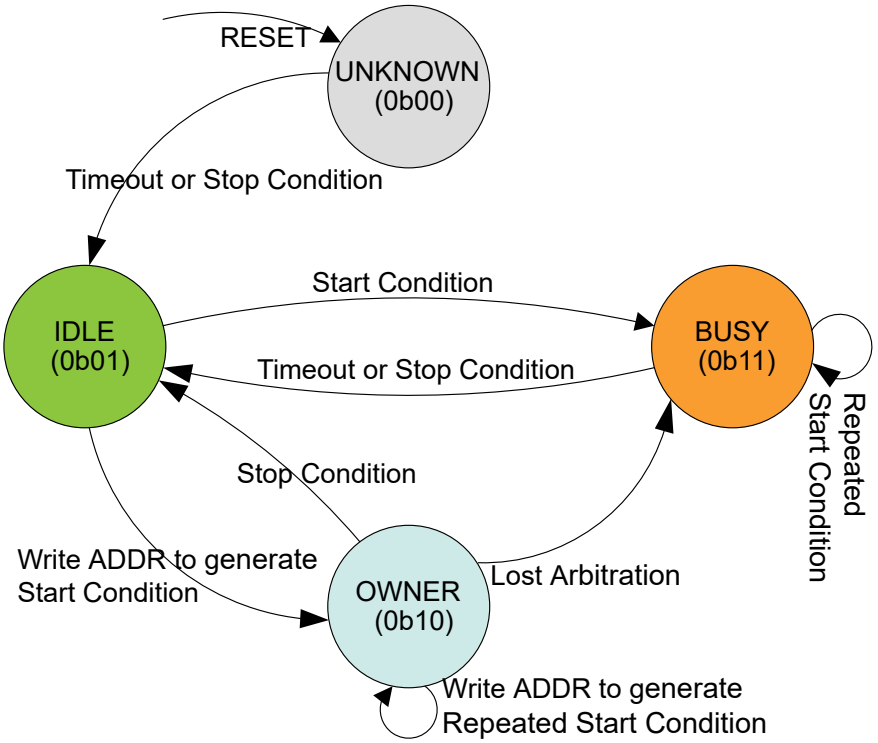
This peripheral is enabled by writing '1' to the Enable bit in the Control A register (CTRLA.ENABLE), and disabled by writing '0' to it.

Writing '1' to the Software Reset bit in the Control A register (CTRLA.SWRST) will reset all registers of this peripheral to their initial states, except the DBGCTRL register, and the peripheral is disabled.

36.6.2.3 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_SERCOM_x clocks. The start and stop detectors and the bit counter are all essential in the process of determining the current bus state. The bus state is determined according to the following figure. 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 figure is shown in binary.

Figure 36-4. Bus State Diagram



The bus state machine is active when the I²C host is enabled.

After the I²C host has been 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: Once 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 will re-enter 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, i.e., arbitration was not lost, the I²C host can issue a stop condition, which will change 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 will change 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').

36.6.2.4 I²C Host Operation

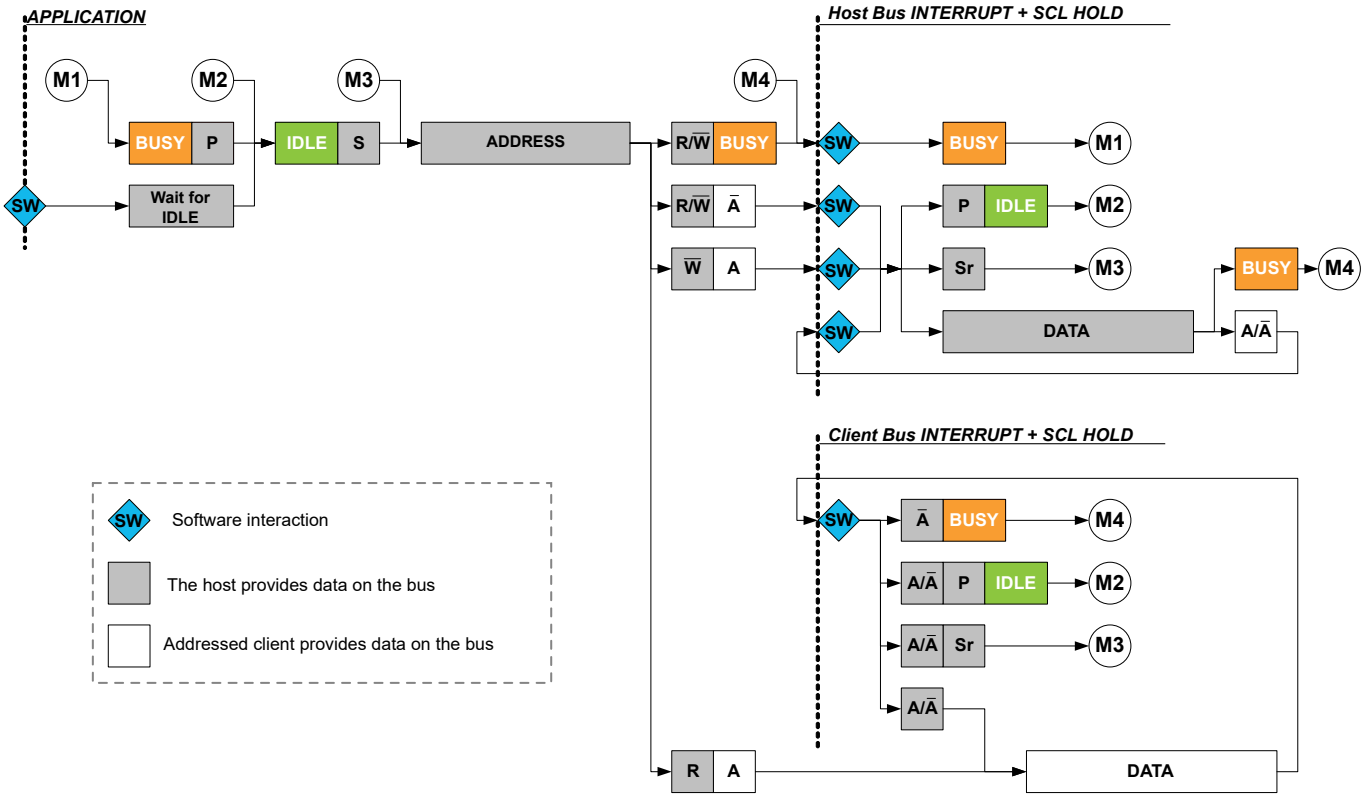
The I²C host is byte-oriented and interrupt based. The number of interrupts generated is kept at a minimum by automatic handling of most incidents. The software driver complexity and code size are reduced by auto-triggering of operations, and a special smart mode, which can be enabled by the Smart Mode 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 the following figure. 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 reference for the description of the I²C host operation throughout the document.

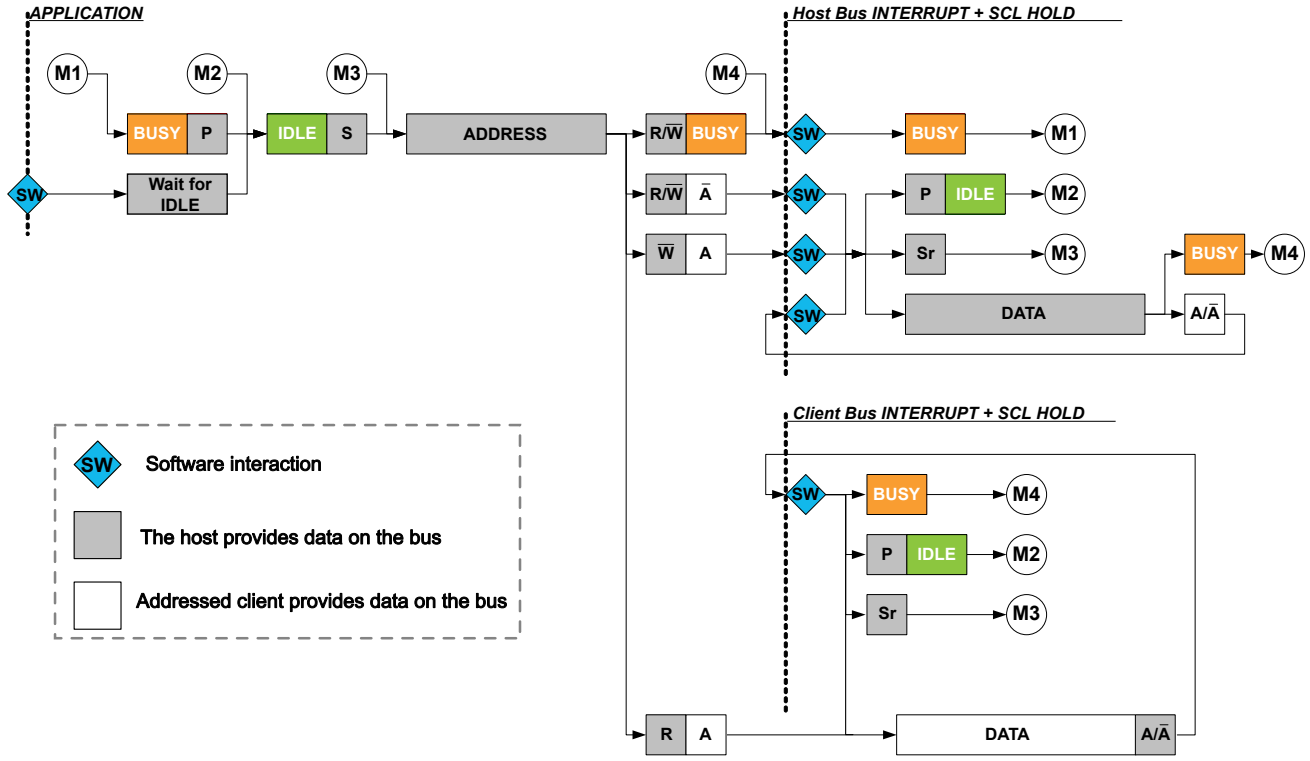
Figure 36-5. I²C Host Behavioral Diagram (SCLSM=0)



In the second strategy (CTRLA.SCLSM=1), interrupts only occur after the ACK bit, as in the following figure. This strategy can be used when it is not necessary to check DATA before acknowledging.

Note: I²C High-speed (*Hs*) mode requires CTRLA.SCLSM=1.

Figure 36-6. I²C Host Behavioral Diagram (SCLSM=1)



36.6.2.4.1 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
- High-speed mode (*Hs*) up to 3.4 MHz

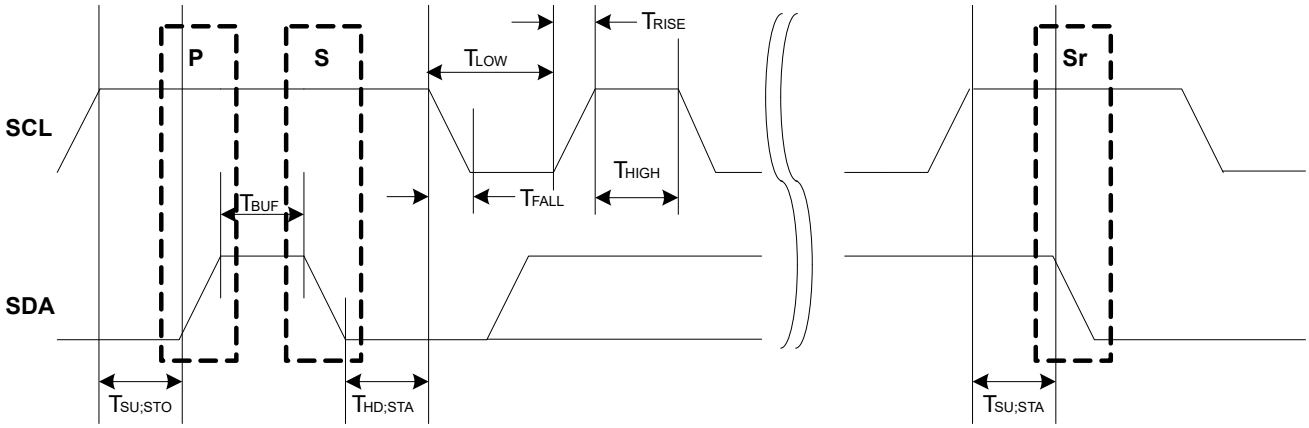
The Host clock configuration for *Sm, Fm,* and *Fm+* are described in Clock Generation (Standard-Mode, Fast-Mode, and Fast-Mode Plus). For *Hs*, refer to Host Clock Generation (High-Speed Mode).

Clock Generation (Standard-Mode, Fast-Mode, and Fast-Mode Plus)

In I²C *Sm, Fm*, and *Fm+* mode, 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} will be considered as part of T_{LOW}. Likewise, T_{RISE} will be in a state between T_{LOW} and T_{HIGH} until a high state has been detected.

Figure 36-7. 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
- $T_{SU;STO}$ – Set-up time for stop condition
- T_{BUF} – Bus free time between stop and start conditions
- $T_{HD;STA}$ – Hold time (repeated) start condition
- $T_{SU;STA}$ – Set-up 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.

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 will give 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 should be set accordingly. At a minimum, BAUD.BAUD and/or BAUD.BAUDLOW must be non-zero.

Startup 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 startup 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 user, the Smart Mode cannot be enabled.

For more information, refer to the Electrical Specifications chapter.

Host Clock Generation (High-Speed Mode)

For I²C *Hs* transfers, there is no SCL synchronization. Instead, the SCL frequency is determined by the GCLK_SERCOMx_CORE frequency (f_{GCLK}) and the High-Speed Baud setting in the Baud register (BAUD.HSBAUD). When BAUD.HSBAUDLOW=0, the HSBAUD value will determine both SCL high and SCL low. In this case the following formula determines the SCL frequency.

$$f_{SCL} = \frac{f_{GCLK}}{2 + 2 \cdot HS\,BAUD}$$

When HSBAUDLOW is non-zero, the following formula determines the SCL frequency.

$$f_{SCL} = \frac{f_{GCLK}}{2 + HS\,BAUD + HSBAUDLOW}$$

Note: The I²C standard *Hs* (High-speed) requires a nominal high to low SCL ratio of 1:2, and HSBAUD should be set accordingly. At a minimum, BAUD.HSBAUD and/or BAUD.HSBAUDLOW must be non-zero.

36.6.2.4.2 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 36.6.1. Principle of Operation. 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 I²C host receives an ACK from the I²C client, the I²C host proceeds to receive the next byte of data from the I²C 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 I²C operation to continue:

- Let the I²C 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 should be sent.

36.6.2.4.3 Transmitting Data Packets

When an address packet with direction Host Write (see Figure 36-3) was transmitted successfully , INTFLAG.MB will be set. The I²C host will start transmitting data via the I²C bus by writing to DATA.DATA, and monitor continuously for packet collisions. I

If a collision is detected, the I²C host will lose arbitration and STATUS.ARBLOST will be set. If the transmit was successful, the I²C host will receive an ACK bit from the I²C 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 I²C 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 I²C host is not allowed to continue transmitting data packets if a NACK is received from the I²C client.

36.6.2.4.4 Receiving Data Packets (SCLSM=0)

When INTFLAG.SB is set, the I²C host will already have received one data packet. The I²C 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.

36.6.2.4.5 Receiving Data Packets (SCLSM=1)

When INTFLAG.SB is set, the I²C 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.

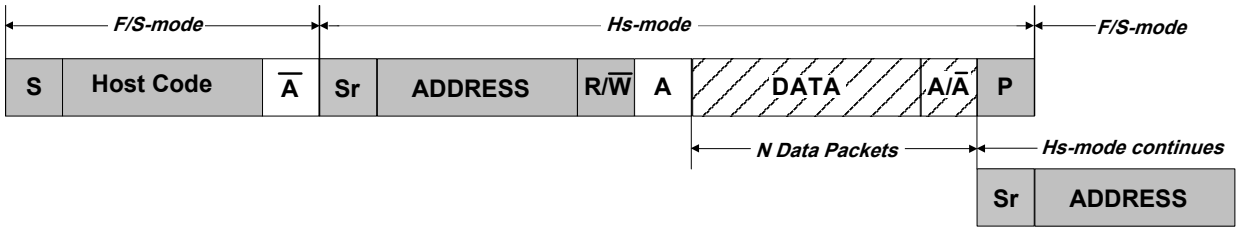
36.6.2.4.6 High-Speed Mode

High-speed transfers are a multi-step process, see High Speed Transfer.

First, a host code (0b00001nnn, where 'nnn' is a unique host code) is transmitted in Full-speed mode, followed by a NACK since no client should acknowledge. Arbitration is performed only during the Full-speed Host Code phase. The host code is transmitted by writing the host code to the address register (ADDR.ADDR) and writing the high-speed bit (ADDR.HS) to '0'.

After the host code and NACK have been transmitted, the host write interrupt will be asserted. In the meanwhile, the client address can be written to the ADDR.ADDR register together with ADDR.HS=1. Now in High-speed mode, the host will generate a repeated start, followed by the client address with RW-direction. The bus will remain in High-speed mode until a stop is generated. If a repeated start is desired, the ADDR.HS bit must again be written to '1', along with the new address ADDR.ADDR to be transmitted.

Figure 36-8. High Speed Transfer



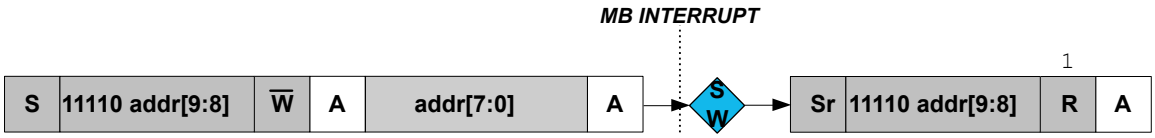
Transmitting in High-speed mode requires the I²C host to be configured in High-speed mode (CTRLA.SPEED=0x2) and the SCL clock stretch mode (CTRLA.SCLSM) bit set to '1'.

36.6.2.4.7 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 will be transmitted, see 10-bit Address Transmission for a Read Transaction. 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 zero.

If the host receives a NACK after the first byte, the write interrupt flag will be raised and the STATUS.RXNACK bit will be set. If the first byte is acknowledged by one or more clients, then the host will proceed 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 36-9. 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. Once the Host on Bus interrupt is asserted, Write 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.

36.6.2.5 I²C Client Operation

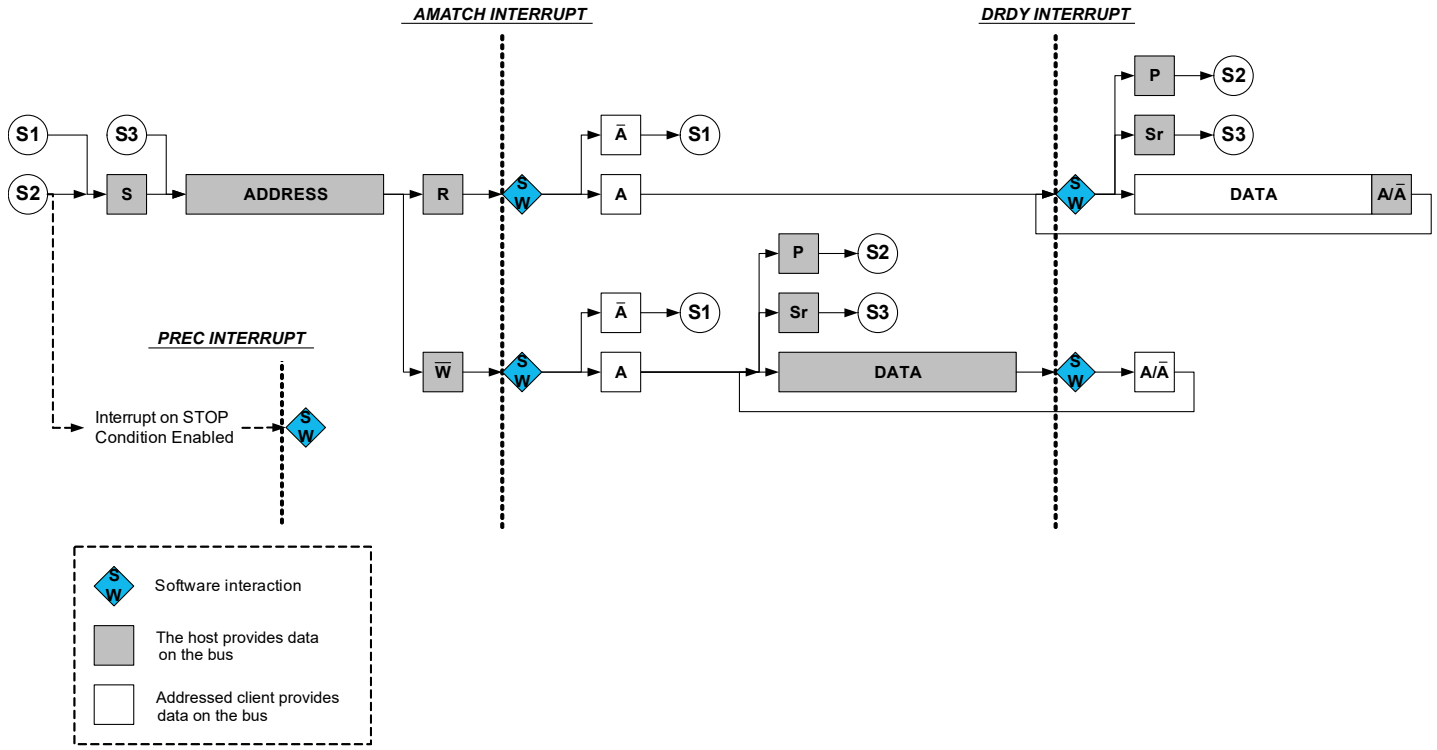
The I²C client is byte-oriented and interrupt-based. The number of interrupts generated is kept at a minimum by automatic handling of most events. The software driver complexity and code size are reduced by auto-triggering of operations, and a special smart mode, which can be enabled by the Smart Mode 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 the following figure. The circles labeled "S*n*" (S1, S2..) indicate the nodes the bus logic can jump to, based on software or hardware interaction.

This diagram is used as reference for the description of the I²C client operation throughout the document.

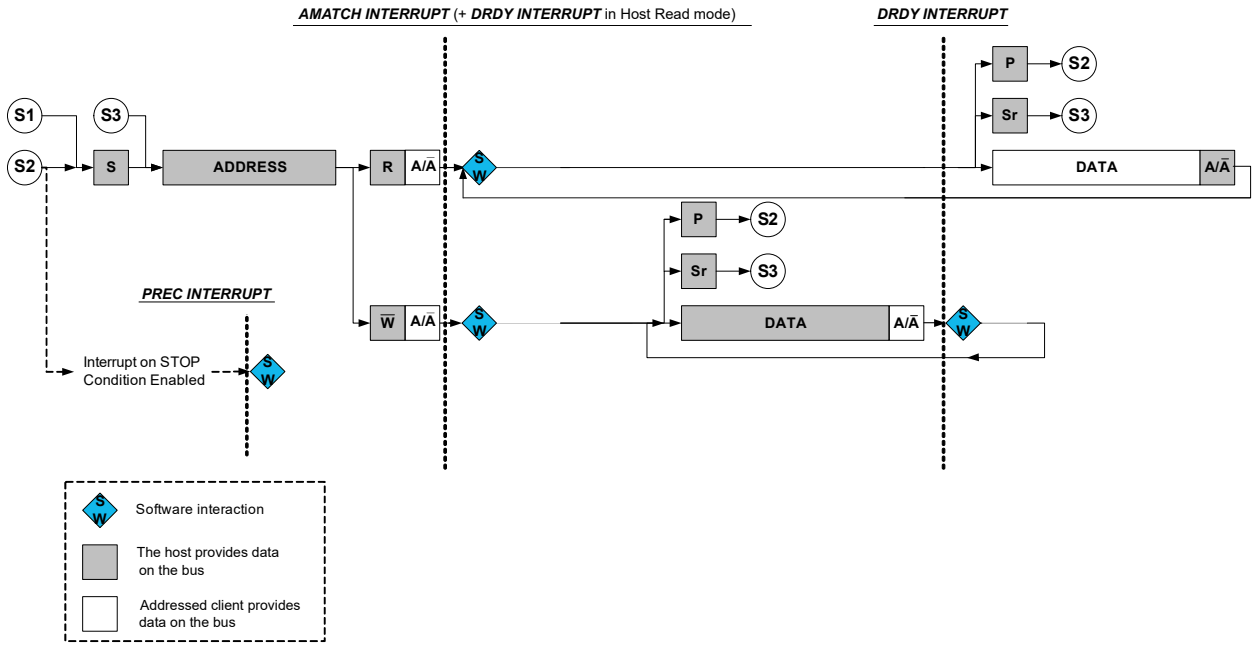
Figure 36-10. I²C Client Behavioral Diagram (SCLSM=0)



In the second strategy (CTRLA.SCLSM=1), interrupts only occur after the ACK bit is sent as shown in the figure below. This strategy can be used when it is not necessary to check DATA before acknowledging. For host reads, an address and data interrupt will be issued simultaneously after the address acknowledge. However, for host writes, the first data interrupt will be seen after the first data byte has been received by the client and the acknowledge bit has been sent to the host.

Note: For I²C High-speed mode (*Hs*), SCLSM=1 is required.

Figure 36-11. I²C Client Behavioral Diagram (SCLSM=1)



36.6.2.5.1 Receiving Address Packets (SCLSM=0)

When CTRLA.SCLSM=0, the I²C client stretches the SCL line according to Figure 36-10. When the I²C client is properly configured, it will wait for a start condition.

When a start condition is detected, the successive address packet will be received and checked by the address match logic. If the received address is not a match, the packet will be rejected, and the I²C client will wait for a new start condition. If the received address is a match, the Address Match bit in the Interrupt Flag register (INTFLAG.AMATCH) will be set.

SCL will be 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 will be 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 has been received from the I²C host, one of two cases will arise 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 will immediately acknowledge 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 will wait for data to be received. Data, repeated start or stop can be received.

If a NACK is sent, the I²C client will wait for a new start condition and address match. Typically, software will immediately acknowledge the address packet by sending an ACK/NACK. The I²C client command CTRLB.CMD = 3 can be used for both read and write operation as the command execution is dependent on STATUS.DIR.

Writing '1' to INTFLAG.AMATCH will also cause an ACK/NACK to be sent corresponding to the CTRLB.ACKACT bit.

36.6.2.5.2 Receiving Address Packets (SCLSM=1)

When SCLSM=1, the I²C client will stretch the SCL line only after an ACK, see Client Behavioral Diagram (SCLSM=1). When the I²C client is properly configured, it will wait for a start condition to be detected.

When a start condition is detected, the successive address packet will be received and checked by the address match logic.

If the received address is not a match, the packet will be rejected and the I²C client will wait for a new start condition.

If the address matches, the acknowledge action as configured by the Acknowledge Action bit Control B register (CTRLB.ACKACT) will be sent and the Address Match bit in the Interrupt Flag register (INTFLAG.AMATCH) is set. SCL will be 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 will be 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 has been received from the I²C host, INTFLAG.AMATCH be set to '1' to clear it.

36.6.2.5.3 Receiving and Transmitting Data Packets

After the I²C client has received an address packet, it will respond 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 will be set. After receiving data, the I²C client will send an acknowledge according to CTRLB.ACKACT.

Case 1: Data received

INTFLAG.DRDY is set, and SCL is held low, pending for SW 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) will be set and the I²C client will return to IDLE state.

36.6.2.5.4 High-Speed Mode

When the I²C client is configured in High-speed mode (*Hs*, CTRLA.SPEED=0x2) and CTRLA.SCLSM=1, switching between Full-speed and High-speed modes is automatic. When the client recognizes a START followed by a host code transmission and a NACK, it automatically switches to High-speed mode and sets the High-speed status bit (STATUS.HS). The client will then remain in High-speed mode until a STOP is received.

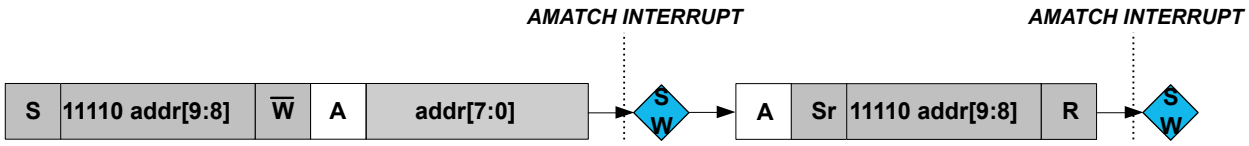
36.6.2.5.5 10-Bit Addressing

When 10-bit addressing is enabled (ADDR.TENBITEN=1), the two address bytes following a START will be checked against the 10-bit client address recognition. The first byte of the address will always be acknowledged, and the second byte will raise the address Interrupt flag, see 10-bit Addressing.

If the transaction is a write, then the 10-bit address will be followed by *N* data bytes.

If the operation is a read, the 10-bit address will be followed by a repeated START and reception of '11110 `ADDR[9:8]` 1', and the second address interrupt will be received with the DIR bit set. The client matches on the second address as it was addressed by the previous 10-bit address.

Figure 36-12. 10-bit Addressing



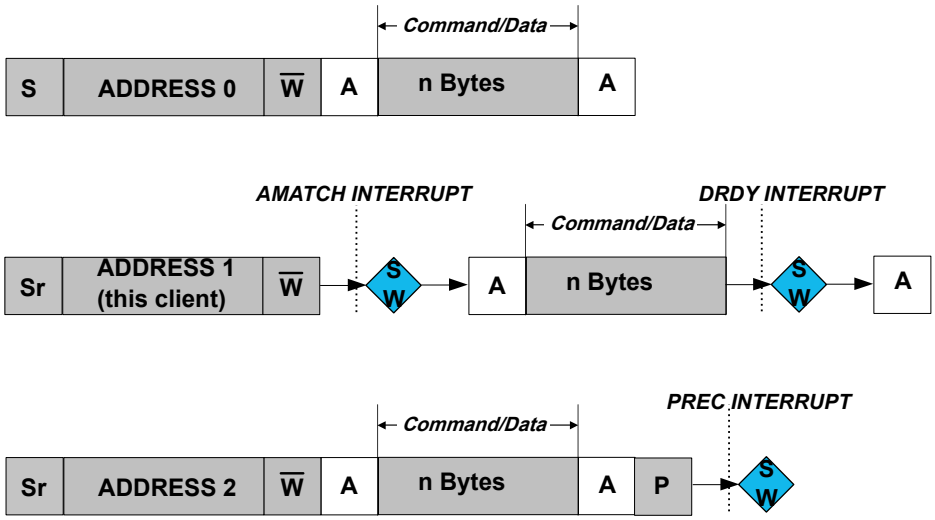
36.6.2.5.6 PMBus Group Command

When the PMBus Group Command bit in the CTRLB register is set (CTRLB.GCMD=1) and 7-bit addressing is used, INTFLAG.PREC will be set if the client has been addressed since the last STOP condition. When CTRLB.GCMD=0, a STOP condition without address match will not be 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 shows 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 36-13. PMBus Group Command Example



36.6.3 Additional Features

36.6.3.1 SMBus

The I^2C includes three hardware SCL low time-outs which 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 32.768 kHz oscillator. The I^2C interface also allows for a SMBus compatible SDA hold time.

- $T_{TIMEOUT}$: SCL low time of 25..35ms – 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.

The SMBus-compatible logic levels are enabled by writting the SMBus Input Buffer Enable bit in Control A register (CTRLA.SMBUFEN).

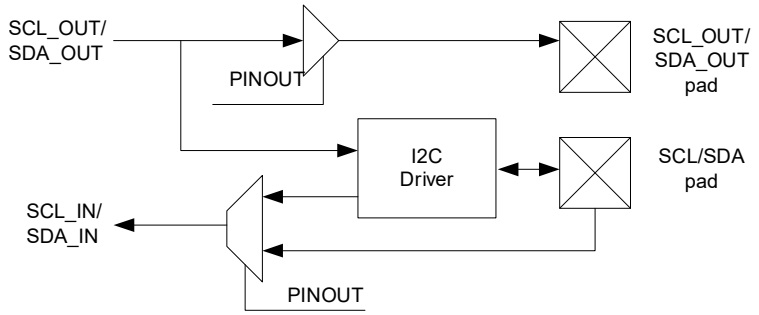
36.6.3.2 Smart Mode

The I^2C interface has a smart mode that simplifies application code and minimizes the user interaction needed to adhere to the I^2C 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.

36.6.3.3 4-Wire Mode

Writing a '1' to the Pin Usage bit in the Control A register (CTRLA.PINOUT) will enable 4-wire mode operation. In this mode, the internal I^2C tri-state drivers are bypassed, and an external I^2C compliant tri-state driver is needed when connecting to an I^2C bus.

Figure 36-14. I²C Pad Interface



36.6.3.4 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.

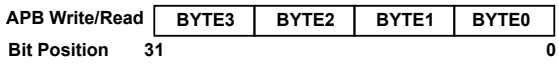
36.6.3.5 32-bit Extension

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 transaction to/from 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 data transfer begins.

The following figure shows the order of transmit and receive when using 32-bit mode. Bytes are transmitted or received and stored in order from 0 to 3.

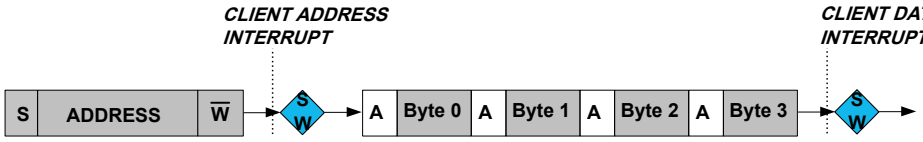
Figure 36-15. 32-bit Extension Byte Ordering



32-bit Extension Client Operation

The following figure shows 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 36-16. 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 will be automatically NACKed. On address recognition, the internal length counter is reset in preparation for the incoming frame.

High Speed transactions start with a Full Speed Host Code. When a Host Code is detected, no data is received and the next expected operation is a repeated start. For this reason, the length is not counted after a Host Code is received. In this case, no Length Error (STATUS.LENERR) is registered, regardless of the LENGTH.LENEN setting.

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, therefore the last byte in a 32-bit word will be NACKed.

Writing to the LENGTH register while a frame is in progress will produce unpredictable results. If LENGTH.LENEN is not set and a frame is not a multiple of 4 Bytes, the remainder will be lost.

32-bit Extension Host Operation

When using the I²C 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 SCL clock stretch mode is enabled (CTRLA.SCLSM = 1) and the transaction is a host read, 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 set CTRLB.ACKACT = 1 in the final interrupt, so that the last byte in a 32-bit word will be NACKed.

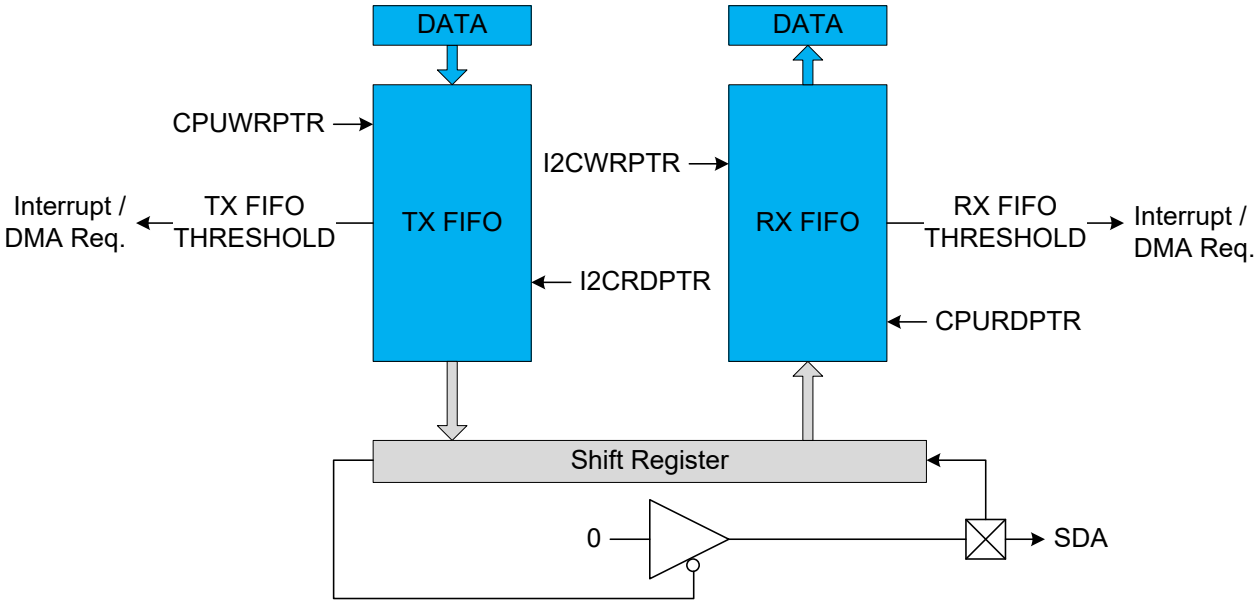
If a NACK is received by the client for a host write transaction before ADDR.LEN bytes, a STOP will be automatically generated, and the length error (STATUS.LENERR) is raised along with the INTFLAG.ERROR interrupt.

36.6.3.6 FIFO Operation

For better system bus utilization, the I²C embeds up to 16-bytes FIFO capability. The receive / transmit buffer is considered to have the FIFO mode enabled when the FIFOEN bit in CTRLC register is set (CTRLC.FIFOEN = 1). By default, the FIFO can act as a 16-by-8-bit array, or as a 4-by-32-bit array, depending on the setting of the CTRLC.DATA32B bit.

The hardware around this array implements four pointers, called the CPU Write Pointer (CPUWRPTR), the CPU Read Pointer (CPURDPTR), the I²C Write pointer (I2CWRPTR) and the I²C Read pointer (I2CRDPTR). All of these pointers reset to '0'. The CPUWRPTR and CPURDPTR pointers are native to the CPU clock domain, while the I2CWRPTR and I2CRDPTR are native to the I²C domain. The location pointed to by the CPUWRPTR is the current TX FIFO. The location pointed to by the CPURDPTR becomes the current RX FIFO. Writes to DATA register by the CPU will point to TX FIFO. Reads to DATA register by the CPU will point to RX FIFO. The location pointed to by the I2CWRPTR / I2CRDPTR is logically the current shift register.

Figure 36-17. FIFO Overview



When using the I²C 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 if the 32-bit extension is enabled (CTRLC.DATA32B).

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 the SCL clock is stretched as long as the FIFO is empty in host read mode.

The FIFO threshold settings allow (CTRLC.TXTRHOLD, CTRLC.RXTRHOLD) allow flexible interrupt, DMA trigger and bus condition generations, as described bellow.

The FIFO is fully accessible if the SERCOM is halted, by writing the corresponding CPU FIFO pointer in the FIFOPTR register. The RX or TX FIFO can be individually cleared, by setting the respective FIFO Clear bit in the Control B register (CTRLB.FIFOCLR). The FIFO Clear must be written before data transfer begins. Writing the FIFO Clear bits while a frame is in progress will produce unpredictable results.

In Client mode, when the FIFO is enabled (CTRLC.FIFOEN = 1), the data can be preloaded into the FIFO prior to the address reception.

In Host mode, when the FIFO is enabled (CTRLC.FIFOEN = 1), the data can be preloaded into the FIFO prior to the address transmission.

36.6.3.6.1 Hardware Actions in Host Mode

Table 36-3. Interrupts Request Conditions for Valid SERCOM I²C Host Configurations

Direction	CTRLB.SMEN	CTRLC.DATA32B	LENGTH.LENEN	Action
Host Write	0	0	0	• INTFLAG.TXFE = 1 if TX FIFO is empty • INTFLAG.TXFE = 1 if TX FIFO threshold reached • INTFLAG.MB = 1 if TX FIFO is empty and SCL hold
	0	1	0	
	0	1	1	• INTFLAG.TXFE = 1 if TX FIFO is empty • INTFLAG.TXFE = 1 if TX FIFO threshold reached • INTFLAG.MB = 1 if TX FIFO is empty and SCL hold, or length transaction is completed
	1	0	0	• INTFLAG.TXFE = 1 if TX FIFO is empty • INTFLAG.TXFE = 1 if TX FIFO threshold reached • INTFLAG.MB = 1 if TX FIFO is empty and SCL hold, or length transaction is completed
	1	1	0	
	1	1	1	
Host Read	0	0	0	• INTFLAG.SB = 1 if RX FIFO is full • INTFLAG.RXFE = 1 if RX FIFO threshold reached or length transaction is completed
	0	1	0	
	0	1	1	
	1	0	0	
	1	1	0	
	1	1	1	

Table 36-4. Bus Actions for Valid SERCOM I²C Host Configurations

Direction	CTRLB.SMEN	CTRLC.DATA32B	LENGTH.LENEN	Actions
Host Write	0	0	0	• SCL hold if TX FIFO is empty
	0	1	0	
	0	1	1	• SCL hold if TX FIFO is empty and length transaction not completed • Issue STOP when transaction is completed
	1	0	0	• SCL hold if TX FIFO is empty, when no automatic stop is sent • STOP is sent on SW decision
	1	1	0	• SCL hold if TX FIFO is empty, when no automatic stop is sent • STOP is sent on SW decision
	1	1	1	• SCL hold if TX FIFO is empty and length transaction is not completed • Issue STOP when transaction is completed
Host Read	0	0	0	• SCL hold if RX FIFO is full
	0	1	0	• SCL hold if RX FIFO is full
	0	1	1	• SCL stretched if RX FIFO is full • ACK/NACK last frame byte, depending on Acknowledge Action (CTRLB.ACKACT) • ACK all other bytes
	1	0	0	• SCL stretched if Rx FIFO is full
	1	1	0	
	1	1	1	• SCL stretched if RX FIFO is full • ACK/NACK last frame byte, depending on Acknowledge Action (CTRLB.ACKACT) • ACK all other bytes

36.6.3.6.2 Hardware Actions in Client Mode

Table 36-5. Interrupt Request Conditions for Valid SERCOM I²C Client Configurations

Direction	CTRLB.SMEN	CTRLC.DATA32B	LENGTH.LENEN	Condition
Host Write	0	0	0	• INTFLAG.DRDY = 1 if Rx FIFO is full • INTFLAG.RXFF = 1 if Rx FIFO threshold is reached or length transaction is completed
	0	1	0	
	0	1	1	
	1	0	0	
	1	1	0	
	1	1	1	
Host Read	0	0	0	• INTFLAG.DRDY = 1 if Tx FIFO is empty and SCL hold • INTFLAG.TXFE = 1 if Tx FIFO is empty or Tx FIFO threshold is reached
	0	1	0	
	0	1	1	
	1	0	0	
	1	1	0	
	1	1	1	

Table 36-6. Bus Actions for Valid SERCOM I²C Client Configurations

Direction	CTRLB.SMEN	CTRLC.DATA32B	LENGTH.LENEN	Actions
Host Write	0	0	0	• Byte mode operation • SCL stretched if RX FIFO is full
	0	1	0	• 32-bit mode operation • SCL stretched if RX FIFO is full • ACK/NACK each 4th byte, depending on Acknowledge Action (CTRLB.ACKACT) • ACK all other bytes
	0	1	1	• 32-bit mode operation with length control • SCL stretched if RX FIFO is full • ACK/NACK last byte of the frame, depending on Acknowledge Action (CTRLB.ACKACT) • ACK all other bytes
	1	0	0	• SCL stretched if RX FIFO is full • ACK all bytes received
	1	1	0	• 32-bit mode operation • SCL stretched if RX FIFO is full • ACK/NACK each 4th byte, depending on Acknowledge Action (CTRLB.ACKACT) • ACK all other bytes
	1	1	1	• 32-bit mode operation with length control • SCL stretched if RX FIFO is full • ACK/NACK last byte of the frame, depending on Acknowledge Action (CTRLB.ACKACT) • ACK all other bytes
Host Read	0	0	0	• SCL stretched if TX FIFO is empty
	0	1	0	• SCL stretched if TX FIFO is empty
	0	1	1	• SCL stretched if TX FIFO is empty and length transaction is not completed
	1	0	0	• SCL stretched if TX FIFO is empty
	1	1	0	• SCL stretched if TX FIFO is empty
	1	1	1	• SCL stretched if TX FIFO is empty and length transaction is not completed

36.6.4 DMA, Interrupts and Events

This chapter provides DMA and interrupt conditions when the optional FIFO is disabled. For details when the FIFO is enabled, refer to FIFO Support.

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 meet. 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 I²C is reset. See the 36.7.6. INTFLAG (Client) or 36.8.7. INTFLAG (Host) register for details on how to clear interrupt flags.

Table 36-7. Module Request for SERCOM I²C 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	
TX FIFO Empty (TXFE)		Yes	
RX FIFO Full (RXFF)		Yes	
Error (ERROR)		Yes	

Table 36-8. 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	
TX FIFO Empty (TXFE)		Yes	
RX FIFO Full (RXFF)		Yes	
Error (ERROR)		Yes	

36.6.4.1 DMA Operation

Smart mode must be enabled for DMA operation in the Control B register by writing CTRLB.SMEN=1.

36.6.4.1.1 Client DMA

When using the I²C client with DMA, an address match will cause the address interrupt flag (INTFLAG.ADDRMATCH) to be raised. After the interrupt has been serviced, data transfer will be performed through DMA.

The I²C client generates the following requests:

- 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 is 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.

36.6.4.1.2 Host DMA

When using the I^2C 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 to 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 will be automatically generated and the length error (STATUS.LENERR) will be raised along with the INTFLAG.ERROR interrupt.

The I^2C host generates the following requests:

- Read data received (RX): If the FIFO is disabled, the request is set when host read data is 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 is 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.

36.6.4.2 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)
- RX FIFO Full (RXFF)
- TX FIFO Empty (TXFE)
- 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)
- RX FIFO Full (RXFF)
- TX FIFO Empty (TXFE)
- 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 meet. 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 active until the interrupt flag is cleared, the interrupt is disabled or the I^2C is reset. See the INTFLAG register for details on how to clear interrupt flags.

The value of INTFLAG indicates which interrupt is executed. Note that interrupts must be globally enabled for interrupt requests. Refer to Nested Vector Interrupt Controller for details.

36.6.5 Sleep Mode Operation

I^2C Host Operation

The generic clock (GCLK_SERCOMx_CORE) will continue to run in Idle Sleep mode. If the Run In Standby bit in the Control A register (CTRLA.RUNSTDBY) is '1', the GLK_SERCOMx_CORE will also run in standby Sleep mode. Any interrupt can wake up the device.

If CTRLA.RUNSTDBY=0, the GLK_SERCOMx_CORE will be disabled after any ongoing transaction is finished. Any interrupt can wake up the device.

I²C Client Operation

Writing CTRLA.RUNSTDBY=1 will allow the Address Match interrupt to wake up the device.

When CTRLA.RUNSTDBY=0, all receptions will be dropped.

36.6.6 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 - refer to the Debug Control (DBGCTRL) register for details.

36.6.7 Synchronization

Some registers (or bit fields within a register) require synchronization when read and/or written.

Synchronization is denoted by the "Read-Synchronized" (or "Read-Synchronized Bits”) and/or "Write-Synchronized" (or "Write-Synchronized Bits”) property in each individual register description.

For more details, refer to Register Synchronization.

36.7 Register Summary - I2C Client

Refer to the Registers Description section for more details on register properties and access permissions.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRLA	31:24		LOWTOUTEN			SCLSM		SPEED[1:0]	
		23:16	SEXTTOEN		SDAHOLD[1:0]					PINOUT
		15:8								
		7:0	RUNSTDBY			MODE[2:0]			ENABLE	SWRST
0x04	CTRLB	31:24								
		23:16	FIFOCLR[1:0]					ACKACT	CMD[1:0]	
		15:8	AMODE[1:0]					AACKEN	GCMD	SMEN
		7:0								
0x08	CTRLC	31:24	TXTRHOLD[1:0]		RXTRHOLD[1:0]		FIFOEN			DATA32B
		23:16								
		15:8								
		7:0					SDASETUP[3:0]			
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	15:8					LENERR	HS	SEXTTOUT	
		7:0	CLKHOLD	LOWTOUT		SR	DIR	RXNACK	COLL	BUSERR
0x1C	SYNCBUSY	31:24								
		23:16								
		15:8								
		7:0				LENGTH		SYSOP	ENABLE	SWRST
0x20 ... 0x21	Reserved									
0x22	LENGTH	15:8								LENEN
		7:0	LEN[7:0]							
0x24	ADDR	31:24						ADDRMASK[9:7]		
		23:16	ADDRMASK[6:0]							
		15:8	TENBITEN					ADDR[9:7]		
		7:0	ADDR[6:0]							GENCEN
0x28	DATA	31:24	DATA[31:24]							
		23:16	DATA[23:16]							
		15:8	DATA[15:8]							
		7:0	DATA[7:0]							
0x2C ... 0x33	Reserved									
0x34	FIFOSPACE	15:8				RXSPACE[4:0]				
		7:0				TXSPACE[4:0]				
0x36	FIFOPTR	15:8					CPURDPTR[3:0]			
		7:0					CPUWRPTR[3:0]			

36.7.1 Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected Bits, Write-Synchronized Bits

Bit	31	30	29	28	27	26	25	24
		LOWTOUTEN			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		SDAHOLD[1:0]					PINOUT
Access	R/W		R/W	R/W				R/W
Reset	0		0	0				0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	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 – LOWTOUTEN SCL Low Time-Out Enable
This bit enables the SCL low time-out. If SCL is held low for 25ms-35ms, the client will release its clock hold, if enabled, and reset the internal state machine. Any interrupt flags set at the time of time-out will remain set.
Note: This bit is enable-protected. 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 will be stretched for software interaction.
Note: This bit is enable-protected. This bit is not synchronized.

Value	Description
0	SCL stretch according to Figure 36-10
1	SCL stretch only after ACK bit according to Figure 36-11

Bits 25:24 – SPEED[1:0] Transfer Speed
These bits define bus speed.
Note: This bit field is enable-protected. This bit field is not synchronized.

Value	Name	Description
0x0	SM	Standard-mode (Sm) up to 100 kHz and Fast-mode (Fm) up to 400 kHz
0x1	FMP	Fast-mode Plus (Fm+) up to 1 MHz
0x2	HS	High-speed mode (Hs-mode) up to 3.4 MHz
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 25ms from the initial START to a STOP, the client will release its clock hold if enabled and reset the internal state machine. Any

interrupt flags set at the time of time-out will remain set. If the address was recognized, PREC will be set when a STOP is received.

Note: This bit is enable-protected. This bit is not synchronized.

Value	Description
0	Time-out disabled
1	Time-out enabled

Bits 21:20 – SDAHOLD[1:0] SDA Hold Time

These bits define the SDA hold time with respect to the negative edge of SCL.

Note: This bit field is enable-protected. This bit field is not synchronized.

Value	Name	Description
0x0	DIS	Disabled
0x1	75NS	50-100ns hold time
0x2	450NS	300-600ns hold time
0x3	600NS	400-800ns hold time

Bit 16 – PINOUT Pin Usage

This bit sets the pin usage to either two- or four-wire operation:

Note: This bit is enable-protected. This bit is not synchronized.

Value	Description
0	4-wire operation disabled
1	4-wire operation enabled

Bit 7 – RUNSTDBY Run in Standby

This bit defines the functionality in standby sleep mode.

Note: This bit is enable-protected. 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.

Note: This bit field is enable-protected. This bit field is not synchronized.

Bit 1 – ENABLE Enable

Notes:

1. This bit is write-synchronized: SYNCBUSY.ENABLE must be checked to ensure the CTRLA.ENABLE synchronization is complete.
2. 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 will be disabled.

Writing '1' to CTRLA.SWRST will always take precedence, meaning that all other writes in the same write-operation will be discarded. Any register write access during the ongoing reset will result in an APB error. Reading any register will return the reset value of the register.

Notes:

1. When the CTRLA.SWRST is written, the user must poll SYNCBUSY.SWRST bit to know when the reset operation is complete.
2. During a SWRST, access to registers/bits without SWRST are disallowed until SYNCBUSY.SWRST is cleared by hardware.
3. This bit is not enable protected.

Value	Description
0	There is no reset operation ongoing.
1	The reset operation is ongoing.

36.7.2 Control B

Name: CTRLB
Offset: 0x04
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected Bits, Write-Synchronized Bits

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

Note: This bit field is write-synchronized: SYNCBUSY.SYSOP must be checked to ensure the CTRLB.FIFOCLR synchronization is complete.

Note: This bit field is not enable-protected.

Value	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 shall not be updated more than once between each peripheral interrupts request.
Note: This bit is not enable-protected. This bit is not synchronized.

Value	Description
0	Send ACK
1	Send NACK

Bits 17:16 – CMD[1:0] Command

This bit field triggers the client operation as the below. The CMD bits are strobe bits, and always read as zero.
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.
Note: This bit field is not enable-protected. This bit field is not synchronized.

Table 36-9. Command Description

Value	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.

Note: This bit is enable-protected. This bit is not synchronized.

Value	Name	Description
0x0	MASK	The slave responds to the address written in ADDR.ADDR masked by the value in ADDR.ADDRMASK. See *SERCOM – Serial Communication Interface* for additional information.
0x1	2ADDRS	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.

Note: This bit is enable-protected. This bit is not synchronized.

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 Recived interrupt flag (INTFLAG.PREC) will be set when a STOP condition is detected if the client has been addressed since the last STOP condition on the bus.

Note: This bit is enable-protected. This bit is not synchronized.

Value	Description
0	Group command is disabled.
1	Group command is enabled.

Bit 8 – SMEN Smart Mode Enable

When smart mode is enabled, data is acknowledged automatically when DATA.DATA is read.

Note: This bit is enable-protected. This bit is not synchronized.

Value	Description
0	Smart mode is disabled.
1	Smart mode is enabled.

36.7.3 Control C

Name: CTRLC
Offset: 0x08
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
	TXTRHOLD[1:0]		RXTRHOLD[1:0]		FIFOEN			DATA32B
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
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

Bits 31:30 – TXTRHOLD[1:0] Transmit FIFO Threshold
These bits define the threshold for generating the Data Register Empty interrupt and DMA TX trigger.

Value	Name	Description
0	DEFAULT	Interrupt and DMA triggers can be generated as long as the FIFO is not full.
1	HALF	Interrupt and DMA triggers are generated when half FIFO space is free.
2	EMPTY	Interrupt and DMA triggers are generated when the FIFO is empty.
3	-	Reserved

Bits 29:28 – RXTRHOLD[1:0] Receive FIFO Threshold
These bits define the threshold for generating the RX Complete interrupt and DMA RX trigger.

Value	Name	Description
0	DEFAULT	Interrupt and DMA triggers can be generated when a DATA is present in the FIFO.
1	HALF	Interrupt and DMA triggers can be generated only when the FIFO is half-full.
2	FULL	Interrupt and DMA triggers can be generated only when the FIFO is full.
3	-	Reserved

Bit 27 – FIFOEN FIFO Enable
This bit enables the FIFO operation.

Value	Description
0	FIFO operation is disabled
1	FIFO operation is enabled

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:
$t_{SU:DAT} = (\text{CLK\_SERCOMx} \times \text{APB period}) \times (6 + 16 \times \text{SDASETUP})$

36.7.4 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 will clear 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 will clear 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 will clear 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 will clear 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 will clear 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 will clear 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.

36.7.5 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 will set 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 will set 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 will set 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 will set the Data Ready 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 will set 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 will set 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.

36.7.6 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 or reception 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 PM Bus 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.

36.7.7 Status

Name: STATUS
Offset: 0x1A
Reset: 0x0000
Property: -

Bit	15	14	13	12	11	10	9	8
					LENERR	HS	SEXTTOUT	
Access					R/W	R/W	R/W	
Reset					0	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 10 – HS High-speed
This bit is set if the client detects a START followed by a Host Code transmission.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the status. However, this flag is automatically cleared when a STOP is received.

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 should 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 I^2C client was not able to transmit a high data or NACK bit, the I^2C 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 should be treated as a bus error.
Note that this status will not trigger any interrupt, and should 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 I^2C 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 '1' to this bit will clear the status.
Writing a '0' to this bit has no effect.

Value	Description
0	No bus error detected.
1	Bus error detected.

36.7.8 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. 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.

Value	Description
0	SWRST synchronization is not busy.

Value	Description
1	SWRST synchronization is busy.

36.7.9 Length

Name: LENGTH
Offset: 0x22
Reset: 0x0000
Property: Write-Synchronized

Note: This register is write-synchronized: SYNCBUSY.LENGTH must be checked to ensure the LENGTH register synchronization is complete.

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.

36.7.10 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
	TENBITEN					ADDR[9:7]		
Access	R/W					R/W	R/W	R/W
Reset	0					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.

Bit 15 – TENBITEN Ten Bit Addressing Enable

Value	Description
0	10-bit address recognition disabled.
1	10-bit address recognition enabled.

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 using 10-bit addressing (ADDR.TENBITEN=1), the client address is represented by ADDR[9:0]
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.

36.7.11 Data

Name: DATA
Offset: 0x28
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

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.

Writing or reading DATA.DATA when not in smart mode does not require synchronization.

36.7.12 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).

36.7.13 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.

36.8 Register Summary - I2C Host

Refer to the Registers Description section for more details on register properties and access permissions.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRLA	31:24		LOWTOUTEN	INACTOUT[1:0]		SCLSM		SPEED[1:0]	
		23:16	SEXTTOEN	MEXTTOEN	SDAHOLD[1:0]					PINOUT
		15:8								
		7:0	RUNSTDBY			MODE[2:0]			ENABLE	SWRST
0x04	CTRLB	31:24								
		23:16	FIFOCLR[1:0]					ACKACT	CMD[1:0]	
		15:8							QCEN	SMEN
		7:0								
0x08	CTRLC	31:24	TXTRHOLD[1:0]		RXTRHOLD[1:0]		FIFOEN			DATA32B
		23:16								
		15:8								
		7:0								
0x0C	BAUD	31:24	HSBAUDLOW[7:0]							
		23:16	HSBAUD[7:0]							
		15:8	BAUDLOW[7:0]							
		7:0	BAUD[7:0]							
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	15:8						LENERR	SEXTTOUT	MEXTTOUT
		7:0	CLKHOLD	LOWTOUT	BUSSTATE[1:0]			RXNACK	ARBLOST	BUSERR
0x1C	SYNCBUSY	31:24								
		23:16								
		15:8								
		7:0						SYSOP	ENABLE	SWRST
0x20 ... 0x23	Reserved									
0x24	ADDR	31:24								
		23:16	LEN[7:0]							
		15:8	TENBITEN	HS	LENEN			ADDR[10:8]		
		7:0	ADDR[7:0]							
0x28	DATA	31:24	DATA[31:24]							
		23:16	DATA[23:16]							
		15:8	DATA[15:8]							
		7:0	DATA[7:0]							
0x2C ... 0x2F	Reserved									
0x30	DBGCTRL	7:0								DBGSTOP
0x31 ... 0x33	Reserved									
0x34	FIFOSPACE	15:8				RXSPACE[4:0]				
		7:0				TXSPACE[4:0]				
0x36	FIFOPTR	15:8					CPURDPTR[3:0]			
		7:0					CPUWRPTR[3:0]			

36.8.1 Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected Bits, Write-Synchronized Bits

Bit	31	30	29	28	27	26	25	24
		LOWTOUTEN	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	SDAHOLD[1:0]					PINOUT
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
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 – LOWTOUTEN SCL Low Time-Out Enable
This bit enables the SCL low time-out. If SCL is held low for 25ms-35ms, the host will release its clock hold, if enabled, and complete the current transaction. A stop condition will automatically be transmitted. INTFLAG.SB or INTFLAG.MB will be set as normal, but the clock hold will be released. The STATUS.LOWTOUT and STATUS.BUSERR status bits will be set.
Note: This bit is enable-protected. 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 will be set to idle. An inactive bus arise 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 set-up.
Calculated time-out periods are based on a 100kHz baud rate.
Note: This bit field is enable-protected. This bit field is not synchronized.

Value	Name	Description
0x0	DIS	Disabled
0x1	55US	5-6 SCL cycle time-out (50-60µs)
0x2	105US	10-11 SCL cycle time-out (100-110µs)
0x3	205US	20-21 SCL cycle time-out (200-210µs)

Bit 27 – SCLSM SCL Clock Stretch Mode
This bit controls when SCL will be stretched for software interaction.
Note: This bit is enable-protected. This bit is not synchronized.

Value	Description
0	SCL stretch according to Figure 36-5.

Value	Description
1	SCL stretch only after ACK bit, Figure 36-6.

Bits 25:24 – SPEED[1:0] Transfer Speed

These bits define bus speed.

Note: This bit field is enable-protected. This bit field is not synchronized.

Value	Name	Description
0x0	SM	Standard-mode (Sm) up to 100 kHz and Fast-mode (Fm) up to 400 kHz
0x1	FMP	Fast-mode Plus (Fm+) up to 1 MHz
0x2	HS	High-speed mode (Hs-mode) up to 3.4 MHz
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 25ms from the initial START to a STOP, the host will release its clock hold if enabled, and complete the current transaction. A STOP will automatically be transmitted.

SB or MB will be set as normal, but CLKHOLD will be release. The MEXTTOUT and BUSERR status bits will be set.

Note: This bit is enable-protected. 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 10ms from START-to-ACK, ACK-to-ACK, or ACK-to-STOP the host will release its clock hold if enabled, and complete the current transaction. A STOP will automatically be transmitted.

SB or MB will be set as normal, but CLKHOLD will be released. The MEXTTOUT and BUSERR status bits will be set.

Note: This bit is enable-protected. This bit is not synchronized.

Value	Description
0	Time-out disabled
1	Time-out enabled

Bits 21:20 – SDAHOLD[1:0] SDA Hold Time

These bits define the SDA hold time with respect to the negative edge of SCL.

Note: This bit field is enable-protected. This bit field is not synchronized.

Value	Name	Description
0x0	DIS	Disabled
0x1	75NS	50-100ns hold time
0x2	450NS	300-600ns hold time
0x3	600NS	400-800ns hold time

Bit 16 – PINOUT Pin Usage

This bit set the pin usage to either two- or four-wire operation:

Note: This bit is enable-protected. This bit is not synchronized.

Value	Description
0	4-wire operation disabled.
1	4-wire operation enabled.

Bit 7 – RUNSTDBY Run in Standby

This bit defines the functionality in standby sleep mode.

Note: This bit is enable-protected. This bit is not synchronized.

Value	Description
0	GCLK_SERCOMx_CORE is disabled and the I^2C host will not operate in 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.

Note: This bit field is enable-protected. This bit field is not synchronized.

Bit 1 – ENABLE Enable

Notes:

1. This bit is write-synchronized: SYNCBUSY.ENABLE must be checked to ensure the CTRLA.ENABLE synchronization is complete.
2. 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 will be disabled.

Writing '1' to CTRLA.SWRST will always take precedence, meaning that all other writes in the same write-operation will be discarded. Any register write access during the ongoing reset will result in an APB error. Reading any register will return the reset value of the register.

Notes:

1. When the CTRLA.SWRST is written, the user must poll SYNCBUSY.SWRST bit to know when the reset operation is complete.
2. During a SWRST, access to registers/bits without SWRST are disallowed until SYNCBUSY.SWRST is cleared by hardware.
3. This bit is not enable protected.

Value	Description
0	There is no reset operation ongoing.
1	The reset operation is ongoing.

36.8.2 Control B

Name: CTRLB
Offset: 0x04
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected Bits, Write-Synchronized Bits

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
							QCEN	SMEN
Access							R/W	R/W
Reset							0	0

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bits 23:22 – FIFOCLR[1:0] FIFO Clear

Note: This bit field is write-synchronized: SYNCBUSY.SYSOP must be checked to ensure the CTRLB.FIFOCLR synchronization is complete.

Note: This bit field is not enable-protected.

Value	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 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.

Note: This bit is not enable-protected. This bit is not 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 zero. 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 will be 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 will be 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 will trigger a repeated start followed by transmission of the new address.
Note: This bit field is write-synchronized: SYNCBUSY.SYSOP must be checked to ensure the CTRLB.CMD synchronization is complete.

Table 36-10. Command Description

Value	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

Note: This bit field is not enable-protected.

Bit 9 – QCEN Quick Command Enable
Note: This bit is enable-protected. This bit is not 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.
Note: This bit is enable-protected. This bit is not synchronized.

Value	Description
0	Smart mode is disabled.
1	Smart mode is enabled.

36.8.3 Control C

Name: CTRLC
Offset: 0x08
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
	TXTRHOLD[1:0]		RXTRHOLD[1:0]		FIFOEN			DATA32B
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
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bits 31:30 – TXTRHOLD[1:0] Transmit FIFO Threshold
These bits define the threshold for generating the Data Register Empty interrupt and DMA TX trigger.

Value	Name	Description
0	DEFAULT	Interrupt and DMA triggers can be generated as long as the FIFO is not full.
1	HALF	Interrupt and DMA triggers are generated when half FIFO space is free.
2	EMPTY	Interrupt and DMA triggers are generated when the FIFO is empty.
3	-	Reserved

Bits 29:28 – RXTRHOLD[1:0] Receive FIFO Threshold
These bits define the threshold for generating the RX Complete interrupt and DMA RX trigger.

Value	Name	Description
0	DEFAULT	Interrupt and DMA triggers can be generated when a DATA is present in the FIFO.
1	HALF	Interrupt and DMA triggers can be generated only when the FIFO is half-full.
2	FULL	Interrupt and DMA triggers can be generated only when the FIFO is full.
3	-	Reserved

Bit 27 – FIFOEN FIFO Enable
This bit enables the FIFO operation.

Value	Description
0	FIFO operation is disabled
1	FIFO operation is enabled

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

36.8.4 Baud Rate

Name: BAUD
Offset: 0x0C
Reset: 0x0000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
	HSBAUDLOW[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
	HSBAUD[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
	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 31:24 – HSBAUDLOW[7:0] High Speed Host Baud Rate Low

HSBAUDLOW non-zero: HSBAUDLOW indicates the SCL low time in High-speed mode according to

$$\mathrm{HSBAUDLOW} = f_{\mathrm{GCLK}} \cdot T_{\mathrm{LOW}} - 1$$

HSBAUDLOW equal to zero: The HSBAUD register is used to time T_{LOW}, T_{HIGH}, $T_{SU;STO}$, $T_{HD;STA}$ and $T_{SU;STA}$. T_{BUF} is timed by the BAUD register.

Bits 23:16 – HSBAUD[7:0] High Speed Host Baud Rate

This bit field indicates the SCL high time in High-speed mode according to the following formula. When HSBAUDLOW is zero, T_{LOW}, T_{HIGH}, $T_{SU;STO}$, $T_{HD;STA}$ and $T_{SU;STA}$ are derived using this formula. T_{BUF} is timed by the BAUD register.

$$\mathrm{HSBAUD} = f_{\mathrm{GCLK}} \cdot T_{\mathrm{HIGH}} - 1$$

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 information on how to calculate the frequency, see SERCOM 33.6.2.3. Clock Generation – Baud-Rate Generator.

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 information on how to calculate the frequency, see SERCOM 33.6.2.3. Clock Generation – Baud-Rate Generator.

36.8.5 Interrupt Enable Clear

Name: INTENCLR
Offset: 0x14
Reset: 0x00
Property: PAC Write-Protection

This register allows the user to disable an interrupt without 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		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 will clear 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 will clear 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 will clear 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 will clear 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 will clear 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.

36.8.6 Interrupt Enable Set

Name: INTENSET
Offset: 0x16
Reset: 0x00
Property: PAC Write-Protection

This register allows the user to enable an interrupt without 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		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 will set 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 will set 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 will set 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 will set 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 will set 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.

36.8.7 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 will 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 will clear the flag.

Bit 4 – RXFF RX FIFO Full
This flag is set when RX FIFO Threshold locations are fullfilled.
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 1 – SB Client on Bus
The Client on Bus flag (SB) is set when a byte is successfully received in host read mode, i.e., 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 will be released and SB will be 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 will clear 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 NACK in 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 will be released and MB will be 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 will clear 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.

36.8.8 Status

Name: STATUS
Offset: 0x1A
Reset: 0x0000
Property: Write-Synchronized Bits

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 have been written by the host.
Writing '1' to this bit location will clear 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 will clear 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 will clear 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 should 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 will clear 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 I^2C 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 will set SYNCBUSY.SYSOP.

Value	Name	Description
0x0	UNKNOWN	The bus state is unknown to the I²C host and will wait 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) will be set when STATUS.ARBLOST is set.
Writing the ADDR.ADDR register will automatically clear STATUS.ARBLOST.
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear it.
This bit is not write-synchronized.

Bit 0 – BUSERR Bus Error

This bit 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 BUSERR.
If the I²C host is the bus owner at the time a bus error occurs, STATUS.ARBLOST and INTFLAG.MB will be set in addition to BUSERR.
Writing the ADDR.ADDR register will automatically clear the BUSERR flag.
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear it.
This bit is not write-synchronized.

36.8.9 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
						SYSOP	ENABLE	SWRST
Access						R	R	R
Reset						0	0	0

Bit 2 – SYSOP System Operation Synchronization Busy

Writing CTRLB.CMD or CTRLB.FIFOCLR, 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.

Value	Description
0	SWRST synchronization is not busy.
1	SWRST synchronization is busy.

36.8.10 Address

Name: ADDR
Offset: 0x24
Reset: 0x0000
Property: Write-Synchronized Bits

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	HS	LENEN			ADDR[10:8]		
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
	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 to 255 bytes. The Transfer Length Enable (LENEN) bit must be written to '1' in order 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 14 – HS High Speed
This bit enables High-speed mode for the current transfer from repeated START to STOP. This bit can be written simultaneously with ADDR for a high speed transfer.

Value	Description
0	High-speed transfer disabled.
1	High-speed transfer 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 will depend on the bus state:
UNKNOWN: INTFLAG.MB and STATUS.BUSERR are set, and the operation is terminated.
BUSY: The I^2C host will await further operation until the bus becomes IDLE.
IDLE: The I^2C host will issue 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 will be 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.
STATUS.BUSERR, STATUS.ARBLOST, INTFLAG.MB and INTFLAG.SB will be 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.
Note: This bit field is write-synchronized: SYNCBUSY.SYSOP must be checked to ensure the ADDR.ADDR synchronization is complete.

36.8.11 Data

Name: DATA
Offset: 0x28
Reset: 0x00000000
Property: Write-Synchronized, Read-Synchronized

Note: This register is read- and write-synchronized: SYNCBUSY.SYSOP must be checked to ensure the DATA register synchronization is complete.

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.

Writing or reading DATA.DATA when not in smart mode does not require synchronization.

36.8.12 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.

36.8.13 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).

36.8.14 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.

37. Inter-IC Sound Controller (I^2S)

37.1 Overview

The Inter-IC Sound Controller (I^2S) provides bidirectional, synchronous and digital audio link with external audio devices.

This controller is compliant with the Inter-IC Sound (I^2S) bus specification. It supports TDM interface with external multi-slot audio codecs. It also supports PDM interface with external MEMS microphones.

The I^2S consists of two Clock Units, one Transmit Serializer, and one Receive Serializer, that can be enabled separately, to provide Host, Client, or controller modes.

The pins associated with I^2S peripheral are SDO,SDI, FSn, SCKn, and MCKn pins.

Peripheral DMAC channels, separate for each Serializer, allow a continuous high bitrate data transfer without processor intervention to the following:

- Audio codecs in Host, Client, or Controller mode
- Stereo DAC or ADC through dedicated I^2S serial interface
- Multi-slot or multiple stereo DACs or ADCs, using the TDM format
- Mono or stereo MEMS microphones, using the PDM interface

Each Serializer supports using either a single DMAC channel for all data channels, or two separate DMAC channels for different data channels.

The I^2S supports 8-bit and 16-bit compact stereo format. This helps in reducing the required DMA bandwidth by transferring the left and right samples within the same data word.

Usually, an external audio codec or digital signal processor (DSP) requires a clock which is a multiple of the sampling frequency *fs* (for example, 384×*fs*). The I^2S peripheral in Host Mode and Controller mode is capable of outputting an output clock ranging from 16×*fs* to 1024×*fs* on the Host Clock pin (MCKn).

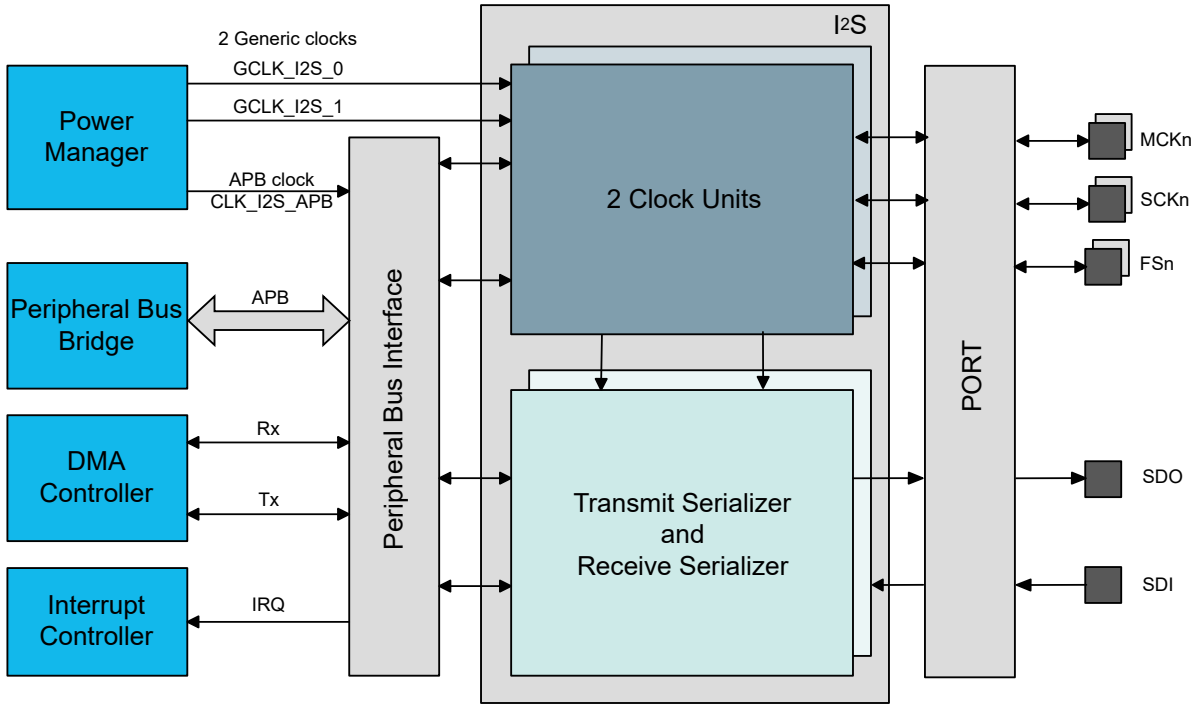
37.2 Features

- Compliant with Inter-IC Sound (I^2S) bus specification
- Supported data formats:
  - 32-bit, 24-bit, 20-bit, 18-bit, 16-bit, and 8-bit mono or stereo format
  - 16-bit and 8-bit compact stereo format, with left and right samples packed in the same word to reduce data transfers
- Supported data frame formats:
  - 2-channel I^2S with Word Select
  - 1- to 8-slot Time Division Multiplexed (TDM) with Frame Sync and individually enabled slots
  - 1- or 2-channel Pulse Density Modulation (PDM) reception for MEMS microphones
  - 1-channel burst transfer with non-periodic Frame Sync
- 2 independent Clock Units handling either the same clock or separate clocks for the Serializers:
  - Suitable for a wide range of sample frequencies *fs*, including 32 kHz, 44.1 kHz, 48 kHz, 88.2 kHz, 96 kHz, and 192 kHz
  - 16×*fs* to 1024×*fs* Host Clock generated for external audio CODECs
- Host, client, and controller modes:
  - **Host:** Data received/transmitted based on internally-generated clocks. Output Serial Clock on SCKn pin, Host Clock on MCKn pin, and Frame Sync Clock on FSn pin
  - **Client:** Data received/transmitted based on external clocks on Serial Clock pin (SCKn) or Host Clock pin (MCKn)
  - **Controller:** Only output internally generated Host clock (MCKn), Serial Clock (SCKn), and Frame Sync Clock (FSn)

- Individual enabling and disabling of Clock Units and Serializers
- DMA interfaces for each Serializer receiver or transmitter to reduce processor overhead:
  - Either one DMA channel for all data slots or
  - One DMA channel per data channel in stereo
- Smart Data Holding register management to avoid data slots mix after overrun or underrun

37.3 Block Diagram

Figure 37-1. I²S Block Diagram



37.4 Signal Description

Table 37-1. Signal Description

Pin Name	Pin Description	Type
MCKn [n=0..1]	Host Clock for Clock Unit n	Input/Output
SCKn [n=0..1]	Serial Clock for Clock Unit n	Input/Output
FSn [n=0..1]	I²S Word Select or TDM Frame Sync for Clock Unit n	Input/Output
SDO	Serial Data Output for Transmit Serializer	Output
SDI	Serial Data Input for Receive Serializer	Input

⚠ CAUTION I/Os for I²S peripheral are grouped into IO sets, listed in their 'IOSET' column in the pinout tables. For this peripheral, it is mandatory to use I/Os that belong to the same IO set. The timings are not guaranteed when IOs from different IO sets are mixed. Refer to the Pinout and Packaging chapter to get IOSET definitions.

37.5 Peripheral Dependencies

Table 37-2. I2S Configuration Summary

Peripheral name	Base address	NVIC IRQ Index	MCLK AHB/APB Clocks	GCLK Peripheral Channel Index (GCLK.PCHCTRL)	PAC Peripheral Identifier Index (PAC.WRCTRL)	DMA Trigger Source Index (DMAC.CHCTRLB)	Power Domain (PM.STDBYCFG)
I2S	0x42004C00	69: RXRDY0-1, TXRDY0-1, RXOR0-1, TXUR0-1	CLK_I2S_APB	33: GCLK_I2S_0 34: GCLK_I2S_1	83	47-48: RX0-1 49-50: TX0-1	PDSW

37.6 Functional Description

37.6.1 Principle of Operation

The I^2S uses three or four communication lines for synchronous data transfer:

- SDO output for Transmit Serializer
- SDI input for Receive Serializer
- SCKn for the serial clock in Clock Unit n (n=0..1)
- FSn for the frame synchronization or I^2S word select, identifying the beginning of each frame
- Optionally, MCKn to output an oversampling clock to an external codec

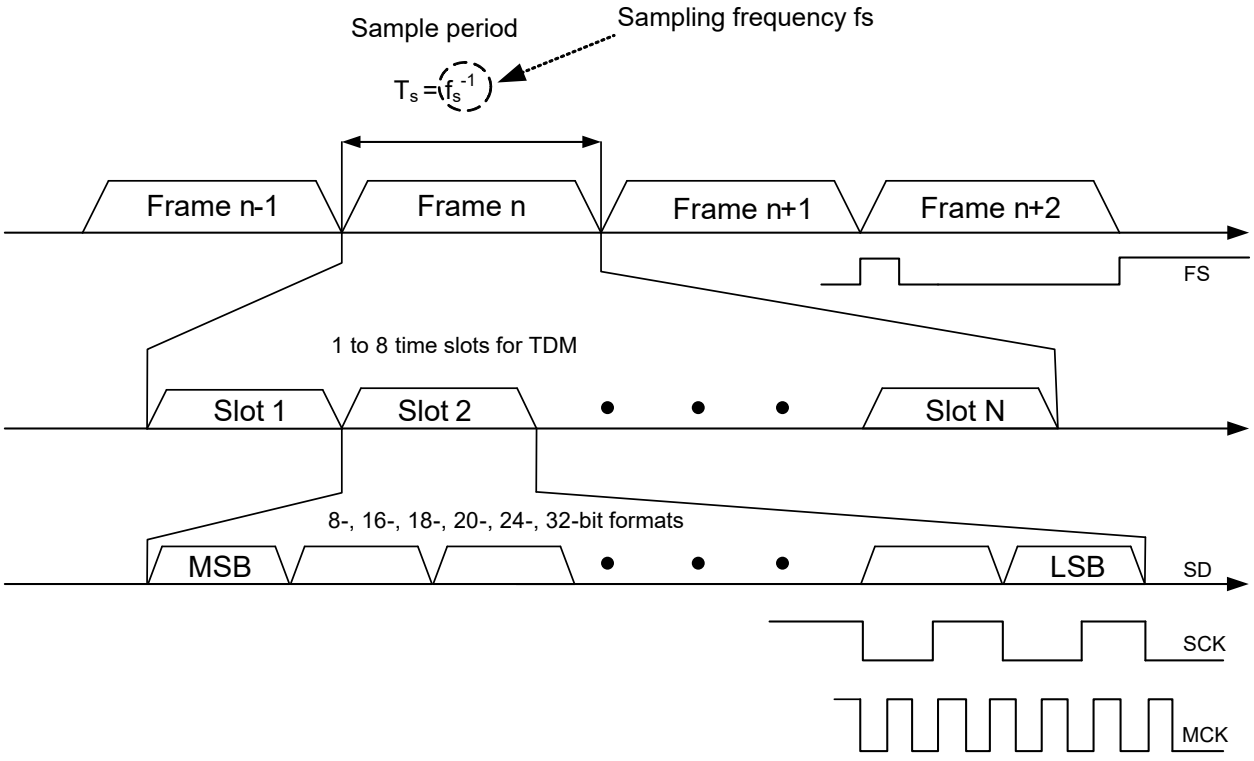
I^2S data transfer is frame based, where a serial frame:

- Starts with the frame synchronization active edge, and
- Consists of 1 to 8 data slots, that are 8-, 16-, 24-, or 32-bit wide.

Each data slot is used to transfer one data sample of 8, 16, 18, 20, 24 or 32 bits.

Frame based data transfer is described in the following figure:

Figure 37-2. Data Format: Frames, Slot, Bits and Clocks



I^2S supports multiple data formats such as:

- 32-, 24-, 20-, 18-, 16-, and 8-bit mono or stereo format
- 16- and 8-bit compact stereo format, with left and right samples packed in the same word to reduce data transfers

In mono format, Transmit mode, data written to the left channel is duplicated to the right output channel. In mono format, Receiver mode, data received from the right channel is ignored and data received from the left channel is duplicated in to the right channel.

In mono format, TDM Transmit mode with more than two slots, data written to the even-numbered slots is duplicated in to the following odd-numbered slot.

In mono format, TDM Receiver mode with more than two slots, data received from the even-numbered slots is duplicated in to the following odd-numbered slot.

Mono format can be enabled by writing a '1' to the MONO bit in the Serializer m Control register (SERCTRLm.MONO).

I^2S support different data frame formats:

- 2-channel I^2S with Word Select
- 1- to 8-slot Time Division Multiplexed (TDM) with Frame Sync and individually enabled slots
- 1- or 2-channel Pulse Density Modulation (PDM) reception for MEMS microphones
- 1-channel burst transfer with non-periodic Frame Sync

In 2 channel I^2S mode, number of slots configured is one or two and successive data words corresponds to left and right channel. Left and right channel are identified by polarity of Word Select signal (FSn signal). Each frame consists of one or two data word(s). In the case of compact stereo format, the number of slots can be one. When 32-bit slot size is used, the number of slots can be two.

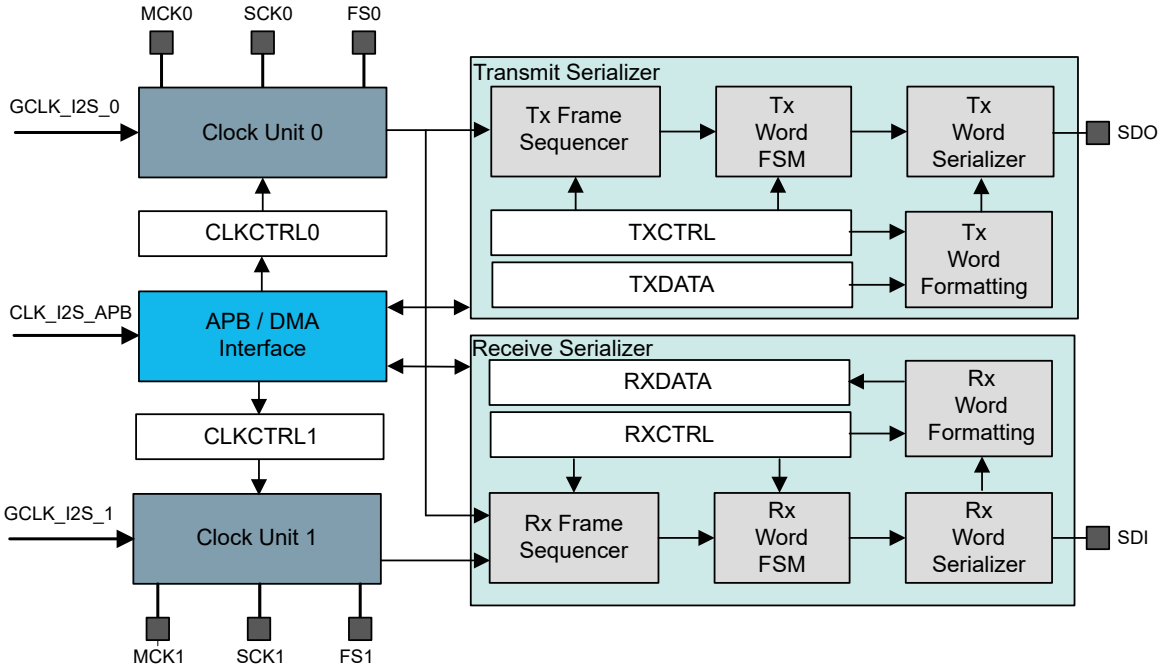
In TDM format, number slots can be configured up to 8 slots. If 4 slots are configured, each frame consists of 4 data words.

In PDM format, continuous 1-bit data samples are available on the SDI line for each SCKn rising and SCKn falling edge as in case of a MEMS microphone with PDM interface.

1-channel burst transfer with non-periodic Frame Sync mode is useful typically for passing control non-auto data as in case of DSP. In Burst mode, a single Data transfer starts at each Frame Sync pulse, and these pulses are 1-bit wide and occur only when a Data transfer is requested.
Note: Compact stereo modes (16C and 8C) are not supported in the Burst mode.

Sections 37.6.4. I2S Format - Reception and Transmission Sequence with Word Select, 37.6.5. TDM Format - Reception and Transmission Sequence and 37.7. I2S Application Examples describe more about frame/data formats and register settings required for different I^2S applications.

Figure 37-3. I²S Functional Block Diagram



37.6.1.1 Initialization

The I²S features two Clock Units, one Transmit Serializer, and One Receive Serializer. The Transmit Serializer uses Clock Unit 0, while the Receive Serializer can either share the same Clock Unit 0 or use the Clock Unit 1.

There are two generic clocks, GCLK_I2S_0 and GCLK_I2S_1, connected to the I²S peripheral, one for each I²S clock unit. The generic clocks (GCLK_I2S_n, n=0..1) can be set to a wide range of frequencies and clock sources. The GCLK_I2S_n must be enabled and configured before use.

The GCLK_I2S_n clocks must be enabled and configured before triggering Software Reset, so that the logic in all clock domains can be reset.

The generic clocks are only used in Host mode and Controller mode. In Host mode, the clock from clock unit 0 can be used for both Serializers to handle synchronous transfers, or a separate clock from different clock units can be used for each Serializer to handle transfers on non-related clocks.

Before enabling the I²S, the following registers must be configured:

- Clock Control registers (CLKCTRLn)
- Serializer Control registers (TXCTRL and/or RXCTRL)

In Host mode, one of the generic clocks for the I²S must also be configured to operate at the required frequency, as described in 37.6.1. Principle of Operation.

- *fs* is the sampling frequency that defines the frame period
- CLKCTRLn.NBSLOTS defines the number of slots in each frame
- CLKCTRLn.SLOTSIZE defines the number of bits in each slot
- SCKn frequency must be f_{SCKn} = *fs* × number_of_slots × number_of_bits_per_slot)

Once the configuration has been written, the I²S Clock Units and Serializers can be enabled by writing a '1' to the CKENn, TXEN, and/or RXEN bits and to the ENABLE bit in the Control register (CTRLA). The Clock Unit n can be enabled alone, in Controller Mode, to output clocks to the MCKn, SCKn, and FSn pins. The Clock Units must be enabled if Serializers are enabled.

The Clock Units, the Transmit Serializer and the Receive Serializer can be disabled independently by writing a '0' to CTRLA.CKENn, CTRLA.TXEN, and CTRLA.RXEN, respectively. Once requested to stop, they will only stop when the pending transmit frames will be completed, if any. When requested to stop, the ongoing reception of the current slot will be completed and then the Serializer will be stopped.

Example 37-1. ***Example Requirements: fs*=48kHz, MCKn=384×*fs***

If a 384×*fs* MCKn Host Clock is required (i.e. 18.432MHz), the I^2S generic clock could run at 18.432MHz with a Host Clock Output Division Factor of 1 (selected by writing CLKCTRLn.MCKOUTDIV=0x0) in order to obtain the desired MCKn frequency.

When using 6 slots per frame (CLKCTRLn.NBSLOTS=0x5) and 32-bit slots (CLKCTRLn.SLOTSIZE=0x3), the desired SCKn frequency is

f_{SCKn} = 48kHz × 6 × 32 = 9.216MHz

This frequency can be achieved by dividing the I^2S generic clock output of 18.432MHz by factor 2: Writing CLKCTRLn.MCKDIV=0x1 will select the correct division factor and output the desired SCKn frequency of 9.216MHz to the SCKn pin.

If MCKn is not required, the generic clock could be set to 9.216MHz and CLKCTRLn.MCKDIV=0x0.

37.6.2 Basic Operation

The Receiver can be operated by reading the Receive Data Holding register (RXDATA), whenever the Receive Ready m bit in the Interrupt Flag Status and Clear register (INTFLAG.RXRDYm) is set. Successive values read from the RXDATA register will correspond to the samples from the left and right audio channels. In TDM mode, the successive values read from RXDATA correspond to the first slot to the last slot. For instance, if I^2S is configured in TDM mode with 4 slots in a frame, then successive values written to RXDATA register correspond to first, second, third, and fourth slot. The number of slots in TDM is configured in CLKCTRLn.NBSLOTS.

The Transmitter can be operated by writing to the Transmit Data Holding register (TXDATA), whenever the Transmit Ready m bit in the Interrupt Flag Status and Clear register (INTFLAG.TXRDYm) is set. Successive values written to TXDATA register should correspond to the samples from the left and right audio channels. In TDM mode, the successive values written to TXDATA correspond to the first, second, third, slot to the last slot. The number of slots in TDM is configured in CLKCTRLn.NBSLOTS.

The Receive Ready and Transmit Ready bits can be polled by reading the INTFLAG register.

The processor load can be reduced by enabling interrupt-driven operation. The RXRDYm and/or TXRDYm interrupt requests can be enabled by writing a '1' to the corresponding bit in the Interrupt Enable register (INTENSET). The interrupt service routine associated to the I^2S interrupt request will then be executed whenever Receive Ready or Transmit Ready status bits are set.

The processor load can be reduced further by enabling DMA-driven operation. Then, the DMA channels support up to four trigger sources from the I^2S peripheral. These four trigger sources in DMAC channel are

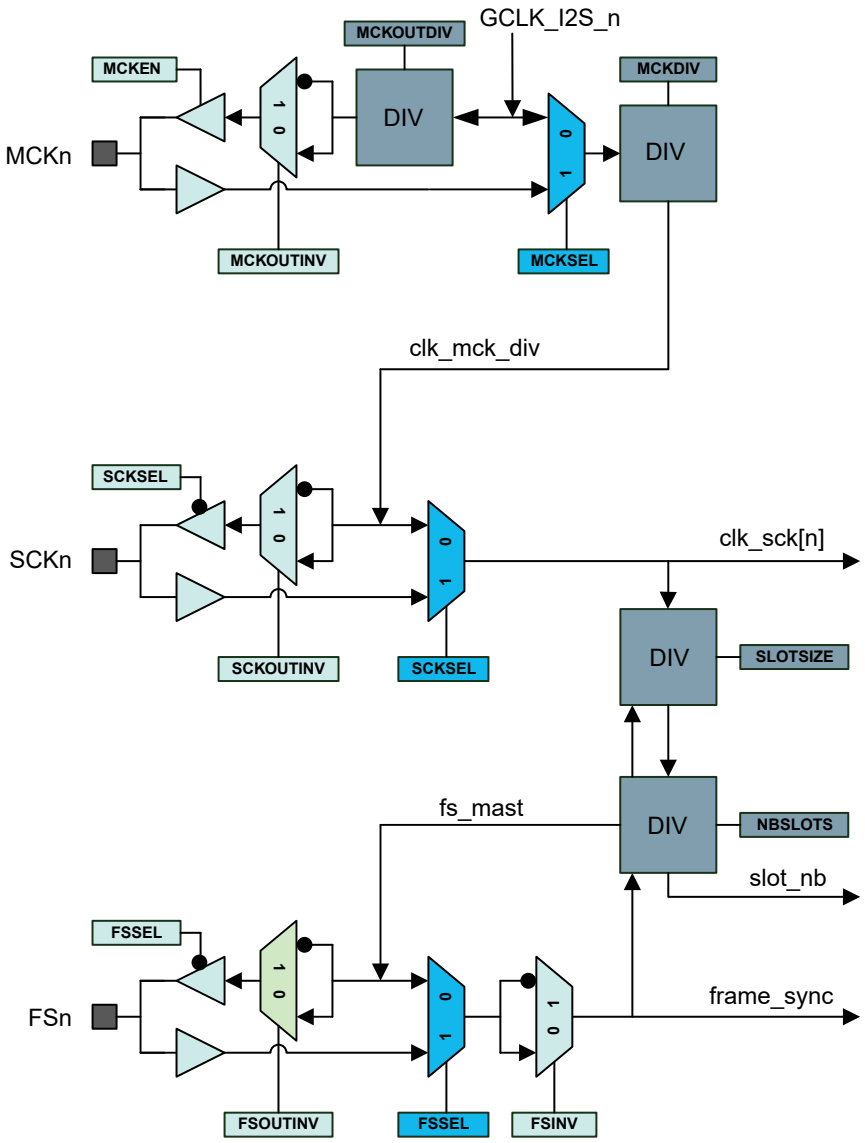
- I2S RX 0,
- I2S RX 1,
- I2S TX 0, and
- I2S TX 1.

For further reference, these are called I2S_DMAC_ID_RX_m and I2S_DMAC_ID_TX_m triggers (m=0..1). By using these trigger sources, one DMA data transfer will be executed whenever the Receive Ready or Transmit Ready status bits are set.

37.6.2.1 Host Clock, Serial Clock, and Frame Sync Generation

The generation of clocks in the I^2S is described in the next figure.

Figure 37-4. I²S Clocks



Generation

37.6.2.1.1 Client Mode

In Client mode, the Serial Clock and Frame Sync (Word Select in I²S mode and Frame Sync in TDM mode) are driven by an external host. SCKn and FSn pins are inputs and no generic clock is required by the I²S.

37.6.2.1.2 Host Mode and Controller Mode

In Host Mode, the Host Clock (MCKn), the Serial Clock (SCKn), and the Frame Sync Clock (FSn) are generated by the I²S controller. The user can configure the Host Clock, Serial Clock, and Word Select Frame Sync signal (Word Select in I²S mode and Frame Sync in TDM mode) using the Clock Unit n Control register (CLKCTRLn). MCKn, SCKn, and FSn pins are outputs and a generic clock is used to derive the I²S clocks.

In some applications, audio CODECs connected to the I²S pins may require a Host Clock signal with a frequency multiple of the audio sample frequency *fs*, such as 256×*fs*.

In Controller mode, only the Clock generation unit needs to be configured by writing to the CTRLA and CLKCTRLn registers, where parameters such as clock division factors, Number of slots, Slot size, Frame Sync signal, clock enable are selected.

37.6.2.1.3 MCKn Clock Frequency

When the I²S is in Host mode, writing a '1' to CLKCTRLn.MCKEN will output GCLK_I2S_n as Host Clock to the MCKn pin. The Host Clock to MCKn pin can be divided by writing to CLKCTRLn.MCKSEL and CLKCTRLn.MCKOUTDIV. The Host Clock (MCKn) frequency is GCLK_I2S_n frequency divided by (MCLKOUTDIV+1).

$$f(\mathrm{MCKn}) = \frac{f(\mathrm{GCLK\_I2S\_}n)}{(\mathrm{MCKOUTDIV}+1)}$$

37.6.2.1.4 SCKn Clock Frequency

When the Serial Clock (SCKn) is generated from GCLK_I2S_n and both CLKCTRLn.MCKSEL and CLKCTRLn.SCKSEL are zero, the Serial Clock (SCKn) frequency is GCLK_I2S_n frequency divided by (MCKDIV+1).

i.e.

$$f(\mathrm{SCKn}) = \frac{f(\mathrm{GCLK\_I2S\_}n)}{(\mathrm{MCKDIV}+1)}$$

37.6.2.1.5 Relation Between MCKn, SCKn, and Sampling Frequency *fs*

Based on sampling frequency ***fs***, the SCKn frequency requirement can be calculated:

- SCKn frequency: $f_{\mathrm{SCKn}} = fs \times \mathrm{total\_number\_of\_bits\_per\_frame}$,
- Where $\mathrm{total\_number\_of\_bits\_per\_frame} = \mathrm{number\_of\_slots} \times \mathrm{number\_of\_bits\_per\_slots}$.
- The number of slots is selected by writing to the Number of Slots in Frame bit field in the Clock Unit n Control (CLKCTRLn) register: $\mathrm{number\_of\_slots} = \mathrm{NBSLOTS} + 1$.
- The number of bits per slot (8, 16, 24, or 32 bit) is selected by writing to the Slot Size bit field in CLKCTRLn: .
- Consequently, $f_{\mathrm{SCKn}} = 8 \times fs \times (\mathrm{NBSLOTS}+1) \times (\mathrm{SLOTSIZE}+1)$.

The clock frequencies f_{SCKn} and f_{MCKn} are derived from the generic clock frequency $f_{\mathrm{GCLK\_I2S\_n}}$:

- $$f_{\mathrm{GCLK\_I2S\_n}} = f_{\mathrm{SCKn}} \times (\mathrm{CLKCTRLn.MCKDIV}+1) = 8 \times fs \times (\mathrm{NBSLOTS}+1) \times (\mathrm{SLOTSIZE}+1) \times (\mathrm{MCKDIV}+1)$$

  , and
- $f_{\mathrm{GCLK\_I2S\_n}} = f_{\mathrm{MCKn}} \times (\mathrm{MCKOUTDIV}+1)$.

Substituting the right hand sides of the two last equations yields:

$$f_{\mathrm{MCKn}} = \frac{f_{\mathrm{GCLK\_I2S\_n}}}{\mathrm{MCKOUTDIV}+1}$$

$$f_{\mathrm{MCKn}} = \frac{8 \cdot fs \cdot (\mathrm{SLOTSIZE}+1) \cdot (\mathrm{NBSLOTS}+1) \cdot (\mathrm{MCKDIV}+1)}{\mathrm{MCKOUTDIV}+1}$$

If a Host Clock output is not required, the GCLK_I2S generic clock can be configured as SCKn by writing a '0'to CLKCTRLn.MCKDIV. Alternatively, if the frequency of the generic clock is a multiple of the required SCKn frequency, the MCKn-to-SCKn divider can be used with the ratio defined by writing the CLKCTRLn.MCKDIV field.

The FSn pin is used as Word Select in I²S format and as Frame Synchronization in TDM format, as described in 37.6.4. I2S Format - Reception and Transmission Sequence with Word Select and 37.6.5. TDM Format - Reception and Transmission Sequence, respectively.

37.6.2.2 Data Holding Registers

For both the Transmit and the Receive Serializer, the I²S user interface includes a Data register (TXDATA and RXDATA, respectively). They are used to access data samples for all data slots.

37.6.2.2.1 Data Reception Mode

In receiver mode, the RXDATA register stores the received data.

When a new data word is available in the RXDATA register, the Receive Ready bit (RXRDYm) in the Interrupt Flag Status and Clear register (INTFLAG) is set. Reading the RXDATA register will clear this bit.

A receive overrun condition occurs if a new data word becomes available before the previous data word has been read from the RXDATA register. Then, the Receive Overrun bit in INTFLAG will be set (INTFLAG.RXORm). This interrupt can be cleared by writing a '1' to it.

37.6.2.2.2 Data Transmission Mode

In Transmitter mode, the TXDATA register contains the data to be transmitted.

when TXDATA is empty, the Transmit Ready bit in the Interrupt Flag Status and Clear register is set (INTFLAG.TXRDYm). Writing to TXDATA will clear this bit.

A transmit underrun condition occurs if data present in TXDATA is sent and no new data is written to TXDATA register before the next time slot. Then, the Transmit Underrun bit in INTFLAG will be set (INTFLAG.TXURm). This interrupt can be cleared by writing a '1' to it. The Transmit Data when Underrun bit in the Tx Serializer Control register (TXCTRL.TXSAME) configures whether a zero data word is transmitted in case of underrun (TXCTRL.TXSAME=0), or the previous data word for the current transmit slot number is transmitted again (TXCTRL.TXSAME=1).

37.6.3 Host, Controller, and Client Modes

In Host and Controller modes, the I^{2}S provides the Serial Clock, a Word Select/Frame Sync signal and optionally a Host Clock.

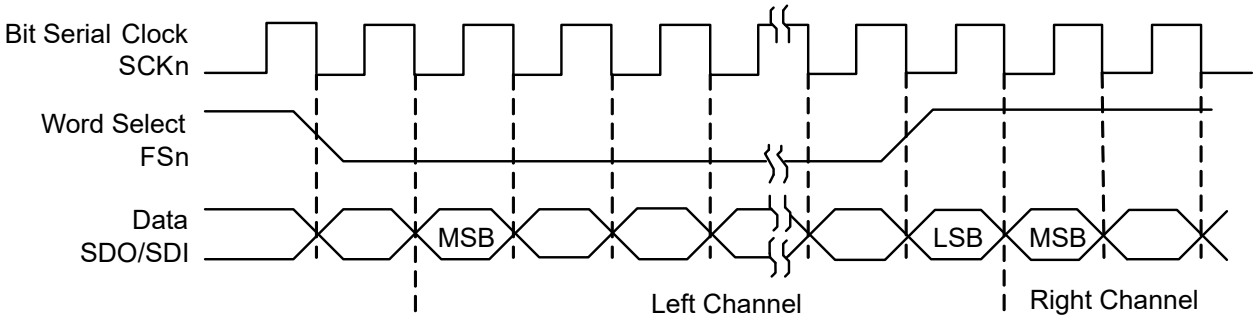
In Controller mode, the I^{2}S Serializers are disabled. Only the clocks are enabled and output for external receivers and/or transmitters.

In Client mode, the I^{2}S receives the Serial Clock and the Word Select/Frame Sync Signal from an external host. SCKn and FSn pins are inputs.

37.6.4 I^{2}S Format - Reception and Transmission Sequence with Word Select

As specified in the I^{2}S protocol, data bits are left-adjusted in the Word Select slot, with the MSB transmitted first, starting one clock period after the transition on the Word Select line.

Figure 37-5. I^{2}S Reception and Transmission Sequence



Data bits are sent on the falling edge of the Serial Clock and sampled on the rising edge of the Serial Clock. The Word Select line indicates the channel in transmission, a low level for the left channel and a high level for the right channel.

In I^{2}S format, typical configurations are described below. These configurations do not list all necessary settings, but only basic ones. Other configuration settings are to be done as per requirement such as clock and DMA configurations.

Case 1: I^{2}S 16-bit compact stereo receiver

- Slot size configured as 16 bits (CLKCTRL0.SLOTSIZE = 0x1)
- Number of slots configured as 2 (CLKCTRL0.NBSLOTS = 0x1)
- Data size configured as 16-bit compact stereo (RXCTRL.DATASIZE = 0x05)
- Data delay from Frame Sync configured as 1-bit delay (CLKCTRLn.BITDELAY = 0x01)
- Frame Sync Width configured as HALF frame (CLKCTRLn.FSWIDTH = 0x01)

Case 2: I^{2}S 24-bit stereo Transmitter with 24-bit slot

- Slot size configured as 24 bits (CLKCTRL0.SLOTSIZE = 0x2)
- Number of slots configured as 2 (CLKCTRL0.NBSLOTS = 0x1)
- Data size configured as 24 bits (TXCTRL.DATASIZE = 0x01)
- Data delay from Frame Sync configured as 1-bit delay (CLKCTRLn.BITDELAY = 0x01)

- Frame Sync Width configured as HALF frame (CLKCTRLn.FSWIDTH = 0x01)

In both cases, it will ensure that Word select signal is 'low level' for the left channel and 'high level' for the right channel.

The length of transmitted words can be chosen among 8, 16, 18, 20, 24, and 32 bits by writing the Data Word Size bit group in the Serializer Control register (RXCTRL.DATASIZE or TXCTRL.DATASIZE, respectively).

If the slot allows for more data bits than the number of bits specified in the respective DATASIZE field, additional bits are appended to the transmitted or received data word as specified in the RXCTRL/TXCTRL.EXTEND field. If the slot allows less data bits than programmed, the extra bits are not transmitted, or received data word is extended based on the EXTEND field value.

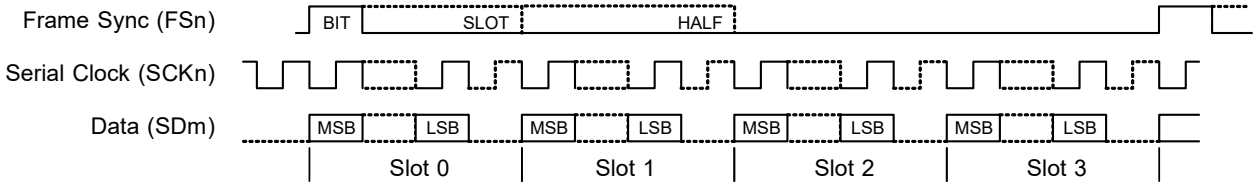
37.6.5 TDM Format - Reception and Transmission Sequence

In Time Division Multiplexed (TDM) format, the number of data slots sent or received within each frame will be (CLKCTRLn.NBSLOTS + 1).

By configuring the CLKCTRLn register (CLKCTRLn.FSWIDTH and CLKCTRLn.FSINV), the Frame Sync pulse width and polarity can be modified.

By configuring RXCTRL and/or TXCTRL, data bits can be left-adjusted or right-adjusted in the slot. It can also configure the data transmission/reception with either the MSB or the LSB transmitted/received first and starting the transmission/reception either at the transition of the FSn pin or one clock period after.

Figure 37-6. TDM Format Reception and Transmission Sequence



Data bits are sent on the falling edge of the Serial Clock and sampled on the rising edge of the Serial Clock. The FSn pin provides a frame synchronization signal, at the beginning of slot 0. The delay between the frame start and the first data bit is defined by writing the CLKCTRLn.BITDELAY field.

The Frame Sync pulse can be either one SCKn period (BIT), one slot (SLOT), or one half frame (HALF). This selection is done by writing the CLKCTRLn.FSWIDTH field.

The number of slots is selected by writing the CLKCTRLn.NBSLOTS field.

The number of bits in each slot is selected by writing the CLKCTRLn.SLOTSIZE field.

The length of transmitted words can be chosen among 8, 16, 18, 20, 24, and 32 bits by writing the DATASIZE field in the Serializer Control register (RXCTRL and/or TXCTRL).

If the slot allows more data bits than the number of bits specified in the RXCTRL. and/or TXCTRL.DATASIZE bit field, additional bits are appended to the transmitted or received data word as specified in the RXCTRL. and/or TXCTRL.EXTEND bit field. If the slot allows less data bits than programmed, the extra bits are not transmitted, or received data word is extended based on the EXTEND field value.

37.6.6 PDM Reception

In Pulse Density Modulation (PDM) reception mode, continuous 1-bit data samples are available on the SDI line on each SCKn rising edge, e.g. by a MEMS microphone with PDM interface. When using two channel PDM microphones, the second one (right channel) is configured to output data on each SCKn falling edge.

For one PDM microphone, the I^2S controller should be configured in normal Receive mode with one slot and 16- or 32-bit data size, so that 16 or 32 samples of the microphone are stored into each data word.

For two PDM microphones, the I^2S controller should be configured in PDM2 mode with one slot and 32-bit data size. The Rx Serializer will store 16 samples of each microphone in one half of the data word, with left microphone bits in lower half and right microphone bits in upper half, like in compact stereo format.

Based on oversampling frequency requirement from PDM microphone, the SCKn frequency must be configured in the I^2S controller.

A microphone that requires a sampling frequency of *fs* = 48 kHz and an oversampling frequency of *fo*=64 × *fs* would require an SCKn frequency of 3.072 MHz.

After selecting a proper frequency for GCLK_I2S_n and according Host Clock Division Factor in the Clock Unit n Control register (CLKCTRLn.MCKDIV), SCKn must be selected as per required frequency.

In PDM mode, only the clock and data line (SCKn and SDIn) pins are used.

To configure PDM2 mode, set SLOTSIZE = 0x01 (16-bits), NBSLOTS = 0x00 (1 slots) and RXCTRL.DATASIZE = 0x00 (32-bit).

37.6.7 Data Formatting Unit

To allow more flexibility, data words received by the Receive Serializer will be formatted by the Receive Formatting Unit before being stored into the Data Holding register (DATAm). The data words written into DATAm register will be formatted by the Transmit Formatting Unit before transmission by the Transmit Serializer .

The formatting options are defined in RXCTRL and TXCTRL:

- SLOTADJ for left or right justification in the slot
- BITREV for bit reversal
- WORDADJ for left or right justification in the data word
- EXTEND for extension to the word size

37.6.8 DMA, Interrupts and Events

Table 37-3. Module Request for I²S

Condition	DMA request	DMA request is cleared	Interrupt request	Event input/ output
Receive Ready	YES	When data is read	YES	
Transmit Ready (Buffer empty)	YES	When data is written	YES	
Receive Overrun			YES	
Transmit Underrun			YES	

37.6.8.1 DMA Operation

Each Serializer can be connected either to one single DMAC channel or to one DMAC channel per data slot in stereo mode. This is selected by writing the RXCTRL/TXCTRL.DMA bit:

Table 37-4. I²C DMA Request Generation

SERCTRLm.DMA	Mode	Slot Parity	DMA Request Trigger
0	Receiver	all	I2S_DMAC_ID_RX_m
	Transmitter	all	I2S_DMAC_ID_TX_m
1	Receiver	even	I2S_DMAC_ID_RX_0
		odd	I2S_DMAC_ID_RX_1
	Transmitter	even	I2S_DMAC_ID_TX_0
		odd	I2S_DMAC_ID_TX_1

The DMAC reads from the RXDATA register and writes to the TXDATA register for all data slots, successively.

The DMAC transfers may use 32-, 16- or or 8-bit transactions according to the value of the TXCTRL/ RXCTRL.DATASIZE field. 8-bit compact stereo uses 16-bit and 16-bit compact stereo uses 32-bit transactions.

37.6.8.2 Interrupts

The I²S has the following interrupt sources:

- Receive Ready (RXRDYm): this is an asynchronous interrupt and can be used to wake-up the device from any sleep mode.
- Receive Overrun (RXORm): this is an asynchronous interrupt and can be used to wake-up the device from any sleep mode.
- Transmit Ready (TXRDYm): this is an asynchronous interrupt and can be used to wake-up the device from any sleep mode.
- Transmit Underrun (TXURm): this is an asynchronous interrupt and can be used to wake-up the device from any sleep mode.

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 I^2S is reset. See INTFLAG register 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. Refer to Nested Vector Interrupt Controller for details. 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. Refer to Nested Vector Interrupt Controller for details.

37.6.9 Sleep Mode Operation

The I^2S continues to operate in all sleep modes that still provide its clocks.

37.6.10 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.

37.6.11 Synchronization

Some registers (or bit fields within a register) require synchronization when read and/or written.

Synchronization is denoted by the "Read-Synchronized" (or "Read-Synchronized Bits”) and/or "Write-Synchronized" (or "Write-Synchronized Bits”) property in each individual register description.

For more details, refer to Register Synchronization.

37.6.12 Loop-Back Mode

For debugging purposes, the I^2S can be configured to loop back the Transmitter to the Receiver. Writing a '1' to the Loop-Back Test Mode bit in the Rx Serializer Control register (RXCTRL.RXLOOP)will connect SDO to SDI, so that transmitted data is also received.

Writing RXCTRL.RXLOOP=0 will restore the normal behavior and connection between Receive Serializer and SDI pin input. As for other changes to the Serializers configuration, the Receive Serializer must be disabled before writing the TXCTRL register to update TXCTRL.RXLOOP.

37.7 I^2S Application Examples

The I^2S can support several serial communication modes used in audio or high-speed serial links. Some standard applications are shown in the following figures.

Note: The following examples are not a complete list of serial link applications supported by the I^2S.

Figure 37-7. Audio Application Block Diagram

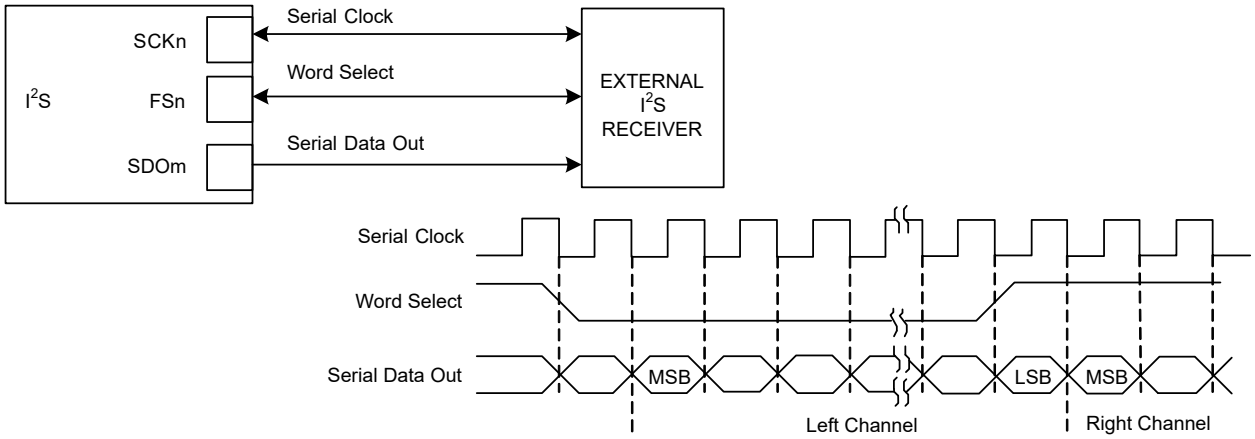


Figure 37-8. Time Slot Application Block Diagram

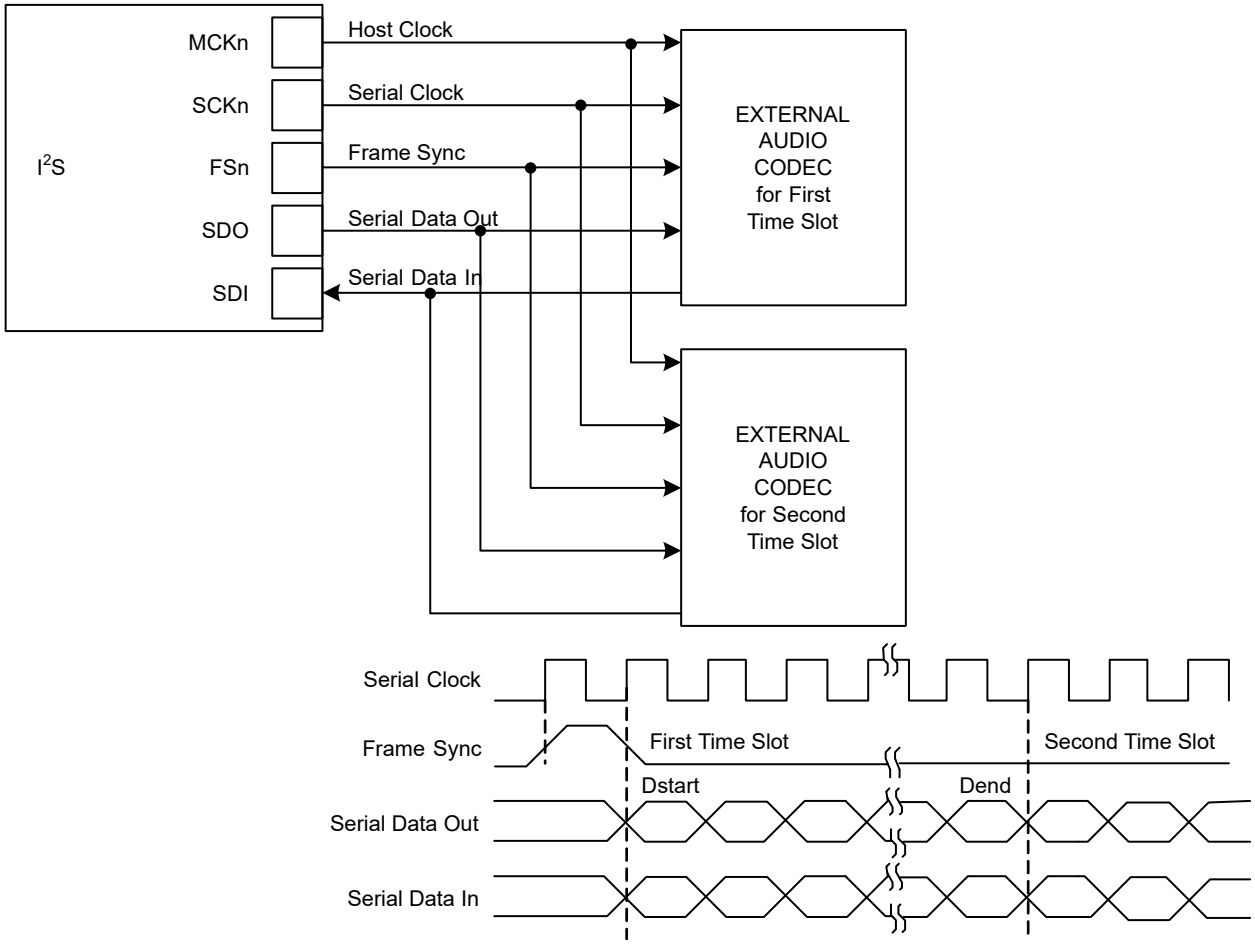


Figure 37-9. Codec Application Block Diagram

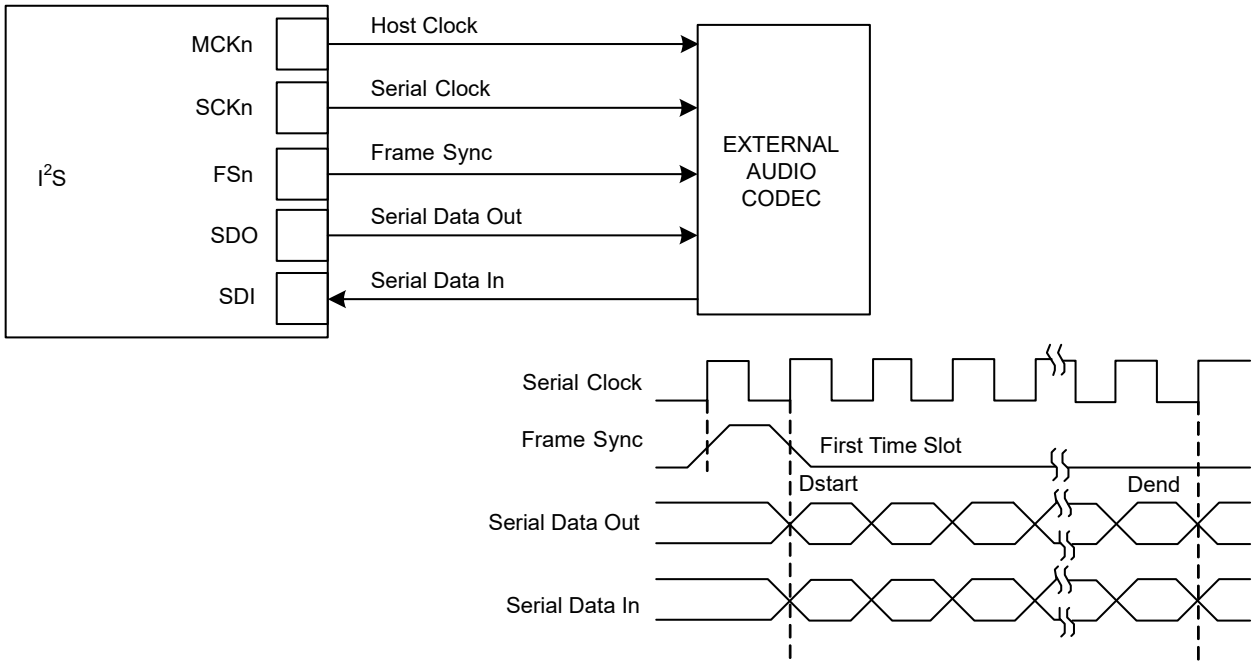
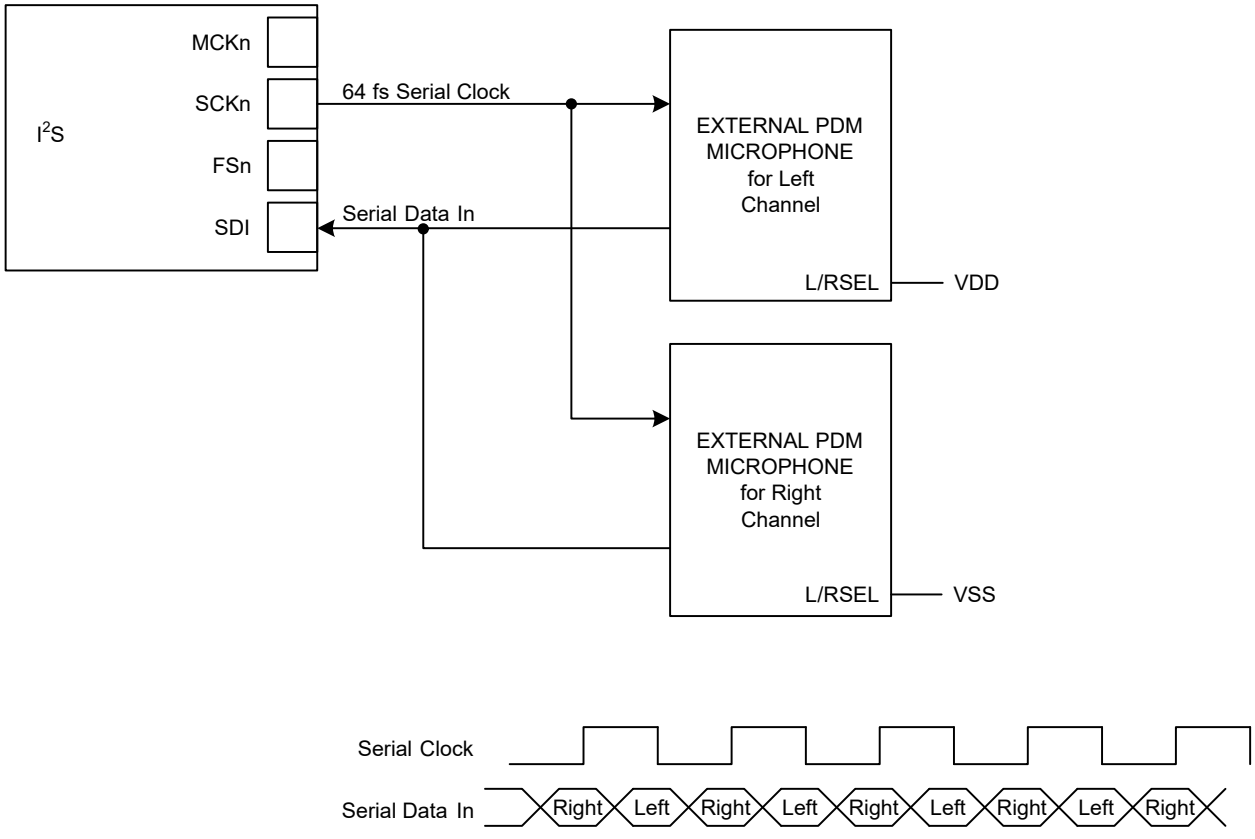


Figure 37-10. PDM Microphones Application Block Diagram



37.8 Register Summary

Refer to the Registers Description section for more details on register properties and access permissions.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRLA	7:0			RXEN	TXEN	CKEN1	CKEN0	ENABLE	SWRST
0x01 ... 0x03	Reserved									
0x04	CLKCTRL0	31:24			MCKOUTDIV[5:0]					
		23:16			MCKDIV[5:0]					
		15:8	MCKOUTINV	MCKEN	MCKSEL	SCKOUTINV	SCKSEL	FSOUTINV	FSINV	FSSEL
		7:0	BITDELAY	FSWIDTH[1:0]		NBSLOTS[2:0]			SLOTSIZE[1:0]	
0x08	CLKCTRL1	31:24			MCKOUTDIV[5:0]					
		23:16			MCKDIV[5:0]					
		15:8	MCKOUTINV	MCKEN	MCKSEL	SCKOUTINV	SCKSEL	FSOUTINV	FSINV	FSSEL
		7:0	BITDELAY	FSWIDTH[1:0]		NBSLOTS[2:0]			SLOTSIZE[1:0]	
0x0C	INTENCLR	15:8			TXUR1	TXUR0			TXRDY1	TXRDY0
		7:0			RXOR1	RXOR0			RXRDY1	RXRDY0
0x0E ... 0x0F	Reserved									
0x10	INTENSET	15:8			TXUR1	TXUR0			TXRDY1	TXRDY0
		7:0			RXOR1	RXOR0			RXRDY1	RXRDY0
0x12 ... 0x13	Reserved									
0x14	INTFLAG	15:8			TXUR1	TXUR0			TXRDY1	TXRDY0
		7:0			RXOR1	RXOR0			RXRDY1	RXRDY0
0x16 ... 0x17	Reserved									
0x18	SYNCBUSY	15:8							RXDATA	TXDATA
		7:0			RXEN	TXEN	CKEN1	CKEN0	ENABLE	SWRST
0x1A ... 0x1F	Reserved									
0x20	TXCTRL	31:24							DMA	MONO
		23:16	SLOTDIS7	SLOTDIS6	SLOTDIS5	SLOTDIS4	SLOTDIS3	SLOTDIS2	SLOTDIS1	SLOTDIS0
		15:8	BITREV	EXTEND[1:0]		WORDADJ		DATASIZE[2:0]		
		7:0	SLOTADJ			TXSAME	TXDEFAULT[1:0]			
0x24	RXCTRL	31:24						RXLOOP	DMA	MONO
		23:16	SLOTDIS7	SLOTDIS6	SLOTDIS5	SLOTDIS4	SLOTDIS3	SLOTDIS2	SLOTDIS1	SLOTDIS0
		15:8	BITREV	EXTEND[1:0]		WORDADJ		DATASIZE[2:0]		
		7:0	SLOTADJ		CLKSEL				SERMODE[1:0]	
0x28 ... 0x2F	Reserved									
0x30	TXDATA	31:24	DATA[31:24]							
		23:16	DATA[23:16]							
		15:8	DATA[15:8]							
		7:0	DATA[7:0]							
0x34	RXDATA	31:24	DATA[31:24]							
		23:16	DATA[23:16]							
		15:8	DATA[15:8]							
		7:0	DATA[7:0]							

37.8.1 Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized Bits

Bit	7	6	5	4	3	2	1	0
			RXEN	TXEN	CKEN1	CKEN0	ENABLE	SWRST
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit 5 – RXEN Rx Serializer Enable
Writing a '0' to this bit will disable the Rx Serializer.
Writing a '1' to this bit will enable the Rx Serializer.
Note: This bit is write-synchronized: SYNCBUSY.RXEN must be checked to ensure the CTRLA.RXEN synchronization is complete.

Value	Description
0	The Rx Serializer is disabled.
1	The Rx Serializer is enabled.

Bit 4 – TXEN Tx Serializer Enable
Writing a '0' to this bit will disable the Tx Serializer.
Writing a '1' to this bit will enable the Tx Serializer.
Note: This bit is write-synchronized: SYNCBUSY.TXEN must be checked to ensure the CTRLA.TXEN synchronization is complete.

Value	Description
0	The Tx Serializer is disabled.
1	The Tx Serializer is enabled.

Bits 2, 3 – CKEN Clock Unit x Enable [x=1..0]
Writing a '0' to this bit will disable the Clock Unit x.
Writing a '1' to this bit will enable the Clock Unit x.
Note: This bit is write-synchronized: SYNCBUSY.CKENx must be checked to ensure the CTRLA.CKENx synchronization is complete.

Value	Description
0	The Clock Unit x is disabled.
1	The Clock Unit x is enabled.

Bit 1 – ENABLE Enable
Writing a '0' to this bit will disable the module.
Writing a '1' to this bit will enable the module.
Note: This bit is write-synchronized: SYNCBUSY.ENABLE must be checked to ensure the CTRLA.ENABLE synchronization 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 to their initial state, and the peripheral will be disabled.
Writing a '1' to CTRL.SWRST will always take precedence, meaning that all other writes in the same write-operation will be discarded.

The I^2S generic clocks must be enabled before triggering Software Reset, so that the logic in all clock domains can be reset.

Notes:

1. When the CTRLA.SWRST is written, the user must poll SYNCBUSY.SWRST bit to know when the reset operation is complete.
2. 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.

37.8.2 Clock Unit n Control

Name: CLKCTRL
Offset: 0x04 + n*0x04 [n=0..1]
Reset: 0x00000000
Property: Enable-Protected, PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
			MCKOUTDIV[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
			MCKDIV[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
	MCKOUTINV	MCKEN	MCKSEL	SCKOUTINV	SCKSEL	FSOUTINV	FSINV	FSSEL
Access	R/W	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
	BITDELAY	FSWIDTH[1:0]		NBSLOTS[2:0]			SLOTSIZE[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 29:24 – MCKOUTDIV[5:0] Host Clock Output Division Factor
The generic clock selected by MCKSEL is divided by (MCKOUTDIV + 1) to obtain the Host Clock n output.

Bits 21:16 – MCKDIV[5:0] Host Clock Division Factor
The Host Clock n is divided by (MCKDIV + 1) to obtain the Serial Clock n.

Bit 15 – MCKOUTINV Host Clock Output Invert

Value	Description
0	The Host Clock n is output without inversion.
1	The Host Clock n is inverted before being output.

Bit 14 – MCKEN Host Clock Enable

Value	Description
0	The Host Clock n division and output is disabled.
1	The Host Clock n division and output is enabled.

Bit 13 – MCKSEL Host Clock Select
This field selects the source of the Host Clock n.

Value	Name	Description
0x0	GCLK	GCLK_I2S_n is used as Host Clock n source
0x1	MCKPIN	MCKn input pin is used as Host Clock n source

Bit 12 – SCKOUTINV Serial Clock Output Invert

Value	Description
0	The Serial Clock n is output without inversion.
1	The Serial Clock n is inverted before being output.

Bit 11 – SCKSEL Serial Clock Select

This field selects the source of the Serial Clock n.

Value	Name	Description
0x0	MCKDIV	Divided Host Clock n is used as Serial Clock n source
0x1	SCKPIN	SCKn input pin is used as Serial Clock n source

Bit 10 – FSOUTINV Frame Sync Output Invert

Value	Description
0	The Frame Sync n is output without inversion.
1	The Frame Sync n is inverted before being output.

Bit 9 – FSINV Frame Sync Invert

Value	Description
0	The Frame Sync n is used without inversion.
1	The Frame Sync n is inverted before being used.

Bit 8 – FSSEL Frame Sync Select

This field selects the source of the Frame Sync n.

Value	Name	Description
0x0	SCKDIV	Divided Serial Clock n is used as Frame Sync n source
0x1	FSPIN	FSn input pin is used as Frame Sync n source

Bit 7 – BITDELAY Data Delay from Frame Sync

Value	Name	Description
0x0	LJ	Left Justified (0 Bit Delay)
0x1	I2S	I2S (1 Bit Delay)

Bits 6:5 – FSWIDTH[1:0] Frame Sync Width

This field selects the duration of the Frame Sync output pulses.
When not in Burst mode, the Clock unit n operates in continuous mode when enabled, with periodic Frame Sync pulses and Data samples.
In Burst mode, a single Data transfer starts at each Frame Sync pulse; these pulses are 1-bit wide and occur only when a Data transfer is requested.
Note: Compact stereo modes (16C and 8C) are not supported in the Burst mode.

Value	Name	Description
0x0	SLOT	Frame Sync Pulse is 1 Slot wide (default for I2S protocol)
0x1	HALF	Frame Sync Pulse is half a Frame wide
0x2	BIT	Frame Sync Pulse is 1 Bit wide
0x3	BURST	Clock Unit n operates in Burst mode, with a 1-bit wide Frame Sync pulse per Data sample, only when Data transfer is requested

Bits 4:2 – NBSLOTS[2:0] Number of Slots in Frame

Each Frame for Clock Unit n is composed of (NBSLOTS + 1) Slots.

Bits 1:0 – SLOTSIZE[1:0] Slot Size

Each Slot for Clock Unit n is composed of a number of bits specified by SLOTSIZE.

Value	Name	Description
0x0	8	8-bit Slot for Clock Unit n
0x1	16	16-bit Slot for Clock Unit n
0x2	24	24-bit Slot for Clock Unit n
0x3	32	32-bit Slot for Clock Unit n

37.8.3 Interrupt Enable Clear

Name: INTENCLR
Offset: 0x0C
Reset: 0x0000
Property: PAC Write-Protection

Bit	15	14	13	12	11	10	9	8
			TXUR1	TXUR0			TXRDY1	TXRDY0
Access			R/W	R/W			R/W	R/W
Reset			0	0			0	0

Bit	7	6	5	4	3	2	1	0
			RXOR1	RXOR0			RXRDY1	RXRDY0
Access			R/W	R/W			R/W	R/W
Reset			0	0			0	0

Bits 12, 13 – TXUR Transmit Underrun x Interrupt Enable [x=1..0]
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Transmit Underrun x Interrupt Enable bit, which disables the Transmit Underrun x interrupt.

Value	Description
0	The Transmit Underrun x interrupt is disabled.
1	The Transmit Underrun x interrupt is enabled.

Bits 8, 9 – TXRDY Transmit Ready x Interrupt Enable [x=1..0]
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Transmit Ready x Interrupt Enable bit, which disables the Transmit Ready x interrupt.

Value	Description
0	The Transmit Ready x interrupt is disabled.
1	The Transmit Ready x interrupt is enabled.

Bits 4, 5 – RXOR Receive Overrun x Interrupt Enable [x=1..0]
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Receive Overrun x Interrupt Enable bit, which disables the Receive Overrun x interrupt.

Value	Description
0	The Receive Overrun x interrupt is disabled.
1	The Receive Overrun x interrupt is enabled.

Bits 0, 1 – RXRDY Receive Ready x Interrupt Enable [x=1..0]
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Receive Ready x Interrupt Enable bit, which disables the Receive Ready x interrupt.

Value	Description
0	The Receive Ready x interrupt is disabled.
1	The Receive Ready x interrupt is enabled.

37.8.4 Interrupt Enable Set

Name: INTENSET
Offset: 0x10
Reset: 0x0000
Property: PAC Write-Protection

Bit	15	14	13	12	11	10	9	8
			TXUR1	TXUR0			TXRDY1	TXRDY0
Access			R/W	R/W			R/W	R/W
Reset			0	0			0	0

Bit	7	6	5	4	3	2	1	0
			RXOR1	RXOR0			RXRDY1	RXRDY0
Access			R/W	R/W			R/W	R/W
Reset			0	0			0	0

Bits 12, 13 – TXUR Transmit Underrun x Interrupt Enable [x=1..0]
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Transmit Underrun Interrupt Enable bit, which enables the Transmit Underrun interrupt.

Value	Description
0	The Transmit Underrun interrupt is disabled.
1	The Transmit Underrun interrupt is enabled.

Bits 8, 9 – TXRDY Transmit Ready x Interrupt Enable [x=1..0]
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Transmit Ready Interrupt Enable bit, which enables the Transmit Ready interrupt.

Value	Description
0	The Transmit Ready interrupt is disabled.
1	The Transmit Ready interrupt is enabled.

Bits 4, 5 – RXOR Receive Overrun x Interrupt Enable [x=1..0]
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Receive Overrun Interrupt Enable bit, which enables the Receive Overrun interrupt.

Value	Description
0	The Receive Overrun interrupt is disabled.
1	The Receive Overrun interrupt is enabled.

Bits 0, 1 – RXRDY Receive Ready x Interrupt Enable [x=1..0]
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Receive Ready Interrupt Enable bit, which enables the Receive Ready interrupt.

Value	Description
0	The Receive Ready interrupt is disabled.
1	The Receive Ready interrupt is enabled.

37.8.5 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x14
Reset: 0x0000
Property: -

Bit	15	14	13	12	11	10	9	8
			TXUR1	TXUR0			TXRDY1	TXRDY0
Access			R/W	R/W			R/W	R/W
Reset			0	0			0	0

Bit	7	6	5	4	3	2	1	0
			RXOR1	RXOR0			RXRDY1	RXRDY0
Access			R/W	R/W			R/W	R/W
Reset			0	0			0	0

Bits 12, 13 – TXUR Transmit Underrun x [x=1..0]
This flag is cleared by writing a '1' to it.
This flag is set when a Transmit Underrun condition occurs in Sequencer x, and will generate an interrupt request if INTENSET.TXURx is set to '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Transmit Underrun x interrupt flag.

Bits 8, 9 – TXRDY Transmit Ready x [x=1..0]
This flag is cleared by writing to DATAx register or writing a '1' to it.
This flag is set when Sequencer x is ready to accept a new data word to be transmitted, and will generate an interrupt request if INTENSET.TXRDYx is set to '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Transmit Ready x interrupt flag.

Bits 4, 5 – RXOR Receive Overrun x [x=1..0]
This flag is cleared by writing a '1' to it.
This flag is set when a Receive Overrun condition occurs in Sequencer x, and will generate an interrupt request if INTENSET.RXORx is set to '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Receive Overrun x interrupt flag.

Bits 0, 1 – RXRDY Receive Ready x [x=1..0]
This flag is cleared by reading from DATAx register or writing a '1' to it.
This flag is set when a Sequencer x has received a new data word, and will generate an interrupt request if INTENSET.RXRDYx is set to '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Receive Ready x interrupt flag.

37.8.6 Synchronization Busy

Name: SYNCBUSY
Offset: 0x18
Reset: 0x0000
Property: -

Bit	15	14	13	12	11	10	9	8
							RXDATA	TXDATA
Access							R	R
Reset							0	0

Bit	7	6	5	4	3	2	1	0
			RXEN	TXEN	CKEN1	CKEN0	ENABLE	SWRST
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bit 9 – RXDATA Rx Data Synchronization Status
This bit is cleared when the synchronization of Rx DATA Holding register (RXDATA) between the clock domains is complete.
This bit is set when the synchronization of Rx DATA Holding register (RXDATA) between the clock domains is started.

Bit 8 – TXDATA Tx Data Synchronization Status
This bit is cleared when the synchronization of Tx DATA Holding register (TXDATA) between the clock domains is complete.
This bit is set when the synchronization of Tx DATA Holding register (TXDATA) between the clock domains is started.

Bit 5 – RXEN Rx Serializer Enable Synchronization Status
This bit is cleared when the synchronization of CTRLA.RXEN bit between the clock domains is complete.
This bit is set when the synchronization of CTRLA.RXEN bit between the clock domains is started.

Bit 4 – TXEN Tx Serializer Enable Synchronization Status
This bit is cleared when the synchronization of CTRLA.TXEN bit between the clock domains is complete.
This bit is set when the synchronization of CTRLA.TXEN bit between the clock domains is started.

Bits 2, 3 – CKEN Clock Unit x Enable Synchronization Status [x=1..0]
Bit CKENx is cleared when the synchronization of CTRLA.CKENx bit between the clock domains is complete.
Bit CKENx is set when the synchronization of CTRLA.CKENx bit between the clock domains is started.

Bit 1 – ENABLE Enable Synchronization Status
This bit is cleared when the synchronization of CTRLA.ENABLE bit between the clock domains is complete.
This bit is set when the synchronization of CTRLA.ENABLE bit between the clock domains is started.

Bit 0 – SWRST Software Reset Synchronization Status
This bit is cleared when the synchronization of CTRLA.SWRST bit between the clock domains is complete.
This bit is set when the synchronization of CTRLA.SWRST bit between the clock domains is started.

37.8.7 Tx Serializer Control

Name: TXCTRL
Offset: 0x20
Reset: 0x00000000
Property: Enable-Protected, Write-Protection

Bit	31	30	29	28	27	26	25	24
							DMA	MONO
Access							R/W	R/W
Reset							0	0

Bit	23	22	21	20	19	18	17	16
	SLOTDIS7	SLOTDIS6	SLOTDIS5	SLOTDIS4	SLOTDIS3	SLOTDIS2	SLOTDIS1	SLOTDIS0
Access	R/W	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
	BITREV	EXTEND[1:0]		WORDADJ		DATASIZE[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
	SLOTADJ			TXSAME	TXDEFAULT[1:0]			
Access	R/W			R/W	R/W	R/W		
Reset	0			0	0	0		

Bit 25 – DMA Single or Multiple DMA Channels
This bit selects whether even- and odd-numbered slots use separate DMA channels or the same DMA channel.

DMA	Name	Description
0x0	SINGLE	Single DMA channel
0x1	MULTIPLE	One DMA channel per data channel

Bit 24 – MONO Mono Mode.

MONO	Name	Description
0x0	STEREO	Normal mode
0x1	MONO	Left channel data is duplicated to right channel

Bits 16, 17, 18, 19, 20, 21, 22, 23 – SLOTDIS Slot x Disabled for this Serializer [x=7..0]
This field allows disabling some slots in each transmit frame:

Value	Description
0	Slot x is used for data transfer.
1	Slot x is not used for data transfer and will be output as specified in the TXDEFAULT field.

Bit 15 – BITREV Data Formatting Bit Reverse
This bit allows changing the order of data bits in the word in the Formatting Unit.

BITREV	Name	Description
0x0	MSBIT	Transfer Data Most Significant Bit (MSB) first (default for I2S protocol)
0x1	LSBIT	Transfer Data Least Significant Bit (LSB) first

Bits 14:13 – EXTEND[1:0] Data Formatting Bit Extension
This field defines the bit value used to extend data samples in the Formatting Unit.

EXTEND[1:0]	Name	Description
0x0	ZERO	Extend with zeros
0x1	ONE	Extend with ones
0x2	MSBIT	Extend with Most Significant Bit
0x3	LSBIT	Extend with Least Significant Bit

Bit 12 – WORDADJ Data Word Formatting Adjust

This field defines left or right adjustment of data samples in the word in the Formatting Unit. for details.

WORDADJ	Name	Description
0x0	RIGHT	Data is right adjusted in word
0x1	LEFT	Data is left adjusted in word

Bits 10:8 – DATASIZE[2:0] Data Word Size

This field defines the number of bits in each data sample. For 8-bit compact stereo, two 8-bit data samples are packed in bits 15 to 0 of the DATAm register. For 16-bit compact stereo, two 16-bit data samples are packed in bits 31 to 0 of the DATAm register.

DATASIZE[2:0]	Name	Description
0x0	32	32 bits
0x1	24	24 bits
0x2	20	20 bits
0x3	18	18 bits
0x4	16	16 bits
0x5	16C	16 bits compact stereo
0x6	8	8 bits
0x7	8C	8 bits compact stereo

Bit 7 – SLOTADJ Data Slot Formatting Adjust

This field defines left or right adjustment of data samples in the slot.

SLOTADJ	Name	Description
0x0	RIGHT	Data is right adjusted in slot
0x1	LEFT	Data is left adjusted in slot

Bit 4 – TXSAME Transmit Data when Underrun.

TXSAME	Name	Description
0x0	ZERO	Zero data transmitted in case of underrun
0x1	SAME	Last data transmitted in case of underrun

Bits 3:2 – TXDEFAULT[1:0] Line Default Line when Slot Disabled

This field defines the default value driven on the SDn output pin during all disabled Slots.

TXDEFAULT[1:0]	Name	Description
0x0	ZERO	Output Default Value is 0
0x1	ONE	Output Default Value is 1
0x2	Reserved	-
0x3	HIZ	Output Default Value is high impedance

37.8.8 Rx Serializer Control

Name: RXCTRL
Offset: 0x24
Reset: 0x00000000
Property: Enable-Protected, PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
						RXLOOP	DMA	MONO
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	23	22	21	20	19	18	17	16
	SLOTDIS7	SLOTDIS6	SLOTDIS5	SLOTDIS4	SLOTDIS3	SLOTDIS2	SLOTDIS1	SLOTDIS0
Access	R/W	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
	BITREV	EXTEND[1:0]		WORDADJ		DATASIZE[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
	SLOTADJ		CLKSEL				SERMODE[1:0]	
Access	R/W		R/W				R/W	R/W
Reset	0		0				0	0

Bit 26 – RXLOOP Loop-back Test Mode
This bit enables a loop-back test mode:

Value	Description
0	Each Receiver uses its SDn pin as input (default mode).
1	Receiver uses as input the transmitter output of the other Serializer in the pair: e.g. SD1 for SD0 or SD0 for SD1.

Bit 25 – DMA Single or Multiple DMA Channels
This bit selects whether even- and odd-numbered slots use separate DMA channels or the same DMA channel.

DMA	Name	Description
0x0	SINGLE	Single DMA channel
0x1	MULTIPLE	One DMA channel per data channel

Bit 24 – MONO Mono Mode.

MONO	Name	Description
0x0	STEREO	Normal mode
0x1	MONO	Left channel data is duplicated to right channel

Bits 16, 17, 18, 19, 20, 21, 22, 23 – SLOTDIS Slot x Disabled for this Serializer [x=7..0]
This field allows disabling some slots in each transmit frame:

Value	Description
0	Slot x is used for data transfer.
1	Slot x is not used for data transfer and will be output as specified in the TXDEFAULT field.

Bit 15 – BITREV Data Formatting Bit Reverse
This bit allows changing the order of data bits in the word in the Formatting Unit.

BITREV	Name	Description
0x0	MSBIT	Transfer Data Most Significant Bit (MSB) first (default for I2S protocol)
0x1	LSBIT	Transfer Data Least Significant Bit (LSB) first

Bits 14:13 – EXTEND[1:0] Data Formatting Bit Extension

This field defines the bit value used to extend data samples in the Formatting Unit.

EXTEND[1:0]	Name	Description
0x0	ZERO	Extend with zeros
0x1	ONE	Extend with ones
0x2	MSBIT	Extend with Most Significant Bit
0x3	LSBIT	Extend with Least Significant Bit

Bit 12 – WORDADJ Data Word Formatting Adjust

This field defines left or right adjustment of data samples in the word in the Formatting Unit. for details.

WORDADJ	Name	Description
0x0	RIGHT	Data is right adjusted in word
0x1	LEFT	Data is left adjusted in word

Bits 10:8 – DATASIZE[2:0] Data Word Size

This field defines the number of bits in each data sample. For 8-bit compact stereo, two 8-bit data samples are packed in bits 15 to 0 of the DATAm register. For 16-bit compact stereo, two 16-bit data samples are packed in bits 31 to 0 of the DATAm register.

DATASIZE[2:0]	Name	Description
0x0	32	32 bits
0x1	24	24 bits
0x2	20	20 bits
0x3	18	18 bits
0x4	16	16 bits
0x5	16C	16 bits compact stereo
0x6	8	8 bits
0x7	8C	8 bits compact stereo

Bit 7 – SLOTADJ Data Slot Formatting Adjust

This field defines left or right adjustment of data samples in the slot.

SLOTADJ	Name	Description
0x0	RIGHT	Data is right adjusted in slot
0x1	LEFT	Data is left adjusted in slot

Bit 5 – CLKSEL Clock Unit Selection.

CLKSEL	Name	Description
0x0	CLK0	Use Clock Unit 0
0x1	CLK1	Use Clock Unit 1

Bits 1:0 – SERMODE[1:0] Serializer Mode.

SERMODE[1:0]	Name	Description
0x0	RX	Receive
0x1	Reserved	-
0x2	PDM2	Receive one PDM data on each serial clock edge
0x3	Reserved	-

37.8.9 Tx Data

Name: TXDATA
Offset: 0x30
Reset: 0x00000000
Property: Write-Synchronized

Note: This register is write-synchronized: SYNCBUSY.TXDATA must be checked to ensure the TXDATA register synchronization is complete.

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] Sample Data
This register is used to transfer data to the Tx Serializer.
Data samples written to TXDATA register will be sent to Tx Serializer for transmission, through the Transmit Formatting Unit that will apply the formatting specified in the TXCTRL register.

37.8.10 Rx Data

Name: RXDATA
Offset: 0x34
Reset: 0x00000000
Property: Read-Synchronized

Note: This register is read-synchronized: SYNCBUSY.RXDATA must be checked to ensure the RXDATA register synchronization is complete.

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] Sample Data
This register is used to transfer data from the Rx Serializer.
Data samples received by Rx Serializer will be available for reading from RXDATA register, through the Receive Formatting Unit, according to formatting information for Rx Serializer in the RXCTRL register.

38. Universal Serial Bus (USB)

38.1 Overview

The Universal Serial Bus interface (USB) module complies with the Universal Serial Bus (USB) 2.1 specification supporting device modes.

The USB device mode supports 8 endpoint addresses. All endpoint addresses have one input and one output endpoint, for a total of 16 endpoints. Each endpoint is fully configurable in any of the four transfer types: control, interrupt, bulk or isochronous. The maximum data payload size is selectable up to 1023 bytes.

Internal SRAM is used to keep the configuration and data buffer for each endpoint. The memory locations used for the endpoint configurations and data buffers is fully configurable. The amount of memory allocated is dynamic according to the number of endpoints in use, and the configuration of these. The USB module has a built-in Direct Memory Access (DMA) and will read/write data from/to the system RAM when a USB transaction takes place. No CPU or DMA Controller resources are required.

To maximize throughput, an endpoint can be configured for ping-pong operation. When this is done the input and output endpoint with the same address are used in the same direction. The CPU or DMA Controller can then read/write one data buffer while the USB module writes/reads from the other buffer. This gives double buffered communication.

Multi-packet transfer enables a data payload exceeding the maximum packet size of an endpoint to be transferred as multiple packets without any software intervention. This reduces the number of interrupts and software intervention needed for USB transfers.

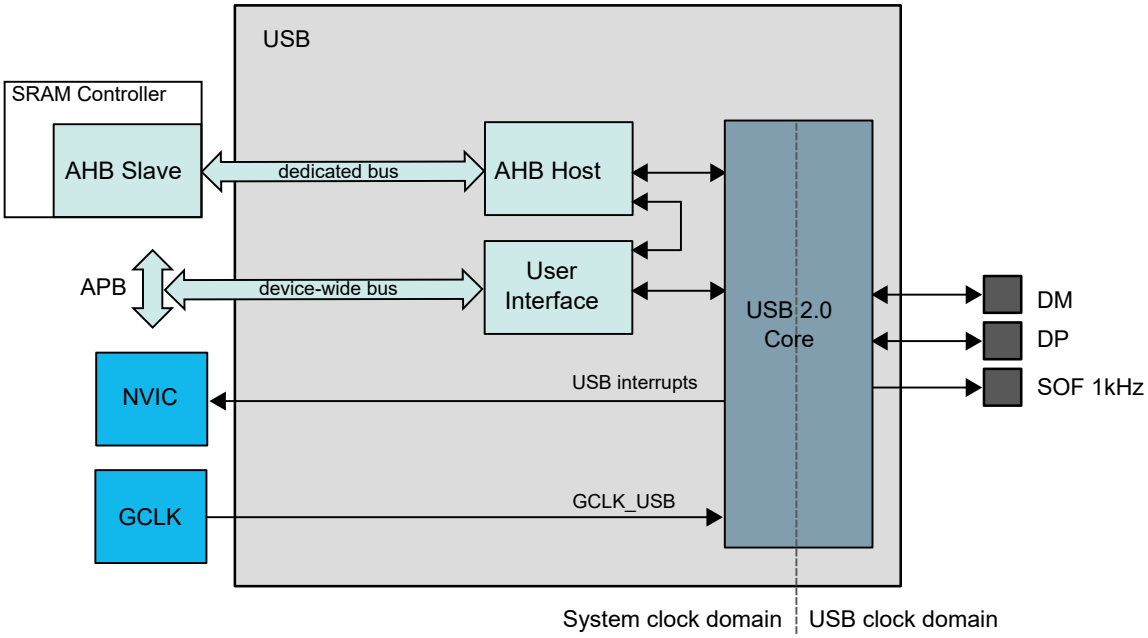
For low power operation the USB module can put the microcontroller in any sleep mode when the USB bus is idle and a suspend condition is given. Upon bus resume, the USB module can wake the microcontroller from any sleep mode.

38.2 Features

- Compatible with the USB 2.1 specification
- USB Host and Device mode
- Supports full (12Mbit/s) and low (1.5Mbit/s) speed communication
- Supports Link Power Management (LPM-L1) protocol
- On-chip transceivers with built-in pull-ups and pull-downs
- On-Chip USB serial resistors
- 1kHz SOF clock available on external pin
- Device mode
  - Supports 8 IN endpoints and 8 OUT endpoints
  - No endpoint size limitations
  - Built-in DMA with multi-packet and dual bank for all endpoints
  - Supports feedback endpoint
  - Supports crystal less clock
- Host mode
  - Supports 8 physical pipes
  - No pipe size limitations
  - Supports multiplexed virtual pipe on one physical pipe to allow an unlimited USB tree
  - Built-in DMA with multi-packet support and dual bank for all pipes
  - Supports feedback endpoint
  - Supports the USB 2.0 Phase-locked SOFs feature

38.3 USB Block Diagram

Figure 38-1. LS/FS Implementation: USB Block Diagram



38.4 Signal Description

Pin Name	Pin Description	Type
DM	Data -: Differential Data Line - Port	Input/Output
DP	Data +: Differential Data Line + Port	Input/Output
SOF 1kHZ	SOF Output	Output

Note: A 1kHz SOF clock is available on an external pin. The user must first configure the I/O Controller to assign the 1kHz SOF clock to the peripheral function. The SOF clock is available for device and host mode.

Refer to the Pinout for details on the pin mapping for this peripheral.

38.5 Peripheral Dependencies

Table 38-1. USB Configuration Summary

Peripheral name	Base address	NVIC IRQ Index	MCLK AHB/APB Clocks	GCLK Peripheral Channel Index (GCLK.PCHCTRL)	PAC Peripheral Identifier Index (PAC.WRCTRL)	Power Domain (PM.STDBYCFG)
USB	0x4100A000	20: EORSM/DNRSM, EORST/RST, LPMNYET/ DCONN, LPMSUSP/ DDISC, RAMACER, RXSTP/TXSTP, SOF/ HSOF, STALL0/STALL, STALL1, SUSPEND, TRCPT0, TRCPT1, TRFAIL0/TRFAIL, TRFAIL1/PERR, UPRSM, WAKEUP	CLK_USB_AHB CLK_USB_APB	7: GCLK_USB	37	PDSW

Note: The WAKEUP is an asynchronous interrupt and can be used to wake-up the device from any sleep mode.

38.6 Functional Description

38.6.1 USB General Operation

38.6.1.1 Initialization

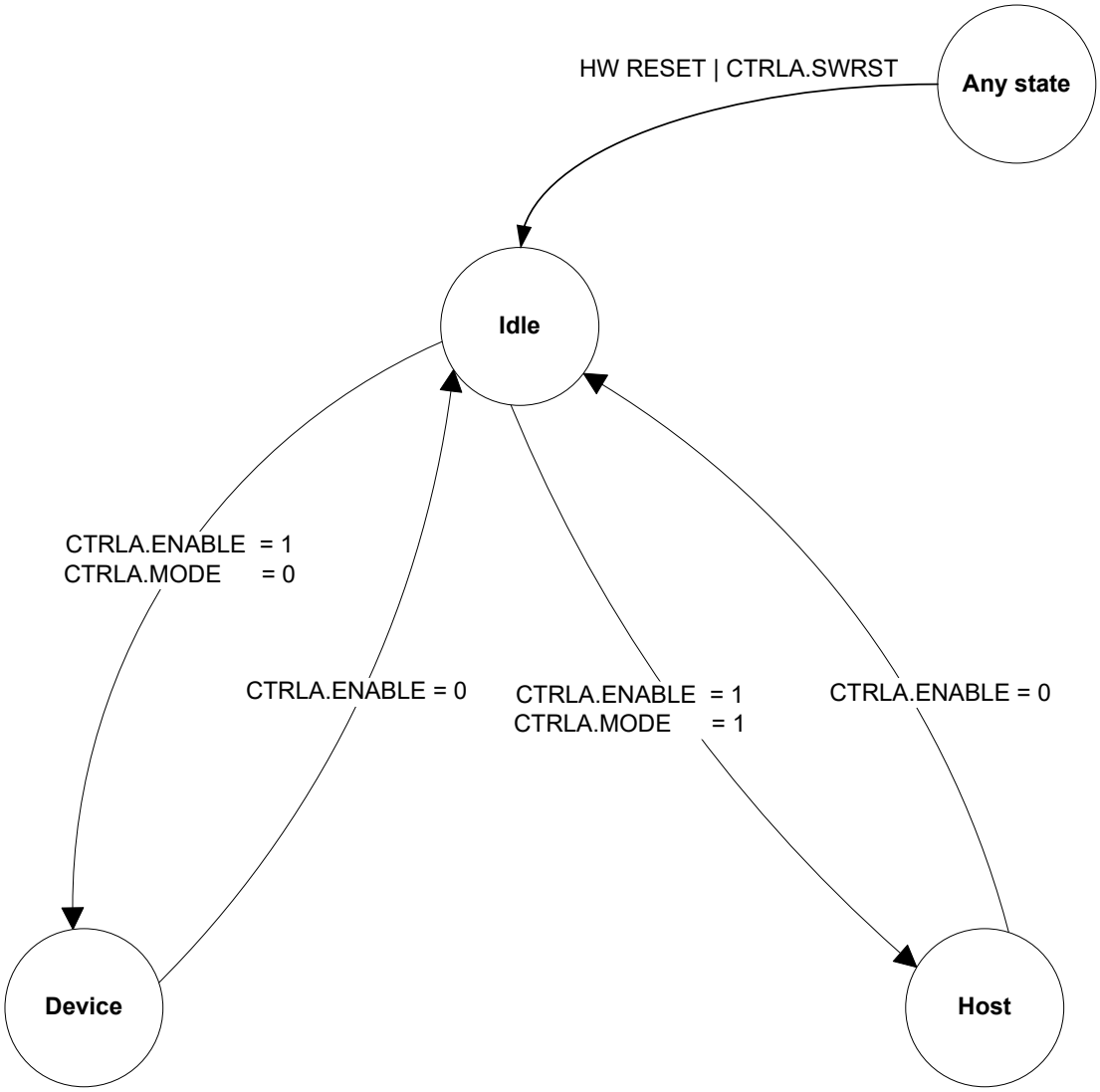
The USB module requires a GCLK_USB of 48 MHz ± 0.25% clock for low speed and full speed operation.

Clock recovery is achieved by a digital phase-locked loop in the USB module, which complies with the USB jitter specifications. If crystal-less operation is used in USB device mode, refer to *USB Clock Recovery Module*.

After a hardware reset, the USB is in the idle state. In this state:

- The module is disabled. The USB Enable bit in the Control A register (CTRLA.ENABLE) is reset.
- The module clock is stopped in order to minimize power consumption
- The USB pad is in suspend mode
- The internal states and registers of the device are reset

Figure 38-2. General States



The output drivers for the DP/DM USB line interface can be fine tuned with calibration values from production tests. The calibration values must be loaded from the NVM Software Calibration Area into the USB Pad Calibration register (PADCAL) by software, before enabling the USB, to achieve the specified accuracy. Refer to *NVM Software Calibration Area Mapping* for further details.

For details, refer to Pad Calibration (38.7.14. PADCAL) register.

The USB is enabled by writing a '1' to CTRLA.ENABLE. The USB is disabled by writing a '0' to CTRLA.ENABLE.

The USB is reset by writing a '1' to the Software Reset bit in CTRLA (CTRLA.SWRST). All registers in the USB will be reset to their initial state, and the USB will be disabled. Refer to the CTRLA register for details.

The user can configure pads and speed before enabling the USB by writing to the Operating Mode bit in the Control A register (CTRLA.MODE) and the Speed Configuration field in the Control B register (CTRLB.SPDCONF). These values are taken into account once the USB has been enabled by writing a '1' to CTRLA.ENABLE.

After writing a '1' to CTRLA.ENABLE, the USB enters device mode or host mode (according to CTRLA.MODE).

The USB can be disabled at any time by writing a '0' to CTRLA.ENABLE.

Refer to 38.6.2. USB Device Operations for the basic operation of the device mode.

Refer to 38.6.3. Host Operations for the basic operation of the host mode.

38.6.1.2 Debug Operation

When the CPU is halted in debug mode the USB peripheral continues normal operation. If the USB peripheral 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.

38.6.2 USB Device Operations

This section gives an overview of the USB module device operation during normal transactions. For more details on general USB and USB protocol, refer to the Universal Serial Bus specification revision 2.1.

38.6.2.1 Initialization

To attach the USB device to start the USB communications from the USB host, a zero should be written to the Detach bit in the Device Control B register (CTRLB.DETACH). To detach the device from the USB host, a one must be written to the CTRLB.DETACH.

After the device is attached, the host will request the USB device descriptor using the default device address zero. On successful transmission, it will send a USB reset. After that, it sends an address to be configured for the device. All further transactions will be directed to this device address. This address should be configured in the Device Address field in the Device Address register (DADD.DADD) and the Address Enable bit in DADD (DADD.ADDEN) should be written to one to accept communications directed to this address. DADD.ADDEN is automatically cleared on receiving a USB reset.

38.6.2.2 Endpoint Configuration

Endpoint data can be placed anywhere in the device RAM. The USB controller accesses these endpoints directly through the AHB host (built-in DMA) with the help of the endpoint descriptors. The base address of the endpoint descriptors needs to be written in the Descriptor Address register (DESCADD) by the user. Refer to the section “Endpoint Descriptor Structure”.

Before using an endpoint, the user should configure the direction and type of the endpoint in Type of Endpoint field in the Device Endpoint Configuration register (EPCFG.EPTYPE0/1). The endpoint descriptor registers should be initialized to known values before using the endpoint, so that the USB controller does not read random values from the RAM.

The Endpoint Size field in the Packet Size register (PCKSIZE.SIZE) should be configured as per the size reported to the host for that endpoint. The Address of Data Buffer register (ADDR) should be set to the data buffer used for endpoint transfers.

The RAM Access Interrupt bit in Device Interrupt Flag register (INTFLAG.RAMACER) is set when a RAM access underflow error occurs during IN data stage.

When an endpoint is disabled, the following registers are cleared for that endpoint:

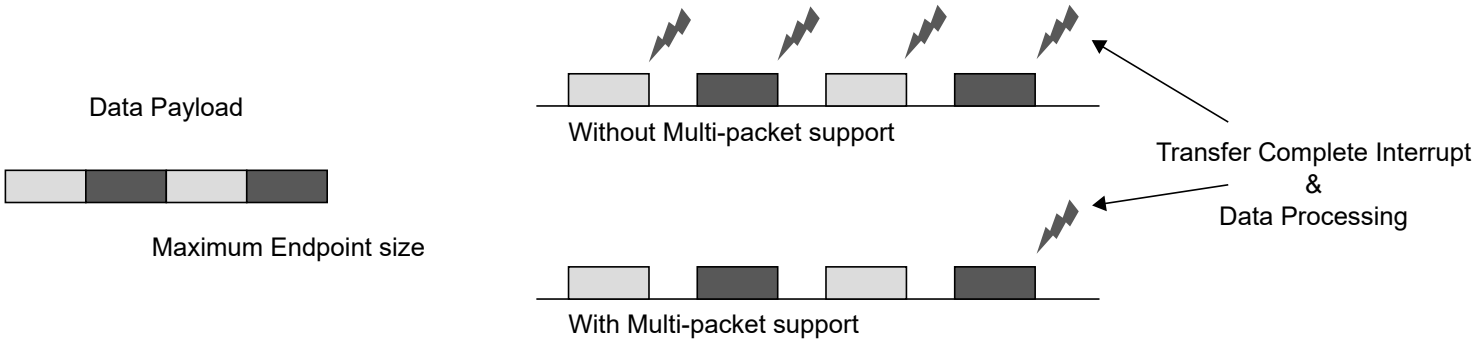
- Device Endpoint Interrupt Enable Clear/Set (EPINTENCLR/SET) register
- Device Endpoint Interrupt Flag (EPINTFLAG) register
- Transmit Stall 0 bit in the Endpoint Status register (EPSTATUS.STALLRQ0)
- Transmit Stall 1 bit in the Endpoint Status register (EPSTATUS.STALLRQ1)

38.6.2.3 Multi-Packet Transfers

Multi-packet transfer enables a data payload exceeding the endpoint maximum transfer size to be transferred as multiple packets without software intervention. This reduces the number of interrupts and software intervention required to manage higher level USB transfers. Multi-packet transfer is identical to the IN and OUT transactions described below unless otherwise noted in this section.

The application software provides the size and address of the RAM buffer to be proceeded by the USB module for a specific endpoint, and the USB module will split the buffer in the required USB data transfers without any software intervention.

Figure 38-3. Multi-Packet Feature - Reduction of CPU Overhead



38.6.2.4 USB Reset

The USB bus reset is initiated by a connected host and managed by hardware.

During USB reset the following registers are cleared:

- Device Endpoint Configuration (EPCFG) register - except for Endpoint 0
- Device Frame Number (FNUM) register
- Device Address (DADD) register
- Device Endpoint Interrupt Enable Clear/Set (EPINTENCLR/SET) register
- Device Endpoint Interrupt Flag (EPINTFLAG) register
- Transmit Stall 0 bit in the Endpoint Status register (EPSTATUS.STALLRQ0)
- Transmit Stall 1 bit in the Endpoint Status register (EPSTATUS.STALLRQ1)
- Endpoint Interrupt Summary (EPINTSMRY) register
- Upstream resume bit in the Control B register (CTRLB.UPRSM)

At the end of the reset process, the End of Reset bit is set in the Interrupt Flag register (INTFLAG.EORST).

38.6.2.5 Start-of-Frame

When a Start-of-Frame (SOF) token is detected, the frame number from the token is stored in the Frame Number field in the Device Frame Number register (FNUM.FNUM), and the Start-of-Frame interrupt bit in the Device Interrupt Flag register (INTFLAG.SOF) is set. If there is a CRC or bit-stuff error, the Frame Number Error status flag (FNUM.FNCERR) in the FNUM register is set.

38.6.2.6 Management of SETUP Transactions

When a SETUP token is detected and the device address of the token packet does not match DADD.DADD, the packet is discarded and the USB module returns to idle and waits for the next token packet.

When the address matches, the USB module checks if the endpoint is enabled in EPCFG. If the addressed endpoint is disabled, the packet is discarded and the USB module returns to idle and waits for the next token packet.

When the endpoint is enabled, the USB module then checks on the EPCFG of the addressed endpoint. If the EPCFG.EPTYPE0 is not set to control, the USB module returns to idle and waits for the next token packet.

When the EPCFG.EPTYPE0 matches, the USB module then fetches the Data Buffer Address (ADDR) from the addressed endpoint's descriptor and waits for a DATA0 packet. If a PID error or any other PID than DATA0 is detected, the USB module returns to idle and waits for the next token packet.

When the data PID matches and if the Received Setup Complete interrupt bit in the Device Endpoint Interrupt Flag register (EPINTFLAG.RXSTP) is equal to zero, ignoring the Bank 0 Ready bit in the Device Endpoint Status register (EPSTATUS.BK0RDY), the incoming data is written to the data buffer pointed to by the Data Buffer Address (ADDR). If the number of received data bytes exceeds the endpoint's maximum data payload size as specified by the PCKSIZE.SIZE, the remainders of the received data bytes are discarded. The packet will still be checked for bit-stuff and CRC errors. Software must never report a endpoint size to the host that is greater than the value configured in PCKSIZE.SIZE. If a bit-stuff or CRC error is detected in the packet, the USB module returns to idle and waits for the next token packet.

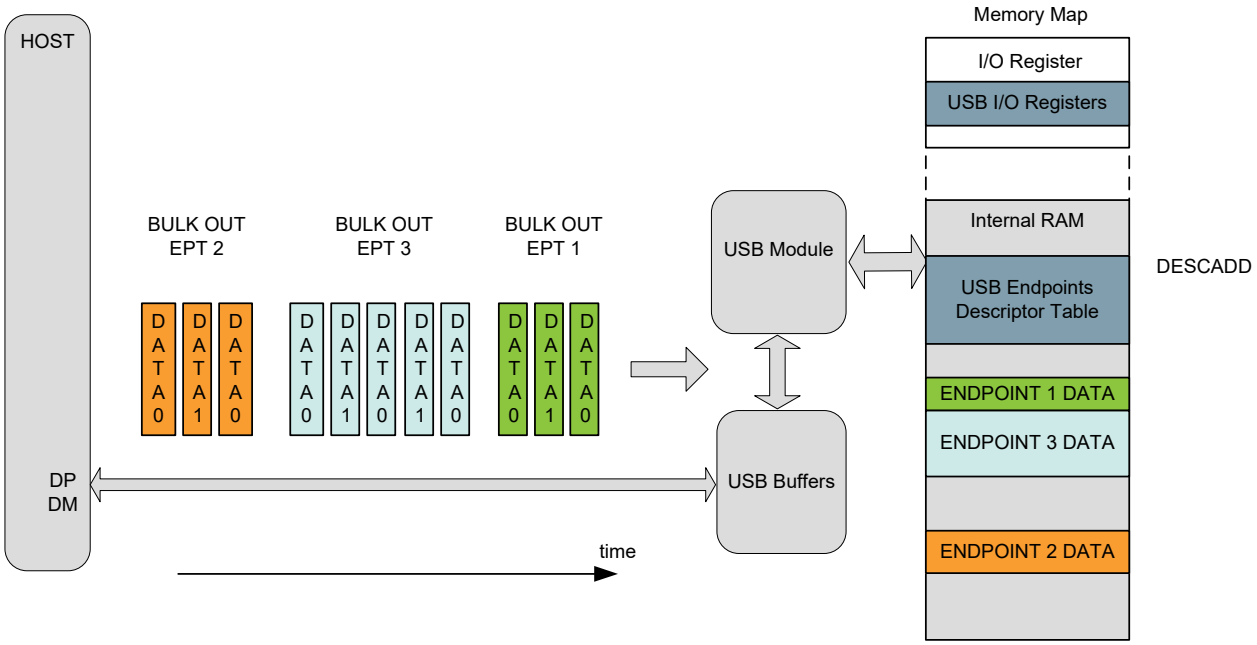
If data is successfully received, an ACK handshake is returned to the host, and the number of received data bytes, excluding the CRC, is written to the Byte Count (PCKSIZE.BYTE_COUNT). If the number of received data bytes is

the maximum data payload specified by PCKSIZE.SIZE, no CRC data is written to the data buffer. If the number of received data bytes is the maximum data payload specified by PCKSIZE.SIZE minus one, only the first CRC data is written to the data buffer. If the number of received data is equal or less than the data payload specified by PCKSIZE.SIZE minus two, both CRC data bytes are written to the data buffer.

Finally the EPSTATUS is updated. Data Toggle OUT bit (EPSTATUS.DTGLOUT), the Data Toggle IN bit (EPSTATUS.DTGLIN), the current bank bit (EPSTATUS.CURRBK) and the Bank Ready 0 bit (EPSTATUS.BK0RDY) are set. Bank Ready 1 bit (EPSTATUS.BK1RDY) and the Stall Bank 0/1 bit (EPSTATUS.STALLQR0/1) are cleared on receiving the SETUP request. The RXSTP bit is set and triggers an interrupt if the Received Setup Interrupt Enable bit is set in Endpoint Interrupt Enable Set/Clear register (EPINTENSET/CLR.RXSTP).

38.6.2.7 Management of OUT Transactions

Figure 38-4. OUT Transfer: Data Packet Host to USB Device



When an OUT token is detected, and the device address of the token packet does not match DADD.DADD, the packet is discarded and the USB module returns to idle and waits for the next token packet.

If the address matches, the USB module checks if the endpoint number received is enabled in the EPCFG of the addressed endpoint. If the addressed endpoint is disabled, the packet is discarded and the USB module returns to idle and waits for the next token packet.

When the endpoint is enabled, the USB module then checks the Endpoint Configuration register (EPCFG) of the addressed output endpoint. If the type of the endpoint (EPCFG.EPTYPE0) is not set to OUT, the USB module returns to idle and waits for the next token packet.

The USB module then fetches the Data Buffer Address (ADDR) from the addressed endpoint's descriptor, and waits for a DATA0 or DATA1 packet. If a PID error or any other PID than DATA0 or DATA1 is detected, the USB module returns to idle and waits for the next token packet.

If EPSTATUS.STALLRQ0 in EPSTATUS is set, the incoming data is discarded. If the endpoint is not isochronous, a STALL handshake is returned to the host and the Transmit Stall Bank 0 interrupt bit in EPINTFLAG (EPINTFLAG.STALL0) is set.

For isochronous endpoints, data from both a DATA0 and DATA1 packet will be accepted. For other endpoint types the PID is checked against EPSTATUS.DTGLOUT. If a PID mismatch occurs, the incoming data is discarded, and an ACK handshake is returned to the host.

If EPSTATUS.BK0RDY is set, the incoming data is discarded, the bit Transmit Fail 0 interrupt bit in EPINTFLAG (EPINTFLAG.TRFAIL0) and the status bit STATUS_BK.ERRORFLOW are set. If the endpoint is not isochronous, a NAK handshake is returned to the host.

The incoming data is written to the data buffer pointed to by the Data Buffer Address (ADDR). If the number of received data bytes exceeds the maximum data payload specified as PCKSIZE.SIZE, the remainders of the received data bytes are discarded. The packet will still be checked for bit-stuff and CRC errors. If a bit-stuff or CRC error is detected in the packet, the USB module returns to idle and waits for the next token packet.

If the endpoint is isochronous and a bit-stuff or CRC error in the incoming data, the number of received data bytes, excluding CRC, is written to PCKSIZE.BYTE_COUNT. Finally the EPINTFLAG.TRFAIL0 and CRC Error bit in the Device Bank Status register (STATUS_BK.CRCERR) is set for the addressed endpoint.

If data was successfully received, an ACK handshake is returned to the host if the endpoint is not isochronous, and the number of received data bytes, excluding CRC, is written to PCKSIZE.BYTE_COUNT. If the number of received data bytes is the maximum data payload specified by PCKSIZE.SIZE no CRC data bytes are written to the data buffer. If the number of received data bytes is the maximum data payload specified by PCKSIZE.SIZE minus one, only the first CRC data byte is written to the data buffer If the number of received data is equal or less than the data payload specified by PCKSIZE.SIZE minus two, both CRC data bytes are written to the data buffer.

Finally in EPSTATUS for the addressed output endpoint, EPSTATUS.BK0RDY is set and EPSTATUS.DTGLOUT is toggled if the endpoint is not isochronous. The flag Transmit Complete 0 interrupt bit in EPINTFLAG (EPINTFLAG.TRCPT0) is set for the addressed endpoint.

38.6.2.8 Multi-Packet Transfers for OUT Endpoint

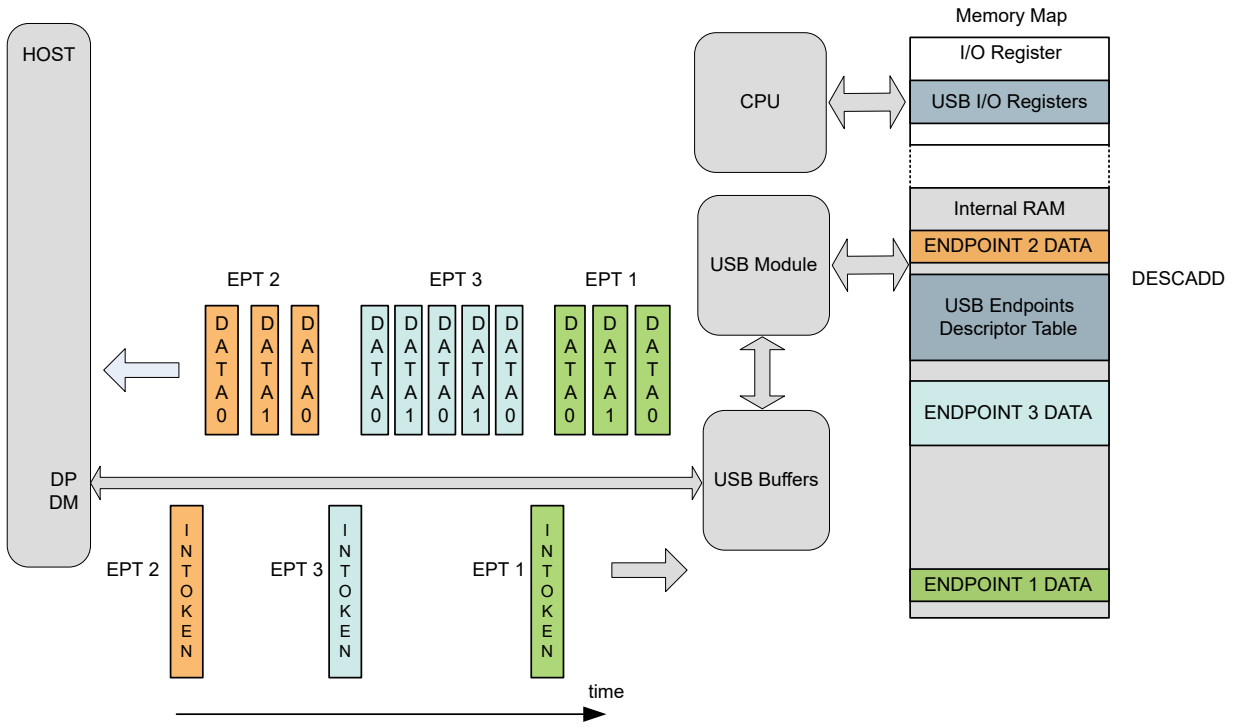
The number of data bytes received is stored in endpoint PCKSIZE.BYTE_COUNT as for normal operation. Since PCKSIZE.BYTE_COUNT is updated after each transaction, it must be set to zero when setting up a new transfer. The total number of bytes to be received must be written to PCKSIZE.MULTI_PACKET_SIZE. This value must be a multiple of PCKSIZE.SIZE, otherwise excess data may be written to SRAM locations used by other parts of the application.

EPSTATUS.DTGLOUT management for non-isochronous packets and EPINTFLAG.BK1RDY/BK0RDY management are as for normal operation.

If a maximum payload size packet is received, PCKSIZE.BYTE_COUNT will be incremented by PCKSIZE.SIZE after the transaction has completed, and EPSTATUS.DTGLOUT will be toggled if the endpoint is not isochronous. If the updated PCKSIZE.BYTE_COUNT is equal to PCKSIZE.MULTI_PACKET_SIZE (i.e. the last transaction), EPSTATUS.BK1RDY/BK0RDY, and EPINTFLAG.TRCPT0/TRCPT1 will be set.

38.6.2.9 Management of IN Transactions

Figure 38-5. IN Transfer: Data Packet USB Device to Host After Request from Host



When an IN token is detected, and if the device address of the token packet does not match DADD.DADD, the packet is discarded and the USB module returns to idle and waits for the next token packet.

When the address matches, the USB module checks if the endpoint received is enabled in the EPCFG of the addressed endpoint and if not, the packet is discarded and the USB module returns to idle and waits for the next token packet.

When the endpoint is enabled, the USB module then checks on the EPCFG of the addressed input endpoint. If the EPCFG.EPTYPE1 is not set to IN, the USB module returns to idle and waits for the next token packet.

If EPSTATUS.STALLRQ1 in EPSTATUS is set, and the endpoint is not isochronous, a STALL handshake is returned to the host and EPINTFLAG.STALL1 is set.

If EPSTATUS.BK1RDY is cleared, the flag EPINTFLAG.TRFAIL1 is set. If the endpoint is not isochronous, a NAK handshake is returned to the host.

The USB module then fetches the Data Buffer Address (ADDR) from the addressed endpoint's descriptor. The data pointed to by the Data Buffer Address (ADDR) is sent to the host in a DATA0 packet if the endpoint is isochronous. For non-isochronous endpoints a DATA0 or DATA1 packet is sent depending on the state of EPSTATUS.DTGLIN. When the number of data bytes specified in endpoint PCKSIZE.BYTE_COUNT is sent, the CRC is appended and sent to the host.

For isochronous endpoints, EPSTATUS.BK1RDY is cleared and EPINTFLAG.TRCPT1 is set.

For all non-isochronous endpoints the USB module waits for an ACK handshake from the host. If an ACK handshake is not received within 16 bit times, the USB module returns to idle and waits for the next token packet. If an ACK handshake is successfully received EPSTATUS.BK1RDY is cleared, EPINTFLAG.TRCPT1 is set and EPSTATUS.DTGLIN is toggled.

38.6.2.10 Multi-Packet Transfers for IN Endpoint

The total number of data bytes to be sent is written to PCKSIZE.BYTE_COUNT as for normal operation. The Multi-packet size register (PCKSIZE.MULTI_PACKET_SIZE) is used to store the number of bytes that are sent, and must be written to zero when setting up a new transfer.

When an IN token is received, PCKSIZE.BYTE_COUNT and PCKSIZE.MULTI_PACKET_SIZE are fetched. If PCKSIZE.BYTE_COUNT minus PCKSIZE.MULTI_PACKET_SIZE is less than the endpoint PCKSIZE.SIZE, endpoint BYTE_COUNT minus endpoint PCKSIZE.MULTI_PACKET_SIZE bytes are transmitted, otherwise PCKSIZE.SIZE number of bytes are transmitted. If endpoint PCKSIZE.BYTE_COUNT is a multiple of PCKSIZE.SIZE, the last packet sent will be zero-length if the AUTOZLP bit is set.

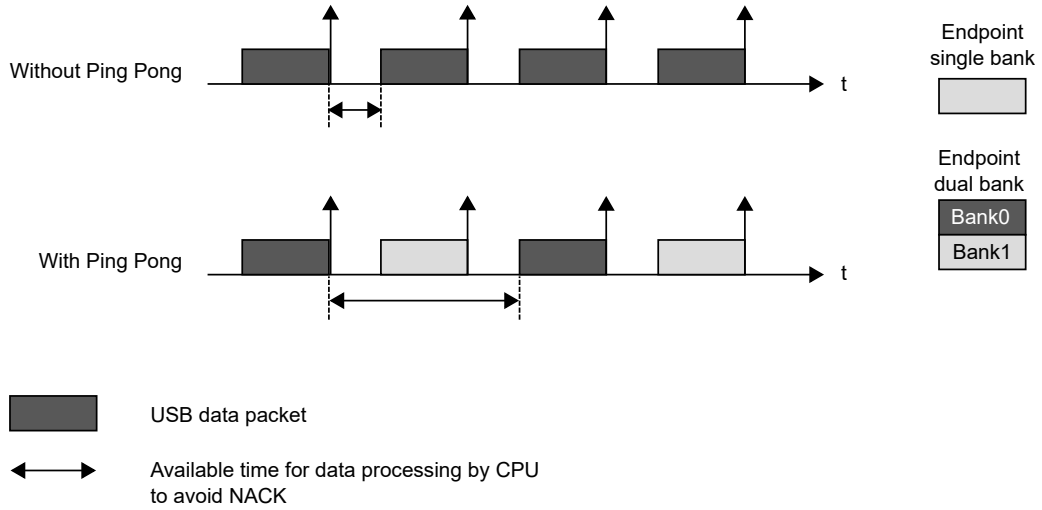
If a maximum payload size packet was sent (i.e. not the last transaction), MULTI_PACKET_SIZE will be incremented by the PCKSIZE.SIZE. If the endpoint is not isochronous the EPSTATUS.DTLGIN bit will be toggled when the transaction has completed. If a short packet was sent (i.e. the last transaction), MULTI_PACKET_SIZE is incremented by the data payload. EPSTATUS.BK0/1RDY will be cleared and EPINTFLAG.TRCPT0/1 will be set.

38.6.2.11 Ping-Pong Operation

When an endpoint is configured for ping-pong operation, it uses both the input and output data buffers (banks) for a given endpoint in a single direction. The direction is selected by enabling one of the IN or OUT direction in EPCFG.EPTYPE0/1 and configuring the opposite direction in EPCFG.EPTYPE1/0 as Dual Bank.

When ping-pong operation is enabled for an endpoint, the endpoint in the opposite direction must be configured as dual bank. The data buffer, data address pointer and byte counter from the enabled endpoint are used as Bank 0, while the matching registers from the disabled endpoint are used as Bank 1.

Figure 38-6. Ping-Pong Overview



The Bank Select flag in EPSTATUS.CURBK indicates which bank data will be used in the next transaction, and is updated after each transaction. According to EPSTATUS.CURBK, EPINTFLAG.TRCPT0 or EPINTFLAG.TRFAIL0 or EPINTFLAG.TRCPT1 or EPINTFLAG.TRFAIL1 in EPINTFLAG and Data Buffer 0/1 ready (EPSTATUS.BK0RDY and EPSTATUS.BK1RDY) are set. The EPSTATUS.DTGLOUT and EPSTATUS.DTGLIN are updated for the enabled endpoint direction only.

38.6.2.12 Feedback Operation

Feedback endpoints are endpoints with the same address but in different directions. This is usually used in explicit feedback mechanism in USB Audio, where a feedback endpoint is associated to one or more isochronous data endpoints to which it provides feedback service.

The feedback endpoint always has the opposite direction from the data endpoint(s). The feedback endpoint has the same endpoint number as the first (lower) data endpoint. A feedback endpoint can be created by configuring an endpoint with different endpoint size (PCKSIZE.SIZE) and different endpoint type (EPCFG.EPTYPE0/1) for the IN and OUT direction.

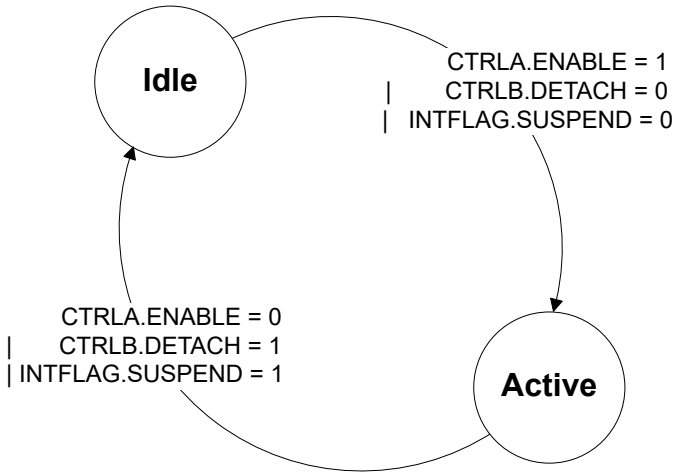
Example Configuration for Feedback Operation:

- Endpoint n / IN: EPCFG.EPTYPE1 = Interrupt IN, PCKSIZE.SIZE = 64.
- Endpoint n / OUT: EPCFG.EPTYPE0= Isochronous OUT, PCKSIZE.SIZE = 512.

38.6.2.13 Suspend State and Pad Behavior

The following figure, Pad Behavior, illustrates the behavior of the USB pad in Device mode.

Figure 38-7. Pad Behavior

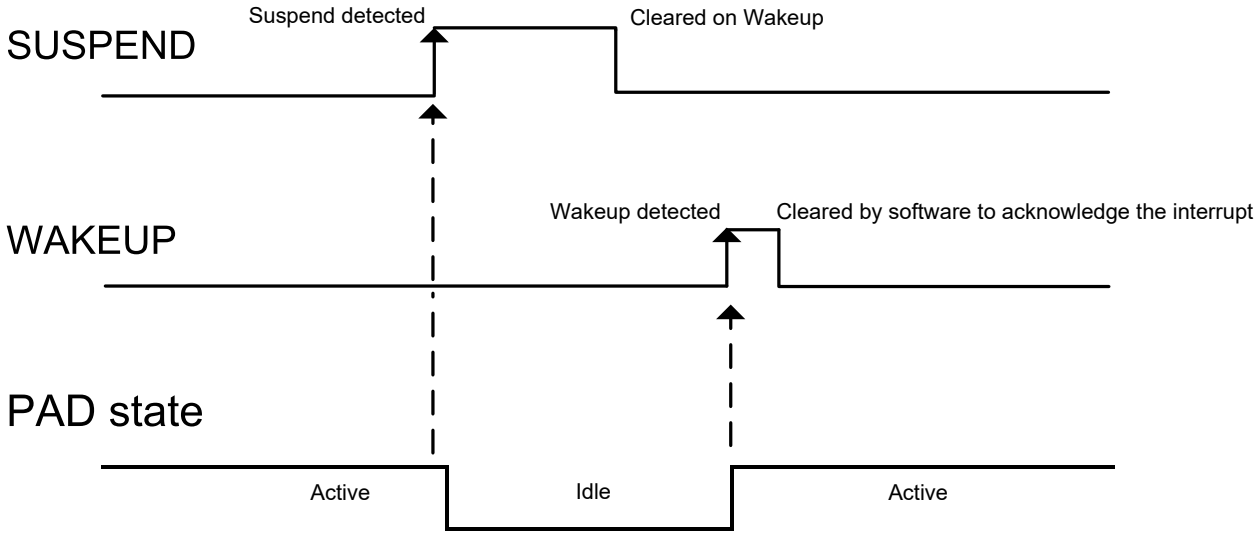


In Idle state, the pad is in Low Power Consumption mode.

In Active state, the pad is active.

The following figure, Pad Events, illustrates the pad events leading to a PAD state change.

Figure 38-8. Pad Events



The Suspend Interrupt bit in the Device Interrupt Flag register (INTFLAG.SUSPEND) is set when a USB Suspend state has been detected on the USB bus. The USB pad is then automatically put in the Idle state. The detection of a non-idle state sets the Wake Up Interrupt bit (INTFLAG.WAKEUP) and wakes the USB pad.

The pad goes to the Idle state if the USB module is disabled or if CTRLB.DETACH is written to one. It returns to the Active state when CTRLA.ENABLE is written to one and CTRLB.DETACH is written to zero.

38.6.2.14 Remote Wakeup

The remote wakeup request (also known as upstream resume) is the only request the device may send on its own initiative. This should be preceded by a DEVICE_REMOTE_WAKEUP request from the host.

First, the USB must have detected a “Suspend” state on the bus, i.e. the remote wakeup request can only be sent after INTFLAG.SUSPEND has been set.

The user may then write a one to the Remote Wakeup bit (CTRLB.UPRSM) to send an Upstream Resume to the host initiating the wakeup. This will automatically be done by the controller after 5 ms of inactivity on the USB bus.

When the controller sends the Upstream Resume INTFLAG.WAKEUP is set and INTFLAG.SUSPEND is cleared.

The CTRLB.UPRSM is cleared at the end of the transmitting Upstream Resume.

In case of a rebroadcast resume initiated by the host, the End of Resume bit (INTFLAG.EORSM) flag is set when the rebroadcast resume is completed.

In the case where the CTRLB.UPRSM bit is set while a host initiated downstream resume is already started, the CTRLB.UPRSM is cleared and the upstream resume request is ignored.

38.6.2.15 Link Power Management L1 (LPM-L1) Suspend State Entry and Exit as Device

The LPM Handshake bit in CTRLB.LPMHDSK should be configured to accept the LPM transaction.

When a LPM transaction is received on any enabled endpoint n and a handshake has been sent in response by the controller according to CTRLB.LPMHDSK, the Device Link Power Manager (EXTREG) register is updated in the bank 0 of the addressed endpoint's descriptor. It contains information such as the Best Effort Service Latency (BESL), the Remote Wake bit (bRemoteWake), and the Link State parameter (bLinkState). Usually, the LPM transaction uses only the endpoint number 0.

If the LPM transaction was positively acknowledged (ACK handshake), USB sets the Link Power Management Interrupt bit (INTFLAG.LPMSUSP) bit which indicates that the USB transceiver is suspended, reducing power consumption. This suspend occurs 9 microseconds after the LPM transaction according to the specification.

To further reduce consumption, it is recommended to stop the USB clock while the device is suspended.

The MCU can also enter in one of the available sleep modes if the wakeup time latency of the selected sleep mode complies with the host latency constraint (see the BESL parameter in 38.8.3. EXTREG register).

Recovering from this LPM-L1 suspend state is exactly the same as the Suspend state (see Section 38.6.2.13. Suspend State and Pad Behavior) except that the remote wakeup duration initiated by USB is shorter to comply with the Link Power Management specification.

If the LPM transaction is responded with a NYET, the Link Power Management Not Yet Interrupt Flag (INTFLAG.LPMNYET) is set. This generates an interrupt if the Link Power Management Not Yet Interrupt Enable bit (INTENCLR/SET.LPMNYET) is set.

If the LPM transaction is responded with no handshake, no flag is set, and the transaction is ignored.

38.6.3 Host Operations

This section gives an overview of the USB module Host operation during normal transactions. For more details on general USB and USB protocol, refer to Universal Serial Bus Specification revision 2.1.

38.6.3.1 Device Detection and Disconnection

Prior to device detection the software must set the VBUS is OK bit (CTRLB.VBUSOK) register when the VBUS is available. This notifies the USB host that USB operations can be started. When the bit CTRLB.VBUSOK is zero and even if the USB HOST is configured and enabled, host operation is halted. Setting the bit CTRLB.VBUSOK will allow host operation when the USB is configured.

The Device detection is managed by the software using the Line State field in the Host Status (STATUS.LINESTATE) register. The device connection is detected by the host controller when DP or DM is pulled high, depending of the speed of the device.

The device disconnection is detected by the host controller when both DP and DM are pulled down using the STATUS.LINESTATE registers.

The Device Connection Interrupt bit (INTFLAG.DCONN) is set if a device connection is detected.

The Device Disconnection Interrupt bit (INTFLAG.DDISC) is set if a device disconnection is detected.

38.6.3.2 Host Terminology

In host mode, the term pipe is used instead of endpoint. A host pipe corresponds to a device endpoint, refer to "Universal Serial Bus Specification revision 2.1." for more information.

38.6.3.3 USB Reset

The USB sends a USB reset signal when the user writes a one to the USB Reset bit (CTRLB.BUSRESET). When the USB reset has been sent, the USB Reset Sent Interrupt bit in the INTFLAG (INTFLAG.RST) is set and all pipes will be disabled.

If the bus was previously in a suspended state (i.e., the Start of Frame Generation Enable bit (CTRLB.SOFE) is zero), the USB will switch it to the Resume state, causing the bus to asynchronously set the Host Wakeup Interrupt flag (INTFLAG.WAKEUP). The CTRLB.SOFE bit will be set in order to generate SOFs immediately after the USB reset.

During USB reset the following registers are cleared:

- All Host Pipe Configuration register (PCFG)
- Host Frame Number register (FNUM)
- Interval for the Bulk-Out/Ping transaction register (BINTERVAL)
- Host Start-of-Frame Control register (HSOFC)
- Pipe Interrupt Enable Clear/Set register (PINTENCLR/SET)
- Pipe Interrupt Flag register (PINTFLAG)
- Pipe Freeze bit in Pipe Status register (PSTATUS.FREEZE)

After the reset the user should check the Speed Status field in the Status register (STATUS.SPEED) to find out the current speed according to the capability of the peripheral.

38.6.3.4 Pipe Configuration

Pipe data can be placed anywhere in the RAM. The USB controller accesses these pipes directly through the AHB host (built-in DMA) with the help of the pipe descriptors. The base address of the pipe descriptors needs to be written in the Descriptor Address register (DESCADD) by the user. Refer to the section ‘Pipe Descriptor Structure”.

Before using a pipe, the user should configure the direction and type of the pipe in Type of Pipe field in the Host Pipe Configuration register (PCFG.PTYPE). The pipe descriptor registers should be initialized to known values before using the pipe, so that the USB controller does not read the random values from the RAM.

The Pipe Size field in the Packet Size register (PCKSIZE.SIZE) should be configured as per the size reported by the device for the endpoint associated with this pipe. The Address of Data Buffer register (ADDR) should be set to the data buffer used for pipe transfers.

The Pipe Bank bit (PCFG.BK) should be set to one if dual banking is desired. Dual bank is not supported for Control pipes.

The Ram Access Interrupt bit in Host Interrupt Flag register (INTFLAG.RAMACER) is set when a RAM access underflow error occurs during an OUT stage.

When a pipe is disabled, the following registers are cleared for that pipe:

- Interval for the Bulk-Out/Ping transaction register (BINTERVAL)
- Pipe Interrupt Enable Clear/Set register (PINTENCLR/SET)
- Pipe Interrupt Flag register (PINTFLAG)
- Pipe Freeze bit in Pipe Status register (PSTATUS.FREEZE)

38.6.3.5 Pipe Activation

A disabled pipe is inactive, and will be reset along with its context registers (pipe registers for the pipe n). Pipes are enabled by writing the Type of the Pipe bit (PCFG.PTYPE) to a value different than 0x0 (disabled).

When a pipe is enabled, the Pipe Freeze bit in the Pipe Status register (PSTATUS.FREEZE) is set. This allows the user to complete the configuration of the pipe, without starting a USB transfer.

When starting an enumeration, the user retrieves the device descriptor by sending a GET_DESCRIPTOR USB request. This descriptor contains the maximal packet size of the device default control endpoint (bMaxPacketSize0), which the user should use to reconfigure the size of the default control pipe.

38.6.3.6 Pipe Address Setup

Once the device has answered the first host requests with the default device address 0, the host assigns a new address to the device. The host controller has to send a USB reset to the device and a SET_ADDRESS(addr) SETUP request with the new address to be used by the device. Once this SETUP transaction is complete, the user writes the new address to the Pipe Device Address field in the Host Control Pipe register (CTRL_PIPE.PDADDR) in Pipe descriptor. All following requests by this pipe will be performed using this new address.

38.6.3.7 Suspend and Wakeup

Setting CTRLB.SOFE to zero when in host mode will cause the USB to cease sending Start-of-Frames on the USB bus and enter the Suspend state. The USB device will enter the Suspend state 3ms later.

Before entering suspend by writing CTRLB.SOFE to zero, the user must freeze the active pipes by setting their PSTATUS.FREEZE bit. Any current on-going pipe will complete its transaction, and then all pipes will be inactive. The user should wait at least 1 complete frame before entering the suspend mode to avoid any data loss.

The device can awaken the host by sending an Upstream Resume (Remote Wakeup feature). When the host detects a non-idle state on the USB bus, it sets the INTFLAG.WAKEUP. If the non-idle bus state corresponds to an Upstream Resume (K state), the Upstream Resume Received Interrupt bit in INTFLAG (INTFLAG.UPRSM) is set and the user must generate a Downstream Resume within 1 ms and for at least 20 ms. It is required to first write a one to the Send USB Resume bit in CTRLB (CTRLB.RESUME) to respond to the upstream resume with a downstream resume. Alternatively, the host can resume from a suspend state by sending a Downstream Resume on the USB bus (CTRLB.RESUME set to 1). In both cases, when the downstream resume is completed, the CTRLB.SOFE bit is automatically set and the host enters again the active state.

38.6.3.8 Phase-locked SOFs

To support the Synchronous Endpoints capability, the period of the emitted Start-of-Frame is maintained while the USB connection is not in the active state. This does not apply for the disconnected/connected/reset states. It applies for active/idle/suspend/resume states. The period of Start-of-Frame will be 1ms when the USB connection is in active state and an integer number of milli-seconds across idle/suspend/resume states.

To ensure the Synchronous Endpoints capability, the GCLK_USB clock must be kept running. If the GCLK_USB is interrupted, the period of the emitted Start-of-Frame will be erratic.

38.6.3.9 Management of Control Pipes

A control transaction is composed of three stages:

- SETUP
- Data (IN or OUT)
- Status (IN or OUT)

The user has to change the pipe token according to each stage using the Pipe Token field in PCFG (PCFG.PTOKEN).

For control pipes only, the token is assigned a specific initial data toggle sequence:

- SETUP: Data0
- IN: Data1
- OUT: Data1

38.6.3.10 Management of IN Pipes

IN packets are sent by the USB device controller upon IN request reception from the host. All the received data from the device to the host will be stored in the bank provided the bank is empty. The pipe and its descriptor in RAM must be configured.

The host indicates it is able to receive data from the device by clearing the Bank 0/1 Ready bit in PSTATUS (PSTATUS.BK0/1RDY), which means that the memory for the bank is available for new USB transfer.

The USB will perform IN requests as long as the pipe is not frozen by the user.

The generation of IN requests starts when the pipe is unfrozen (PSTATUS.PFREEZE is set to zero).

When the current bank is full, the Transmit Complete 0/1 bit in PINTFLAG (PINTFLAG.TRCPT0/1) will be set and trigger an interrupt if enabled and the PSTATUS.BK0/1RDY bit will be set.

PINTFLAG.TRCPT0/1 must be cleared by software to acknowledge the interrupt. This is done by writing a one to the PINTFLAG.TRCPT0/1 of the addressed pipe.

The user reads the PCKSIZE.BYTE_COUNT to know how many bytes should be read.

To free the bank the user must read the IN data from the address ADDR in the pipe descriptor and clear the PKSTATUS.BK0/1RDY bit. When the IN pipe is composed of multiple banks, a successful IN transaction will switch to the next bank. Another IN request will be performed by the host as long as the PSTATUS.BK0/1RDY bit for that bank is set. The PINTFLAG.TRCPT0/1 and PSTATUS.BK0/1RDY will be updated accordingly.

The user can follow the current bank looking at Current Bank bit in PSTATUS (PSTATUS.CURBK) and by looking at Data Toggle for IN pipe bit in PSTATUS (PSTATUS.DTGLIN).

When the pipe is configured as single bank (Pipe Bank bit in PCFG (PCFG.BK) is 0), only PINTFLAG.TRCPT0 and PSTATUS.BK0 are used. When the pipe is configured as dual bank (PCFG.BK is 1), both PINTFLAG.TRCPT0/1 and PSTATUS.BK0/1 are used.

38.6.3.11 Management of OUT Pipes

OUT packets are sent by the host. All the data stored in the bank will be sent to the device provided the bank is filled. The pipe and its descriptor in RAM must be configured.

The host can send data to the device by writing to the data bank 0 in single bank or the data bank 0/1 in dual bank.

The generation of OUT packet starts when the pipe is unfrozen (PSTATUS.PFREEZE is zero).

The user writes the OUT data to the data buffer pointer by ADDR in the pipe descriptor and allows the USB to send the data by writing a one to the PSTATUS.BK0/1RDY. This will also cause a switch to the next bank if the OUT pipe is part of a dual bank configuration.

PINTFLAGn.TRCPT0/1 must be cleared before setting PSTATUS.BK0/1RDY to avoid missing an PINTFLAGn.TRCPT0/1 event.

38.6.3.12 Alternate Pipe

The user has the possibility to run sequentially several logical pipes on the same physical pipe. It allows addressing of any device endpoint of any attached device on the bus.

Before switching pipe, the user should save the pipe context (Pipe registers and descriptor for pipe n).

After switching pipe, the user should restore the pipe context (Pipe registers and descriptor for pipe n) and in particular PCFG, and PSTATUS.

38.6.3.13 Data Flow Error

This error exists only for isochronous and interrupt pipes for both IN and OUT directions. It sets the Transmit Fail bit in PINTFLAG (PINTFLAG.TRFAIL), which triggers an interrupt if the Transmit Fail bit in PINTENCLR/SET(PINTENCLR/SET.TRFAIL) is set. The user must check the Pipe Interrupt Summary register (PINTSMRY) to find out the pipe which triggered the interrupt. Then the user must check the origin of the interrupt's bank by looking at the Pipe Bank Status register (STATUS_BK) for each bank. If the Error Flow bit in the STATUS_BK (STATUS_BK.ERRORFLOW) is set then the user is able to determine the origin of the data flow error. As the user knows that the endpoint is an IN or OUT the error flow can be deduced as OUT underflow or as an IN overflow.

An underflow can occur during an OUT stage if the host attempts to send data from an empty bank. If a new transaction is successful, the relevant bank descriptor STATUS_BK.ERRORFLOW will be cleared.

An overflow can occur during an IN stage if the device tries to send a packet while the bank is full. Typically this occurs when a CPU is not fast enough. The packet data is not written to the bank and is lost. If a new transaction is successful, the relevant bank descriptor STATUS_BK.ERRORFLOW will be cleared.

38.6.3.14 CRC Error

This error exists only for isochronous IN pipes. It sets the PINTFLAG.TRFAIL, which triggers an interrupt if PINTENCLR/SET.TRFAIL is set. The user must check the PINTSMRY to find out the pipe which triggered the interrupt. Then the user must check the origin of the interrupt's bank by looking at the bank descriptor STATUS_BK for each bank and if the CRC Error bit in STATUS_BK (STATUS_BK.CRCERR) is set then the user is able to determine the origin of the CRC error. A CRC error can occur during the IN stage if the USB detects a corrupted packet. The IN packet will remain stored in the bank and PINTFLAG.TRCPT0/1 will be set.

38.6.3.15 PERR Error

This error exists for all pipes. It sets the PINTFLAG.PERR Interrupt, which triggers an interrupt if PINTFLAG.PERR is set. The user must check the PINTSMRY register to find out the pipe which can cause an interrupt.

A PERR error occurs if one of the error field in the STATUS_PIPE register in the Host pipe descriptor is set and the Error Count field in STATUS_PIPE (STATUS_PIPE.ERCNT) exceeds the maximum allowed number of Pipe error(s) as defined in Pipe Error Max Number field in CTRL_PIPE (CTRL_PIPE.PERMAX). Refer to the STATUS_PIPE register.

If one of the error field in the STATUS_PIPE register from the Host Pipe Descriptor is set and the STATUS_PIPE.ERCNT is less than the CTRL_PIPE.PERMAX, the STATUS_PIPE.ERCNT is incremented.

38.6.3.16 Link Power Management L1 (LPM-L1) Suspend State Entry and Exit as Host.

An EXTENDED LPM transaction can be transmitted by any enabled pipe. The PCFGn.PTYPE should be set to EXTENDED. Other fields as PCFG.PTOKEN, PCFG.BK and PCKSIZE.SIZE are irrelevant in this configuration. The user should also set the EXTREG.VARIABLE in the descriptor as described in 38.11.3. EXTREG register.

When the pipe is configured and enabled, an EXTENDED TOKEN followed by a LPM TOKEN are transmitted. The device responds with a valid HANDSHAKE, corrupted HANDSHAKE or no HANDSHAKE (TIME-OUT).

If the valid HANDSHAKE is an ACK, the host will immediately proceed to L1 SLEEP and the PINTFLAG.TRCT0 is set. The minimum duration of the L1 SLEEP state will be the TL1RetryAndResidency as defined in the reference document "ENGINEERING CHANGE NOTICE, USB 2.0 Link Power Management Addendum". When entering the L1 SLEEP state, the CTRLB.SOFE is cleared, avoiding Start-of-Frame generation.

If the valid HANDSHAKE is a NYET PINTFLAG.TRFAIL is set.

If the valid HANDSHAKE is a STALL the PINTFLAG.STALL is set.

If there is no HANDSHAKE or corrupted HANDSHAKE, the EXTENDED/LPM pair of TOKENS will be transmitted again until reaching the maximum number of retries as defined by the CTRL_PIPE.PERMAX in the pipe descriptor.

If the last retry returns no valid HANDSHAKE, the PINTFLAGn.PERR is set, and the STATUS_BK is updated in the pipe descriptor.

All LPM transactions, should they end up with a ACK, a NYET, a STALL or a PERR, will set the PSTATUS.PFREEZE bit, freezing the pipe before a succeeding operation. The user should unfreeze the pipe to start a new LPM transaction.

To exit the L1 STATE, the user initiate a DOWNSTREAM RESUME by setting the bit CTRLB.RESUME or a L1 RESUME by setting the Send L1 Resume bit in CTRLB (CTRLB.L1RESUME). In the case of a L1 RESUME, the K STATE duration is given by the BESL bit field in the EXTREG.VARIABLE field. See 38.11.3. EXTREG.

When the host is in the L1 SLEEP state after a successful LPM transmitted, the device can initiate an UPSTREAM RESUME. This will set the Upstream Resume Interrupt bit in INTFLAG (INTFLAG.UPRSM). The host should proceed then to a L1 RESUME as described above.

After resuming from the L1 SLEEP state, the bit CTRLB.SOFE is set, allowing Start-of-Frame generation.

38.7 Register Summary - USB Device

This Register Description section is valid if the USB is in Device mode (CTRLA.MODE=0).

Refer to the Registers Description section for more details on register properties and access permissions.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRLA	7:0	MODE					RUNSTDBY	ENABLE	SWRST
0x01	Reserved									
0x02	SYNCBUSY	7:0							ENABLE	SWRST
0x03	QOSCTRL	7:0					DQOS[1:0]		CQOS[1:0]	
0x04 ... 0x07	Reserved									
0x08	CTRLB	15:8					LPMHDSK[1:0]		GNAK	
		7:0				NREPLY	SPDCONF[1:0]		UPRSM	DETACH
0x0A	DADD	7:0	ADDEN	DADD[6:0]						
0x0B	Reserved									
0x0C	STATUS	7:0	LINESTATE[1:0]				SPEED[1:0]			
0x0D	FSMSTATUS	7:0		FSMSTATE[6:0]						
0x0E ... 0x0F	Reserved									
0x10	FNUM	15:8	FNCERR		FNUM[10:5]					
		7:0	FNUM[4:0]							
0x12 ... 0x13	Reserved									
0x14	INTENCLR	15:8							LPMSUSP	LPMNYET
		7:0	RAMACER	UPRSM	EORSM	WAKEUP	EORST	SOF		SUSPEND
0x16 ... 0x17	Reserved									
0x18	INTENSET	15:8							LPMSUSP	LPMNYET
		7:0	RAMACER	UPRSM	EORSM	WAKEUP	EORST	SOF		SUSPEND
0x1A ... 0x1B	Reserved									
0x1C	INTFLAG	15:8							LPMSUSP	LPMNYET
		7:0	RAMACER	UPRSM	EORSM	WAKEUP	EORST	SOF		SUSPEND
0x1E ... 0x1F	Reserved									
0x20	EPINTSMRY	15:8								
		7:0	EPINT7	EPINT6	EPINT5	EPINT4	EPINT3	EPINT2	EPINT1	EPINT0
0x22 ... 0x23	Reserved									
0x24	DESCADD	31:24	DESCADD[31:24]							
		23:16	DESCADD[23:16]							
		15:8	DESCADD[15:8]							
		7:0	DESCADD[7:0]							
0x28	PADCAL	15:8		TRIM[2:0]				TRANSN[4:2]		
		7:0	TRANSN[1:0]			TRANSP[4:0]				
0x2A ... 0xFF	Reserved									
0x0100	EPCFG0	7:0		EPTYPE1[2:0]				EPTYPE0[2:0]		
0x0101 ... 0x0103	Reserved									

.....continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0104	EPSTATUSCLR0	7:0	BK1RDY	BK0RDY	STALLRQ1	STALLRQ0		CURBK	DTGLIN	DTGLOUT
0x0105	EPSTATUSSET0	7:0	BK1RDY	BK0RDY	STALLRQ1	STALLRQ0		CURBK	DTGLIN	DTGLOUT
0x0106	EPSTATUS0	7:0	BK1RDY	BK0RDY	STALLRQ1	STALLRQ0		CURBK	DTGLIN	DTGLOUT
0x0107	EPINTFLAG0	7:0		STALL1	STALL0	RXSTP	TRFAIL1	TRFAIL0	TRCPT1	TRCPT0
0x0108	EPINTENCLR0	7:0		STALL1	STALL0	RXSTP	TRFAIL1	TRFAIL0	TRCPT1	TRCPT0
0x0109	EPINTENSET0	7:0		STALL1	STALL0	RXSTP	TRFAIL1	TRFAIL0	TRCPT1	TRCPT0
0x010A ... 0x011F	Reserved									
0x0120	EPCFG1	7:0			EPTYPE1[2:0]				EPTYPE0[2:0]	
0x0121 ... 0x0123	Reserved									
0x0124	EPSTATUSCLR1	7:0	BK1RDY	BK0RDY	STALLRQ1	STALLRQ0		CURBK	DTGLIN	DTGLOUT
0x0125	EPSTATUSSET1	7:0	BK1RDY	BK0RDY	STALLRQ1	STALLRQ0		CURBK	DTGLIN	DTGLOUT
0x0126	EPSTATUS1	7:0	BK1RDY	BK0RDY	STALLRQ1	STALLRQ0		CURBK	DTGLIN	DTGLOUT
0x0127	EPINTFLAG1	7:0		STALL1	STALL0	RXSTP	TRFAIL1	TRFAIL0	TRCPT1	TRCPT0
0x0128	EPINTENCLR1	7:0		STALL1	STALL0	RXSTP	TRFAIL1	TRFAIL0	TRCPT1	TRCPT0
0x0129	EPINTENSET1	7:0		STALL1	STALL0	RXSTP	TRFAIL1	TRFAIL0	TRCPT1	TRCPT0
0x012A ... 0x013F	Reserved									
0x0140	EPCFG2	7:0			EPTYPE1[2:0]				EPTYPE0[2:0]	
0x0141 ... 0x0143	Reserved									
0x0144	EPSTATUSCLR2	7:0	BK1RDY	BK0RDY	STALLRQ1	STALLRQ0		CURBK	DTGLIN	DTGLOUT
0x0145	EPSTATUSSET2	7:0	BK1RDY	BK0RDY	STALLRQ1	STALLRQ0		CURBK	DTGLIN	DTGLOUT
0x0146	EPSTATUS2	7:0	BK1RDY	BK0RDY	STALLRQ1	STALLRQ0		CURBK	DTGLIN	DTGLOUT
0x0147	EPINTFLAG2	7:0		STALL1	STALL0	RXSTP	TRFAIL1	TRFAIL0	TRCPT1	TRCPT0
0x0148	EPINTENCLR2	7:0		STALL1	STALL0	RXSTP	TRFAIL1	TRFAIL0	TRCPT1	TRCPT0
0x0149	EPINTENSET2	7:0		STALL1	STALL0	RXSTP	TRFAIL1	TRFAIL0	TRCPT1	TRCPT0
0x014A ... 0x015F	Reserved									
0x0160	EPCFG3	7:0			EPTYPE1[2:0]				EPTYPE0[2:0]	
0x0161 ... 0x0163	Reserved									
0x0164	EPSTATUSCLR3	7:0	BK1RDY	BK0RDY	STALLRQ1	STALLRQ0		CURBK	DTGLIN	DTGLOUT
0x0165	EPSTATUSSET3	7:0	BK1RDY	BK0RDY	STALLRQ1	STALLRQ0		CURBK	DTGLIN	DTGLOUT
0x0166	EPSTATUS3	7:0	BK1RDY	BK0RDY	STALLRQ1	STALLRQ0		CURBK	DTGLIN	DTGLOUT
0x0167	EPINTFLAG3	7:0		STALL1	STALL0	RXSTP	TRFAIL1	TRFAIL0	TRCPT1	TRCPT0
0x0168	EPINTENCLR3	7:0		STALL1	STALL0	RXSTP	TRFAIL1	TRFAIL0	TRCPT1	TRCPT0
0x0169	EPINTENSET3	7:0		STALL1	STALL0	RXSTP	TRFAIL1	TRFAIL0	TRCPT1	TRCPT0
0x016A ... 0x017F	Reserved									
0x0180	EPCFG4	7:0			EPTYPE1[2:0]				EPTYPE0[2:0]	
0x0181 ... 0x0183	Reserved									
0x0184	EPSTATUSCLR4	7:0	BK1RDY	BK0RDY	STALLRQ1	STALLRQ0		CURBK	DTGLIN	DTGLOUT
0x0185	EPSTATUSSET4	7:0	BK1RDY	BK0RDY	STALLRQ1	STALLRQ0		CURBK	DTGLIN	DTGLOUT
0x0186	EPSTATUS4	7:0	BK1RDY	BK0RDY	STALLRQ1	STALLRQ0		CURBK	DTGLIN	DTGLOUT
0x0187	EPINTFLAG4	7:0		STALL1	STALL0	RXSTP	TRFAIL1	TRFAIL0	TRCPT1	TRCPT0
0x0188	EPINTENCLR4	7:0		STALL1	STALL0	RXSTP	TRFAIL1	TRFAIL0	TRCPT1	TRCPT0
0x0189	EPINTENSET4	7:0		STALL1	STALL0	RXSTP	TRFAIL1	TRFAIL0	TRCPT1	TRCPT0

.....continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x018A ... 0x019F	Reserved									
0x01A0	EPCFG5	7:0			EPTYPE1[2:0]				EPTYPE0[2:0]	
0x01A1 ... 0x01A3	Reserved									
0x01A4	EPSTATUSCLR5	7:0	BK1RDY	BK0RDY	STALLRQ1	STALLRQ0		CURBK	DTGLIN	DTGLOUT
0x01A5	EPSTATUSSET5	7:0	BK1RDY	BK0RDY	STALLRQ1	STALLRQ0		CURBK	DTGLIN	DTGLOUT
0x01A6	EPSTATUS5	7:0	BK1RDY	BK0RDY	STALLRQ1	STALLRQ0		CURBK	DTGLIN	DTGLOUT
0x01A7	EPINTFLAG5	7:0		STALL1	STALL0	RXSTP	TRFAIL1	TRFAIL0	TRCPT1	TRCPT0
0x01A8	EPINTENCLR5	7:0		STALL1	STALL0	RXSTP	TRFAIL1	TRFAIL0	TRCPT1	TRCPT0
0x01A9	EPINTENSET5	7:0		STALL1	STALL0	RXSTP	TRFAIL1	TRFAIL0	TRCPT1	TRCPT0
0x01AA ... 0x01BF	Reserved									
0x01C0	EPCFG6	7:0			EPTYPE1[2:0]				EPTYPE0[2:0]	
0x01C1 ... 0x01C3	Reserved									
0x01C4	EPSTATUSCLR6	7:0	BK1RDY	BK0RDY	STALLRQ1	STALLRQ0		CURBK	DTGLIN	DTGLOUT
0x01C5	EPSTATUSSET6	7:0	BK1RDY	BK0RDY	STALLRQ1	STALLRQ0		CURBK	DTGLIN	DTGLOUT
0x01C6	EPSTATUS6	7:0	BK1RDY	BK0RDY	STALLRQ1	STALLRQ0		CURBK	DTGLIN	DTGLOUT
0x01C7	EPINTFLAG6	7:0		STALL1	STALL0	RXSTP	TRFAIL1	TRFAIL0	TRCPT1	TRCPT0
0x01C8	EPINTENCLR6	7:0		STALL1	STALL0	RXSTP	TRFAIL1	TRFAIL0	TRCPT1	TRCPT0
0x01C9	EPINTENSET6	7:0		STALL1	STALL0	RXSTP	TRFAIL1	TRFAIL0	TRCPT1	TRCPT0
0x01CA ... 0x01DF	Reserved									
0x01E0	EPCFG7	7:0			EPTYPE1[2:0]				EPTYPE0[2:0]	
0x01E1 ... 0x01E3	Reserved									
0x01E4	EPSTATUSCLR7	7:0	BK1RDY	BK0RDY	STALLRQ1	STALLRQ0		CURBK	DTGLIN	DTGLOUT
0x01E5	EPSTATUSSET7	7:0	BK1RDY	BK0RDY	STALLRQ1	STALLRQ0		CURBK	DTGLIN	DTGLOUT
0x01E6	EPSTATUS7	7:0	BK1RDY	BK0RDY	STALLRQ1	STALLRQ0		CURBK	DTGLIN	DTGLOUT
0x01E7	EPINTFLAG7	7:0		STALL1	STALL0	RXSTP	TRFAIL1	TRFAIL0	TRCPT1	TRCPT0
0x01E8	EPINTENCLR7	7:0		STALL1	STALL0	RXSTP	TRFAIL1	TRFAIL0	TRCPT1	TRCPT0
0x01E9	EPINTENSET7	7:0		STALL1	STALL0	RXSTP	TRFAIL1	TRFAIL0	TRCPT1	TRCPT0

38.7.1 Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized Bits

Bit	7	6	5	4	3	2	1	0
	MODE					RUNSTDBY	ENABLE	SWRST
Access	R/W					R/W	R/W	R/W
Reset	0					0	0	0

Bit 7 – MODE Operating Mode
This bit defines the operating mode of the USB.

Value	Description
0	USB Device mode
1	USB Host mode

Bit 2 – RUNSTDBY Run in Standby Mode
This bit is Enable-Protected.

Value	Description
0	USB clock is stopped in standby mode.
1	USB clock is running in standby mode

Bit 1 – ENABLE Enable
Note: This bit is write-synchronized: SYNCBUSY.ENABLE must be checked to ensure the CTRLA.ENABLE synchronization is complete.

Value	Description
0	The peripheral is disabled or being disabled.
1	The peripheral is enabled or being enabled.

Bit 0 – SWRST Software Reset
Writing a zero to this bit has no effect.
Writing a '1' to this bit resets all registers in the USB, to their initial state, and the USB will be disabled.
Writing a '1' to CTRLA.SWRST will always take precedence, meaning that all other writes in the same write-operation will be discarded.
Notes:

1. When the CTRLA.SWRST is written, the user must poll SYNCBUSY.SWRST bit to know when the reset operation is complete.
2. 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.

38.7.2 Synchronization Busy

Name: SYNCBUSY
Offset: 0x02
Reset: 0x00
Property: -

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

Bit 1 – ENABLE Synchronization Enable status bit
This bit is cleared when the synchronization of ENABLE register between the clock domains is complete.
This bit is set when the synchronization of ENABLE register between clock domains is started.

Bit 0 – SWRST Synchronization Software Reset status bit
This bit is cleared when the synchronization of SWRST register between the clock domains is complete.
This bit is set when the synchronization of SWRST register between clock domains is started.

38.7.3 QOS Control

Name: QOSCTRL
Offset: 0x03
Reset: 0x0F
Property: PAC Write-Protection

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

Bits 3:2 – DQOS[1:0] Data Quality of Service
These bits define the memory priority access during the endpoint or pipe read/write data operation. Refer to SRAM Quality of Service.

Bits 1:0 – CQOS[1:0] Configuration Quality of Service
These bits define the memory priority access during the endpoint or pipe read/write configuration operation. Refer to SRAM Quality of Service.

38.7.4 Control B

Name: CTRLB
Offset: 0x08
Reset: 0x0001
Property: PAC Write-Protection

Bit	15	14	13	12	11	10	9	8
					LPMHDSK[1:0]		GNAK	
Access					R/W	R/W	R/W	
Reset					0	0	0	

Bit	7	6	5	4	3	2	1	0
				NREPLY	SPDCONF[1:0]		UPRSM	DETACH
Access				R	R/W	R/W	R/W	R/W
Reset				0	0	0	0	1

Bits 11:10 – LPMHDSK[1:0] Link Power Management Handshake
These bits select the Link Power Management Handshake configuration.

Value	Name	Description
0x0	NO	No handshake. LPM is not supported.
0x1	ACK	ACK
0x2	NYET	NYET
0x3	-	Reserved

Bit 9 – GNAK Global NAK
This bit configures the operating mode of the NAK.
This bit is not synchronized.

Value	Description
0	The handshake packet reports the status of the USB transaction
1	A NAK handshake is answered for each USB transaction regardless of the current endpoint memory bank status

Bit 4 – NREPLY No reply excepted SETUP Token
This bit is cleared by hardware when receiving a SETUP packet.
This bit has no effect for any other endpoint but endpoint 0.

Value	Description
0	Disable the “NO_REPLY” feature: Any transaction to endpoint 0 will be handled according to the USB2.0 standard.
1	Enable the “NO_REPLY” feature: Any transaction to endpoint 0 will be ignored except SETUP.

Bits 3:2 – SPDCONF[1:0] Speed Configuration
These bits select the speed configuration.

Value	Description
0x0	FS: Full-speed
0x1	LS: Low-speed
0x2	Reserved
0x3	Reserved

Bit 1 – UPRSM Upstream Resume
This bit is cleared when the USB receives a USB reset or once the upstream resume has been sent.

Value	Description
0	Writing a zero to this bit has no effect.
1	Writing a one to this bit will generate an upstream resume to the host for a remote wakeup.

Bit 0 – DETACH Detach

Value	Description
0	The device is attached to the USB bus so that communications may occur.
1	It is the default value at reset. The internal device pull-ups are disabled, removing the device from the USB bus.

38.7.5 Device Address

Name: DADD
Offset: 0x0A
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
	ADDEN	DADD[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 – ADDEN Device Address Enable
This bit is cleared when a USB reset is received.

Value	Description
0	Writing a zero will deactivate the DADD field (USB device address) and return the device to default address 0.
1	Writing a one will activate the DADD field (USB device address).

Bits 6:0 – DADD[6:0] Device Address
These bits define the device address. The DADD register is reset when a USB reset is received.

38.7.6 Status

Name: STATUS
Offset: 0x0C
Reset: 0x40
Property: -

Bit	7	6	5	4	3	2	1	0
	LINESTATE[1:0]				SPEED[1:0]			
Access	R	R			R	R		
Reset	0	1			0	1		

Bits 7:6 – LINESTATE[1:0] USB Line State Status
These bits define the current line state DP/DM.

Value	USB Line Status
0x0	SE0/RESET
0x1	FS-J or LS-K State
0x2	FS-K or LS-J State

Bits 3:2 – SPEED[1:0] Speed Status
These bits define the current speed used of the device.

Value	SPEED STATUS
0x0	Low-speed mode
0x1	Full-speed mode
0x2	Reserved
0x3	Reserved

38.7.7 Finite State Machine Status

Name: FSMSTATUS
Offset: 0x0D
Reset: 0x01
Property: -

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

Bits 6:0 – FSMSTATE[6:0] Fine State Machine Status
These bits indicate the state of the finite state machine of the USB controller.

Value	Name	Description
0x01	OFF	Corresponds to the powered-off, disconnected, and disabled state.
0x02	ON	Corresponds to the Idle and Active states.
0x04	SUSPEND	
0x08	SLEEP	
0x10	DNRESUME	Down Stream Resume.
0x20	UPRESUME	Up Stream Resume.
0x40	RESET	USB lines Reset.
Others		Reserved

38.7.8 Device Frame Number

Name: FNUM
Offset: 0x10
Reset: 0x0000
Property: PAC Write-Protection

Bit	15	14	13	12	11	10	9	8
	FNCERR		FNUM[10:5]					
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
	FNUM[4:0]							
Access	R/W	R/W	R/W	R/W	R/W			
Reset	0	0	0	0	0			

Bit 15 – FNCERR Frame Number CRC Error
This bit is cleared upon receiving a USB reset.
This bit is set when a corrupted frame number (or micro-frame number) is received.
This bit and the SOF interrupt bit are updated at the same time.

Bits 13:3 – FNUM[10:0] Frame Number
These bits are cleared upon receiving a USB reset.
These bits are updated with the frame number information as provided from the last SOF packet even if a corrupted SOF is received.

38.7.9 Device Interrupt Enable Clear

Name: INTENCLR
Offset: 0x14
Reset: 0x0000
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	15	14	13	12	11	10	9	8
							LPMSUSP	LPMNYET
Access							R/W	R/W
Reset							0	0

Bit	7	6	5	4	3	2	1	0
	RAMACER	UPRSM	EORSM	WAKEUP	EORST	SOF		SUSPEND
Access	R/W	R/W	R/W	R/W	R/W	R/W		R/W
Reset	0	0	0	0	0	0		0

Bit 9 – LPMSUSP Link Power Management Suspend Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will clear the Link Power Management Suspend Interrupt Enable bit and disable the corresponding interrupt request.

Value	Description
0	The Link Power Management Suspend interrupt is disabled.
1	The Link Power Management Suspend interrupt is enabled and an interrupt request will be generated when the Link Power Management Suspend interrupt Flag is set.

Bit 8 – LPMNYET Link Power Management Not Yet Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will clear the Link Power Management Not Yet interrupt Enable bit and disable the corresponding interrupt request.

Value	Description
0	The Link Power Management Not Yet interrupt is disabled.
1	The Link Power Management Not Yet interrupt is enabled and an interrupt request will be generated when the Link Power Management Not Yet interrupt Flag is set.

Bit 7 – RAMACER RAM Access Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will clear the RAM Access interrupt Enable bit and disable the corresponding interrupt request.

Value	Description
0	The RAM Access interrupt is disabled.
1	The RAM Access interrupt is enabled and an interrupt request will be generated when the RAM Access interrupt Flag is set.

Bit 6 – UPRSM Upstream Resume Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will clear the Upstream Resume interrupt Enable bit and disable the corresponding interrupt request.

Value	Description
0	The Upstream Resume interrupt is disabled.
1	The Upstream Resume interrupt is enabled and an interrupt request will be generated when the Upstream Resume interrupt Flag is set.

Bit 5 – EORSM End Of Resume Interrupt Enable

Writing a zero to this bit has no effect.
Writing a one to this bit will clear the End Of Resume interrupt Enable bit and disable the corresponding interrupt request.

Value	Description
0	The End Of Resume interrupt is disabled.
1	The End Of Resume interrupt is enabled and an interrupt request will be generated when the End Of Resume interrupt Flag is set.

Bit 4 – WAKEUP Wake-Up Interrupt Enable

Writing a zero to this bit has no effect.
Writing a one to this bit will clear the Wake Up interrupt Enable bit and disable the corresponding interrupt request.

Value	Description
0	The Wake Up interrupt is disabled.
1	The Wake Up interrupt is enabled and an interrupt request will be generated when the Wake Up interrupt Flag is set.

Bit 3 – EORST End of Reset Interrupt Enable

Writing a zero to this bit has no effect.
Writing a one to this bit will clear the End of Reset interrupt Enable bit and disable the corresponding interrupt request.

Value	Description
0	The End of Reset interrupt is disabled.
1	The End of Reset interrupt is enabled and an interrupt request will be generated when the End of Reset interrupt Flag is set.

Bit 2 – SOF Start-of-Frame Interrupt Enable

Writing a zero to this bit has no effect.
Writing a one to this bit will clear the Start-of-Frame interrupt Enable bit and disable the corresponding interrupt request.

Value	Description
0	The Start-of-Frame interrupt is disabled.
1	The Start-of-Frame interrupt is enabled and an interrupt request will be generated when the Start-of-Frame interrupt Flag is set.

Bit 0 – SUSPEND Suspend Interrupt Enable

Writing a zero to this bit has no effect.
Writing a one to this bit will clear the Suspend Interrupt Enable bit and disable the corresponding interrupt request.

Value	Description
0	The Suspend interrupt is disabled.
1	The Suspend interrupt is enabled and an interrupt request will be generated when the Suspend interrupt Flag is set.

38.7.10 Device Interrupt Enable Set

Name: INTENSET
Offset: 0x18
Reset: 0x0000
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	15	14	13	12	11	10	9	8
							LPMSUSP	LPMNYET
Access							R/W	R/W
Reset							0	0

Bit	7	6	5	4	3	2	1	0
	RAMACER	UPRSM	EORSM	WAKEUP	EORST	SOF		SUSPEND
Access	R/W	R/W	R/W	R/W	R/W	R/W		R/W
Reset	0	0	0	0	0	0		0

Bit 9 – LPMSUSP Link Power Management Suspend Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will set the Link Power Management Suspend Enable bit and enable the corresponding interrupt request.

Value	Description
0	The Link Power Management Suspend interrupt is disabled.
1	The Link Power Management Suspend interrupt is enabled.

Bit 8 – LPMNYET Link Power Management Not Yet Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will set the Link Power Management Not Yet interrupt bit and enable the corresponding interrupt request.

Value	Description
0	The Link Power Management Not Yet interrupt is disabled.
1	The Link Power Management Not Yet interrupt is enabled.

Bit 7 – RAMACER RAM Access Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will set the RAM Access Enable bit and enable the corresponding interrupt request.

Value	Description
0	The RAM Access interrupt is disabled.
1	The RAM Access interrupt is enabled.

Bit 6 – UPRSM Upstream Resume Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will set the Upstream Resume Enable bit and enable the corresponding interrupt request.

Value	Description
0	The Upstream Resume interrupt is disabled.
1	The Upstream Resume interrupt is enabled.

Bit 5 – EORSM End Of Resume Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will set the End Of Resume interrupt Enable bit and enable the corresponding interrupt request.

Value	Description
0	The End Of Resume interrupt is disabled.
1	The End Of Resume interrupt is enabled.

Bit 4 – WAKEUP Wake-Up Interrupt Enable

Writing a zero to this bit has no effect.
Writing a one to this bit will set the Wake Up interrupt Enable bit and enable the corresponding interrupt request.

Value	Description
0	The Wake Up interrupt is disabled.
1	The Wake Up interrupt is enabled.

Bit 3 – EORST End of Reset Interrupt Enable

Writing a zero to this bit has no effect.
Writing a one to this bit will set the End of Reset interrupt Enable bit and enable the corresponding interrupt request.

Value	Description
0	The End of Reset interrupt is disabled.
1	The End of Reset interrupt is enabled.

Bit 2 – SOF Start-of-Frame Interrupt Enable

Writing a zero to this bit has no effect.
Writing a one to this bit will set the Start-of-Frame interrupt Enable bit and enable the corresponding interrupt request.

Value	Description
0	The Start-of-Frame interrupt is disabled.
1	The Start-of-Frame interrupt is enabled.

Bit 0 – SUSPEND Suspend Interrupt Enable

Writing a zero to this bit has no effect.
Writing a one to this bit will set the Suspend interrupt Enable bit and enable the corresponding interrupt request.

Value	Description
0	The Suspend interrupt is disabled.
1	The Suspend interrupt is enabled.

38.7.11 Device Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x01C
Reset: 0x0000
Property: -

Bit	15	14	13	12	11	10	9	8
							LPMSUSP	LPMNYET
Access							R/W	R/W
Reset							0	0

Bit	7	6	5	4	3	2	1	0
	RAMACER	UPRSM	EORSM	WAKEUP	EORST	SOF		SUSPEND
Access	R/W	R/W	R/W	R/W	R/W	R/W		R/W
Reset	0	0	0	0	0	0		0

Bit 9 – LPMSUSP Link Power Management Suspend Interrupt Flag
This flag is cleared by writing a one to the flag.
This flag is set when the USB module acknowledge a Link Power Management Transaction (ACK handshake) and has entered the Suspended state and will generate an interrupt if INTENSET.LPMSUSP is one.
Writing a zero to this bit has no effect.
Writing a one to this bit clears the LPMSUSP Interrupt Flag.

Bit 8 – LPMNYET Link Power Management Not Yet Interrupt Flag
This flag is cleared by writing a one to the flag.
This flag is set when the USB module acknowledges a Link Power Management Transaction (handshake is NYET) and will generate an interrupt if INTENSET.LPMNYET is one.
Writing a zero to this bit has no effect.
Writing a one to this bit clears the LPMNYET Interrupt Flag.

Bit 7 – RAMACER RAM Access Interrupt Flag
This flag is cleared by writing a one to the flag.
This flag is set when a RAM access underflow error occurs during IN data stage. This bit will generate an interrupt if INTENSET.RAMACER is one.
Writing a zero to this bit has no effect.

Bit 6 – UPRSM Upstream Resume Interrupt Flag
This flag is cleared by writing a one to the flag.
This flag is set when the USB sends a resume signal called “Upstream Resume” and will generate an interrupt if INTENSET.UPRSM is one.
Writing a zero to this bit has no effect.

Bit 5 – EORSM End Of Resume Interrupt Flag
This flag is cleared by writing a one to the flag.
This flag is set when the USB detects a valid “End of Resume” signal initiated by the host and will generate an interrupt if INTENSET.EORSM is one.
Writing a zero to this bit has no effect.

Bit 4 – WAKEUP Wake Up Interrupt Flag
This flag is cleared by writing a one to the flag.
This flag is set when the USB is reactivated by a filtered non-idle signal from the lines and will generate an interrupt if INTENSET.WAKEUP is one.
Writing a zero to this bit has no effect.

Bit 3 – EORST End of Reset Interrupt Flag

This flag is cleared by writing a one to the flag.
This flag is set when a USB “End of Reset” has been detected and will generate an interrupt if INTENSET.EORST is one.
Writing a zero to this bit has no effect.

Bit 2 – SOF Start-of-Frame Interrupt Flag

This flag is cleared by writing a one to the flag.
This flag is set when a USB “Start-of-Frame” has been detected (every 1 ms) and will generate an interrupt if INTENSET.SOF is one.
The FNUM is updated.
Writing a zero to this bit has no effect.

Bit 0 – SUSPEND Suspend Interrupt Flag

This flag is cleared by writing a one to the flag.
This flag is set when a USB “Suspend” idle state has been detected for 3 frame periods (J state for 3 ms) and will generate an interrupt if INTENSET.SUSPEND is one.
Writing a zero to this bit has no effect.

38.7.12 Endpoint Interrupt Summary

Name: EPINTSMRY
Offset: 0x20
Reset: 0x0000
Property: -

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	EPINT7	EPINT6	EPINT5	EPINT4	EPINT3	EPINT2	EPINT1	EPINT0
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 – EPINT EndPoint Interrupt
The flag EPINT[n] is set when an interrupt is triggered by the EndPoint n. See 38.7.19. EPINTFLAGn register in the device EndPoint section.
This bit will be cleared when no interrupts are pending for EndPoint n.

38.7.13 Descriptor Address

Name: DESCADD
Offset: 0x24
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
	DESCADD[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
	DESCADD[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
	DESCADD[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
	DESCADD[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 – DESCADD[31:0] Descriptor Address Value
These bits define the base address of the main USB descriptor in RAM. The two least significant bits must be written to zero.

38.7.14 Pad Calibration

Name: PADCAL
Offset: 0x28
Reset: 0x0000
Property: PAC Write-Protection

The Pad Calibration values must be loaded from the NVM Software Calibration Area into the USB Pad Calibration register by software, before enabling the USB, to achieve the specified accuracy.

Bit	15	14	13	12	11	10	9	8
		TRIM[2:0]				TRANSN[4:2]		
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
	TRANSN[1:0]			TRANSP[4:0]				
Access	R/W	R/W		R/W	R/W	R/W	R/W	R/W
Reset	0	0		0	0	0	0	0

Bits 14:12 – TRIM[2:0] Trim bits for DP/DM
These bits calibrate the matching of rise/fall of DP/DM.

Bits 10:6 – TRANSN[4:0] Trimmable Output Driver Impedance N
These bits calibrate the NMOS output impedance of DP/DM drivers.

Bits 4:0 – TRANSP[4:0] Trimmable Output Driver Impedance P
These bits calibrate the PMOS output impedance of DP/DM drivers.

38.7.15 Device Endpoint Configuration register n

Name: EPCFGn
Offset: 0x0100 + n*0x20 [n=0..7]
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
		EPTYPE1[2:0]				EPTYPE0[2:0]		
Access		R/W	R/W	R/W		R/W	R/W	R/W
Reset		0	0	0		0	0	0

Bits 6:4 – EPTYPE1[2:0] Endpoint Type for IN direction
These bits contains the endpoint type for IN direction.
Upon receiving a USB reset EPCFGn.EPTYPE1 is cleared except for endpoint 0 which is unchanged.

Value	Description
0x0	Bank1 is disabled.
0x1	Bank1 is enabled and configured as Control IN.
0x2	Bank1 is enabled and configured as Isochronous IN.
0x3	Bank1 is enabled and configured as Bulk IN.
0x4	Bank1 is enabled and configured as Interrupt IN.
0x5	Bank1 is enabled and configured as Dual-Bank OUT (Endpoint type is the same as the one defined in EPTYPE0)
0x6-0x7	Reserved

Bits 2:0 – EPTYPE0[2:0] Endpoint Type for OUT direction
These bits contains the endpoint type for OUT direction.
Upon receiving a USB reset EPCFGn.EPTYPE0 is cleared except for endpoint 0 which is unchanged.

Value	Description
0x0	Bank0 is disabled.
0x1	Bank0 is enabled and configured as Control SETUP / Control OUT.
0x2	Bank0 is enabled and configured as Isochronous OUT.
0x3	Bank0 is enabled and configured as Bulk OUT.
0x4	Bank0 is enabled and configured as Interrupt OUT.
0x5	Bank0 is enabled and configured as Dual Bank IN (Endpoint type is the same as the one defined in EPTYPE1)
0x6-0x7	Reserved

38.7.16 EndPoint Status Clear n

Name: EPSTATUSCLRn
Offset: 0x0104 + n*0x20 [n=0..7]
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
	BK1RDY	BK0RDY	STALLRQ1	STALLRQ0		CURBK	DTGLIN	DTGLOUT
Access	W	W	W	W		W	W	W
Reset	0	0	0	0		0	0	0

Bit 7 – BK1RDY Bank 1 Ready Clear
Writing a zero to this bit has no effect.
Writing a one to this bit will clear EPSTATUS.BK1RDY bit.

Bit 6 – BK0RDY Bank 0 Ready Clear
Writing a zero to this bit has no effect.
Writing a one to this bit will clear EPSTATUS.BK0RDY bit.

Bit 5 – STALLRQ1 STALL bank 1 Request Clear
Writing a zero to this bit has no effect.
Writing a one to this bit will clear EPSTATUS.STALLRQ1 bit.

Bit 4 – STALLRQ0 STALL bank 0 Request Clear
Writing a zero to this bit has no effect.
Writing a one to this bit will clear EPSTATUS.STALLRQ0 bit.

Bit 2 – CURBK Current Bank Clear
Writing a zero to this bit has no effect.
Writing a one to this bit will clear EPSTATUS.CURBK bit.

Bit 1 – DTGLIN Data Toggle IN Clear
Writing a zero to this bit has no effect.
Writing a one to this bit will clear EPSTATUS.DTGLIN bit.

Bit 0 – DTGLOUT Data Toggle OUT Clear
Writing a zero to this bit has no effect.
Writing a one to this bit will clear the EPSTATUS.DTGLOUT bit.

38.7.17 EndPoint Status Set n

Name: EPSTATUSSETn
Offset: 0x0105 + n*0x20 [n=0..7]
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
	BK1RDY	BK0RDY	STALLRQ1	STALLRQ0		CURBK	DTGLIN	DTGLOUT
Access	W	W	W	W		W	W	W
Reset	0	0	0	0		0	0	0

Bit 7 – BK1RDY Bank 1 Ready Set
Writing a zero to this bit has no effect.
Writing a one to this bit will set EPSTATUS.BK1RDY bit.

Bit 6 – BK0RDY Bank 0 Ready Set
Writing a zero to this bit has no effect.
Writing a one to this bit will set EPSTATUS.BK0RDY bit.

Bit 5 – STALLRQ1 STALL Request bank 1 Set
Writing a zero to this bit has no effect.
Writing a one to this bit will set EPSTATUS.STALLRQ1 bit.

Bit 4 – STALLRQ0 STALL Request bank 0 Set
Writing a zero to this bit has no effect.
Writing a one to this bit will set EPSTATUS.STALLRQ0 bit.

Bit 2 – CURBK Current Bank Set
Writing a zero to this bit has no effect.
Writing a one to this bit will set EPSTATUS.CURBK bit.

Bit 1 – DTGLIN Data Toggle IN Set
Writing a zero to this bit has no effect.
Writing a one to this bit will set EPSTATUS.DTGLIN bit.

Bit 0 – DTGLOUT Data Toggle OUT Set
Writing a zero to this bit has no effect.
Writing a one to this bit will set the EPSTATUS.DTGLOUT bit.

38.7.18 EndPoint Status n

Name: EPSTATUSn
Offset: 0x0106 + n*0x20 [n=0..7]
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
	BK1RDY	BK0RDY	STALLRQ1	STALLRQ0		CURBK	DTGLIN	DTGLOUT
Access	R	R	R	R		R	R	R
Reset	0	0	0	0		0	0	0

Bit 7 – BK1RDY Bank 1 is ready
Writing a one to the bit EPSTATUSCLR.BK1RDY will clear this bit.
Writing a one to the bit EPSTATUSSET.BK1RDY will set this bit.

Value	Description
0	The bank number 1 is not ready : For IN direction Endpoints, the bank is not yet filled in. For Control/OUT direction Endpoints, the bank is empty.
1	The bank number 1 is ready: For IN direction Endpoints, the bank is filled in. For Control/OUT direction Endpoints, the bank is full.

Bit 6 – BK0RDY Bank 0 is ready
Writing a one to the bit EPSTATUSCLR.BK0RDY will clear this bit.
Writing a one to the bit EPSTATUSSET.BK0RDY will set this bit.

Value	Description
0	The bank number 0 is not ready : For IN direction Endpoints, the bank is not yet filled in. For Control/OUT direction Endpoints, the bank is empty.
1	The bank number 0 is ready: For IN direction Endpoints, the bank is filled in. For Control/OUT direction Endpoints, the bank is full.

Bits 4, 5 – STALLRQ STALL bank x request
Writing a zero to the bit EPSTATUSCLR.STALLRQ will clear this bit.
Writing a one to the bit EPSTATUSSET.STALLRQ will set this bit.
This bit is cleared by hardware when receiving a SETUP packet.

Value	Description
0	Disable STALLRQx feature.
1	Enable STALLRQx feature: a STALL handshake will be sent to the host in regards to bank x.

Bit 2 – CURBK Current Bank
Writing a zero to the bit EPSTATUSCLR.CURBK will clear this bit.
Writing a one to the bit EPSTATUSSET.CURBK will set this bit.

Value	Description
0	The bank0 is the bank that will be used in the next single/multi USB packet.
1	The bank1 is the bank that will be used in the next single/multi USB packet.

Bit 1 – DTGLIN Data Toggle IN Sequence
Writing a zero to the bit EPSTATUSCLR.DTGLINCLR will clear this bit.
Writing a one to the bit EPSTATUSSET.DTGLINSET will set this bit.

Value	Description
0	The PID of the next expected IN transaction will be zero: data 0.
1	The PID of the next expected IN transaction will be one: data 1.

Bit 0 – DTGLOUT Data Toggle OUT Sequence
Writing a zero to the bit EPSTATUSCLR.DTGLOUTCLR will clear this bit.
Writing a one to the bit EPSTATUSSET.DTGLOUTSET will set this bit.

Value	Description
0	The PID of the next expected OUT transaction will be zero: data 0.
1	The PID of the next expected OUR transaction will be one: data 1.

38.7.19 Device EndPoint Interrupt Flag n

Name: EPINTFLAGn
Offset: 0x0107 + n*0x20 [n=0..7]
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
		STALL1	STALL0	RXSTP	TRFAIL1	TRFAIL0	TRCPT1	TRCPT0
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bits 5, 6 – STALL Transmit Stall x Interrupt Flag
This flag is cleared by writing a one to the flag.
This flag is set when a Transmit Stall occurs and will generate an interrupt if EPINTENCLR/SET.STALL is one.
EPINTFLAG.STALL is set for a single bank OUT endpoint or double bank IN/OUT endpoint when current bank is "0".
Writing a zero to this bit has no effect.
Writing a one to this bit clears the STALL Interrupt Flag.

Bit 4 – RXSTP Received Setup Interrupt Flag
This flag is cleared by writing a one to the flag.
This flag is set when a Received Setup occurs and will generate an interrupt if EPINTENCLR/SET.RXSTP is one.
Writing a zero to this bit has no effect.
Writing a one to this bit clears the RXSTP Interrupt Flag.

Bits 2, 3 – TRFAIL Transfer Fail x Interrupt Flag
This flag is cleared by writing a one to the flag.
This flag is set when a transfer fail occurs and will generate an interrupt if EPINTENCLR/SET.TRFAIL is one.
EPINTFLAG.TRFAIL is set for a single bank OUT endpoint or double bank IN/OUT endpoint when current bank is "0".
Writing a zero to this bit has no effect.
Writing a one to this bit clears the TRFAIL Interrupt Flag.

Bits 0, 1 – TRCPT Transfer Complete x interrupt Flag
This flag is cleared by writing a one to the flag.
This flag is set when a Transfer complete occurs and will generate an interrupt if EPINTENCLR/SET.TRCPT is one.
EPINTFLAG.TRCPT is set for a single bank OUT endpoint or double bank IN/OUT endpoint when current bank is "0".
Writing a zero to this bit has no effect.
Writing a one to this bit clears the TRCPT Interrupt Flag.

38.7.20 Device EndPoint Interrupt Clear n

Name: EPINTENCLRn
Offset: 0x0108 + n*0x20 [n=0..7]
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 Endpoint Interrupt Enable Set (EPINTENSET) register.

Bit	7	6	5	4	3	2	1	0
		STALL1	STALL0	RXSTP	TRFAIL1	TRFAIL0	TRCPT1	TRCPT0
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bits 5, 6 – STALL Transmit STALL x Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will clear the Transmit Stall x Interrupt Enable bit and disable the corresponding interrupt request.

Value	Description
0	The Transmit Stall x interrupt is disabled.
1	The Transmit Stall x interrupt is enabled and an interrupt request will be generated when the Transmit Stall x Interrupt Flag is set.

Bit 4 – RXSTP Received Setup Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will clear the Received Setup Interrupt Enable bit and disable the corresponding interrupt request.

Value	Description
0	The Received Setup interrupt is disabled.
1	The Received Setup interrupt is enabled and an interrupt request will be generated when the Received Setup Interrupt Flag is set.

Bits 2, 3 – TRFAIL Transfer Fail x Interrupt Enable
The user should look into the descriptor table status located in ram to be informed about the error condition : ERRORFLOW, CRC.
Writing a zero to this bit has no effect.
Writing a one to this bit will clear the Transfer Fail x Interrupt Enable bit and disable the corresponding interrupt request.

Value	Description
0	The Transfer Fail bank x interrupt is disabled.
1	The Transfer Fail bank x interrupt is enabled and an interrupt request will be generated when the Transfer Fail x Interrupt Flag is set.

Bits 0, 1 – TRCPT Transfer Complete x interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will clear the Transfer Complete x interrupt Enable bit and disable the corresponding interrupt request.

Value	Description
0	The Transfer Complete bank x interrupt is disabled.
1	The Transfer Complete bank x interrupt is enabled and an interrupt request will be generated when the Transfer Complete x Interrupt Flag is set.

38.7.21 Device Endpoint Interrupt Set n

Name: EPINTENSETn
Offset: 0x0109 + n*0x20 [n=0..7]
Reset: 0x0000
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 Endpoint Interrupt Enable Set (EPINTENCLR) register. This register is cleared by USB reset or when EPEN[n] is zero.

Bit	7	6	5	4	3	2	1	0
		STALL1	STALL0	RXSTP	TRFAIL1	TRFAIL0	TRCPT1	TRCPT0
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bits 5, 6 – STALL Transmit Stall x Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will enable the Transmit bank x Stall interrupt.

Value	Description
0	The Transmit Stall x interrupt is disabled.
1	The Transmit Stall x interrupt is enabled.

Bit 4 – RXSTP Received Setup Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will enable the Received Setup interrupt.

Value	Description
0	The Received Setup interrupt is disabled.
1	The Received Setup interrupt is enabled.

Bits 2, 3 – TRFAIL Transfer Fail bank x Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will enable the Transfer Fail interrupt.

Value	Description
0	The Transfer Fail interrupt is disabled.
1	The Transfer Fail interrupt is enabled.

Bits 0, 1 – TRCPT Transfer Complete bank x interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will enable the Transfer Complete x interrupt.
0.2.4 Device Registers - Endpoint RAM

Value	Description
0	The Transfer Complete bank x interrupt is disabled.
1	The Transfer Complete bank x interrupt is enabled.

38.8 Register Summary - USB Device Endpoint n Descriptor Bank 0

This Register Description section is valid if the USB is in Device mode (CTRLA.MODE=0).

Refer to the Registers Description section for more details on register properties and access permissions.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	ADDR	31:24	ADDR[31:24]							
		23:16	ADDR[23:16]							
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0x04	PCKSIZE	31:24	AUTO_ZLP	SIZE[2:0]			MULTI_PACKET_SIZE[13:10]			
		23:16	MULTI_PACKET_SIZE[9:2]							
		15:8	MULTI_PACKET_SIZE[1:0]		BYTE_COUNT[13:8]					
		7:0	BYTE_COUNT[7:0]							
0x08	EXTREG	15:8		VARIABLE[10:4]						
		7:0	VARIABLE[3:0]				SUBPID[3:0]			
0x0A	STATUS_BK	7:0							ERRORFLOW	CRCERR

38.8.1 Address of Data Buffer

Name: ADDR
Offset: 0x00
Reset: -
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	x	x	x	x	x	x	x	x

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	x	x	x	x	x	x	x	x

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	x	x	x	x	x	x	x	x

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	x	x	x	x	x	x	x	x

Bits 31:0 – ADDR[31:0] Data Pointer Address Value
These bits define the data pointer address as an absolute word address in RAM. The two least significant bits must be zero to ensure the start address is 32-bit aligned.

38.8.2 Packet Size

Name: PCKSIZE
Offset: 0x04
Reset: -
Property: -

Bit	31	30	29	28	27	26	25	24
	AUTO_ZLP	SIZE[2:0]			MULTI_PACKET_SIZE[13:10]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	x	0	0	x	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	MULTI_PACKET_SIZE[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	15	14	13	12	11	10	9	8
	MULTI_PACKET_SIZE[1:0]		BYTE_COUNT[13:8]					
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	x	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	BYTE_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	x

Bit 31 – AUTO_ZLP Automatic Zero Length Packet

This bit defines the automatic Zero Length Packet mode of the endpoint.
When enabled, the USB module will manage the ZLP handshake by hardware. This bit is for IN endpoints only. When disabled the handshake should be managed by firmware.

Value	Description
0	Automatic Zero Length Packet is disabled.
1	Automatic Zero Length Packet is enabled.

Bits 30:28 – SIZE[2:0] Endpoint size

These bits contains the maximum packet size of the endpoint.

Value	Description
0x0	8 Byte
0x1	16 Byte
0x2	32 Byte
0x3	64 Byte
0x4	128 Byte(1)
0x5	256 Byte(1)
0x6	512 Byte(1)
0x7	1023 Byte(1)

(1) For Isochronous endpoints only.

Bits 27:14 – MULTI_PACKET_SIZE[13:0] Multiple Packet Size

These bits define the 14-bit value that is used for multi-packet transfers.
For IN endpoints, MULTI_PACKET_SIZE holds the total number of bytes sent. MULTI_PACKET_SIZE should be written to zero when setting up a new transfer.
For OUT endpoints, MULTI_PACKET_SIZE holds the total data size for the complete transfer. This value must be a multiple of the maximum packet size.

Bits 13:0 – BYTE_COUNT[13:0] Byte Count

These bits define the 14-bit value that is used for the byte count.
For IN endpoints, BYTE_COUNT holds the number of bytes to be sent in the next IN transaction.
For OUT endpoint or SETUP endpoints, BYTE_COUNT holds the number of bytes received upon the last OUT or SETUP transaction.

38.8.3 Extended Register

Name: EXTREG
Offset: 0x08
Reset: -
Property: -

Bit	15	14	13	12	11	10	9	8
		VARIABLE[10:4]						
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
	VARIABLE[3:0]				SUBPID[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	x	0	0	0	x

Bits 14:4 – VARIABLE[10:0] Variable field send with extended token
These bits define the VARIABLE field of a received extended token. These bits are updated when the USB has answered by an handshake token ACK to a LPM transaction. See Section 2.1.1 Protocol Extension Token in the reference document “ENGINEERING CHANGE NOTICE, USB 2.0 Link Power Management Addendum”.
To support the USB2.0 Link Power Management addition the VARIABLE field should be read as described below.

VARIABLES	Description
VARIABLE[3:0]	bLinkState (1)
VARIABLE[7:4]	BESL (2)
VARIABLE[8]	bRemoteWake (1)
VARIABLE[10:9]	Reserved

1. For a definition of LPM Token bRemoteWake and bLinkState fields, refer to "Table 2-3 in the reference document ENGINEERING CHANGE NOTICE, USB 2.0 Link Power Management Addendum".
2. For a definition of LPM Token BESL field, refer to "Table 2-3 in the reference document ENGINEERING CHANGE NOTICE, USB 2.0 Link Power Management Addendum" and "Table X-X1 in Errata for ECN USB 2.0 Link Power Management.

Bits 3:0 – SUBPID[3:0] SUBPID field send with extended token
These bits define the SUBPID field of a received extended token. These bits are updated when the USB has answered by an handshake token ACK to a LPM transaction. See Section 2.1.1 Protocol Extension Token in the reference document “ENGINEERING CHANGE NOTICE, USB 2.0 Link Power Management Addendum”.

38.8.4 Device Status Bank

Name: STATUS_BK
Offset: 0x0A
Reset: -
Property: -

Bit	7	6	5	4	3	2	1	0
							ERRORFLOW	CRCERR
Access							R/W	R/W
Reset							x	x

Bit 1 – ERRORFLOW Error Flow Status

This bit defines the Error Flow Status.
This bit is set when a Error Flow has been detected during transfer from/towards this bank.
For OUT transfer, a NAK handshake has been sent.
For Isochronous OUT transfer, an overrun condition has occurred.
For IN transfer, this bit is not valid. EPSTATUS.TRFAIL0 and EPSTATUS.TRFAIL1 should reflect the flow errors.

Value	Description
0	No Error Flow detected.
1	A Error Flow has been detected.

Bit 0 – CRCERR CRC Error

This bit defines the CRC Error Status.
This bit is set when a CRC error has been detected in an isochronous OUT endpoint bank.
0.2.5 Host Registers - Common

Value	Description
0	No CRC Error.
1	CRC Error detected.

38.9 Register Summary - USB Device Endpoint Descriptor Bank 1

This Register Description section is valid if the USB is in Device mode (CTRLA.MODE=0).

Refer to the Registers Description section for more details on register properties and access permissions.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00 ... 0x0F	Reserved									
0x10	ADDR	31:24	ADDR[31:24]							
		23:16	ADDR[23:16]							
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0x14	PCKSIZE	31:24	AUTO_ZLP	SIZE[2:0]			MULTI_PACKET_SIZE[13:10]			
		23:16	MULTI_PACKET_SIZE[9:2]							
		15:8	MULTI_PACKET_SIZE[1:0]		BYTE_COUNT[13:8]					
		7:0	BYTE_COUNT[7:0]							
0x18 ... 0x19	Reserved									
0x1A	STATUS_BK	7:0							ERRORFLOW	CRCERR

38.9.1 Address of Data Buffer

Name: ADDR
Offset: 0x10
Reset: -
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	x	x	x	x	x	x	x	x

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	x	x	x	x	x	x	x	x

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	x	x	x	x	x	x	x	x

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	x	x	x	x	x	x	x	x

Bits 31:0 – ADDR[31:0] Data Pointer Address Value
These bits define the data pointer address as an absolute word address in RAM. The two least significant bits must be zero to ensure the start address is 32-bit aligned.

38.9.2 Packet Size

Name: PCKSIZE
Offset: 0x14
Reset: -
Property: -

Bit	31	30	29	28	27	26	25	24
	AUTO_ZLP	SIZE[2:0]			MULTI_PACKET_SIZE[13:10]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	x	0	0	x	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	MULTI_PACKET_SIZE[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	15	14	13	12	11	10	9	8
	MULTI_PACKET_SIZE[1:0]		BYTE_COUNT[13:8]					
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	x	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	BYTE_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	x

Bit 31 – AUTO_ZLP Automatic Zero Length Packet

This bit defines the automatic Zero Length Packet mode of the endpoint.
When enabled, the USB module will manage the ZLP handshake by hardware. This bit is for IN endpoints only. When disabled the handshake should be managed by firmware.

Value	Description
0	Automatic Zero Length Packet is disabled.
1	Automatic Zero Length Packet is enabled.

Bits 30:28 – SIZE[2:0] Endpoint size

These bits contains the maximum packet size of the endpoint.

Value	Description
0x0	8 Byte
0x1	16 Byte
0x2	32 Byte
0x3	64 Byte
0x4	128 Byte(1)
0x5	256 Byte(1)
0x6	512 Byte(1)
0x7	1023 Byte(1)

(1) For Isochronous endpoints only.

Bits 27:14 – MULTI_PACKET_SIZE[13:0] Multiple Packet Size

These bits define the 14-bit value that is used for multi-packet transfers.
For IN endpoints, MULTI_PACKET_SIZE holds the total number of bytes sent. MULTI_PACKET_SIZE should be written to zero when setting up a new transfer.
For OUT endpoints, MULTI_PACKET_SIZE holds the total data size for the complete transfer. This value must be a multiple of the maximum packet size.

Bits 13:0 – BYTE_COUNT[13:0] Byte Count

These bits define the 14-bit value that is used for the byte count.
For IN endpoints, BYTE_COUNT holds the number of bytes to be sent in the next IN transaction.
For OUT endpoint or SETUP endpoints, BYTE_COUNT holds the number of bytes received upon the last OUT or SETUP transaction.

38.9.3 Device Status Bank

Name: STATUS_BK
Offset: 0x1A
Reset: -
Property: -

Bit	7	6	5	4	3	2	1	0
							ERRORFLOW	CRCERR
Access							R/W	R/W
Reset							x	x

Bit 1 – ERRORFLOW Error Flow Status

This bit defines the Error Flow Status.
This bit is set when a Error Flow has been detected during transfer from/towards this bank.
For OUT transfer, a NAK handshake has been sent.
For Isochronous OUT transfer, an overrun condition has occurred.
For IN transfer, this bit is not valid. EPSTATUS.TRFAIL0 and EPSTATUS.TRFAIL1 should reflect the flow errors.

Value	Description
0	No Error Flow detected.
1	A Error Flow has been detected.

Bit 0 – CRCERR CRC Error

This bit defines the CRC Error Status.
This bit is set when a CRC error has been detected in an isochronous OUT endpoint bank.
0.2.5 Host Registers - Common

Value	Description
0	No CRC Error.
1	CRC Error detected.

38.10 Register Summary - USB Host

This Register Description section is valid if the USB is in Host mode (CTRLA.MODE=1).

Refer to the Registers Description section for more details on register properties and access permissions.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRLA	7:0	MODE					RUNSTDBY	ENABLE	SWRST
0x01	Reserved									
0x02	SYNCBUSY	7:0							ENABLE	SWRST
0x03	QOSCTRL	7:0					DQOS[1:0]		CQOS[1:0]	
0x04 ... 0x07	Reserved									
0x08	CTRLB	15:8					L1RESUME	VBUSOK	BUSRESET	SOFE
		7:0				AUTORESUME	SPDCONF[1:0]		RESUME	
0x0A	HSOFC	7:0	FLENCE				FLENC[3:0]			
0x0B	Reserved									
0x0C	STATUS	7:0	LINESTATE[1:0]				SPEED[1:0]			
0x0D	FSMSTATUS	7:0		FSMSTATE[6:0]						
0x0E ... 0x0F	Reserved									
0x10	FNUM	15:8			FNUM[10:5]					
		7:0	FNUM[4:0]							
0x12	FLENHIGH	7:0	FLENHIGH[7:0]							
0x13	Reserved									
0x14	INTENCLR	15:8							DDISC	DCONN
		7:0	RAMACER	UPRSM	DNRSM	WAKEUP	RST	HSOF		
0x16 ... 0x17	Reserved									
0x18	INTENSET	15:8							DDISC	DCONN
		7:0	RAMACER	UPRSM	DNRSM	WAKEUP	RST	HSOF		
0x1A ... 0x1B	Reserved									
0x1C	INTFLAG	15:8							DDISC	DCONN
		7:0	RAMACER	UPRSM	DNRSM	WAKEUP	RST	HSOF		
0x1E ... 0x1F	Reserved									
0x20	PINTSMRY	15:8								
		7:0	PINT7	PINT6	PINT5	PINT4	PINT3	PINT2	PINT1	PINT0
0x22 ... 0x23	Reserved									
0x24	DESCADD	31:24	DESCADD[31:24]							
		23:16	DESCADD[23:16]							
		15:8	DESCADD[15:8]							
		7:0	DESCADD[7:0]							
0x28	PADCAL	15:8		TRIM[2:0]				TRANSN[4:2]		
		7:0	TRANSN[1:0]			TRANSP[4:0]				
0x2A ... 0xFF	Reserved									
0x0100	PCFG0	7:0			PTYPE[2:0]			BK	PTOKEN[1:0]	
0x0101 ... 0x0102	Reserved									

.....continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0103	BINTERVAL0	7:0	BINTERVAL[7:0]							
0x0104	PSTATUSCLR0	7:0	BK1RDY	BK0RDY		PFREEZE		CURBK		DTGL
0x0105	PSTATUSSET0	7:0	BK1RDY	BK0RDY		PFREEZE		CURBK		DTGL
0x0106	PSTATUS0	7:0	BK1RDY	BK0RDY		PFREEZE		CURBK		DTGL
0x0107	PINTFLAG0	7:0			STALL	TXSTP	PERR	TRFAIL	TRCPT1	TRCPT0
0x0108	PINTENCLR0	7:0			STALL	TXSTP	PERR	TRFAIL	TRCPT1	TRCPT0
0x0109	PINTENSET0	7:0			STALL	TXSTP	PERR	TRFAIL	TRCPT1	TRCPT0
0x010A ... 0x011F	Reserved									
0x0120	PCFG1	7:0			PTYPE[2:0]			BK	PTOKEN[1:0]	
0x0121 ... 0x0122	Reserved									
0x0123	BINTERVAL1	7:0	BINTERVAL[7:0]							
0x0124	PSTATUSCLR1	7:0	BK1RDY	BK0RDY		PFREEZE		CURBK		DTGL
0x0125	PSTATUSSET1	7:0	BK1RDY	BK0RDY		PFREEZE		CURBK		DTGL
0x0126	PSTATUS1	7:0	BK1RDY	BK0RDY		PFREEZE		CURBK		DTGL
0x0127	PINTFLAG1	7:0			STALL	TXSTP	PERR	TRFAIL	TRCPT1	TRCPT0
0x0128	PINTENCLR1	7:0			STALL	TXSTP	PERR	TRFAIL	TRCPT1	TRCPT0
0x0129	PINTENSET1	7:0			STALL	TXSTP	PERR	TRFAIL	TRCPT1	TRCPT0
0x012A ... 0x013F	Reserved									
0x0140	PCFG2	7:0			PTYPE[2:0]			BK	PTOKEN[1:0]	
0x0141 ... 0x0142	Reserved									
0x0143	BINTERVAL2	7:0	BINTERVAL[7:0]							
0x0144	PSTATUSCLR2	7:0	BK1RDY	BK0RDY		PFREEZE		CURBK		DTGL
0x0145	PSTATUSSET2	7:0	BK1RDY	BK0RDY		PFREEZE		CURBK		DTGL
0x0146	PSTATUS2	7:0	BK1RDY	BK0RDY		PFREEZE		CURBK		DTGL
0x0147	PINTFLAG2	7:0			STALL	TXSTP	PERR	TRFAIL	TRCPT1	TRCPT0
0x0148	PINTENCLR2	7:0			STALL	TXSTP	PERR	TRFAIL	TRCPT1	TRCPT0
0x0149	PINTENSET2	7:0			STALL	TXSTP	PERR	TRFAIL	TRCPT1	TRCPT0
0x014A ... 0x015F	Reserved									
0x0160	PCFG3	7:0			PTYPE[2:0]			BK	PTOKEN[1:0]	
0x0161 ... 0x0162	Reserved									
0x0163	BINTERVAL3	7:0	BINTERVAL[7:0]							
0x0164	PSTATUSCLR3	7:0	BK1RDY	BK0RDY		PFREEZE		CURBK		DTGL
0x0165	PSTATUSSET3	7:0	BK1RDY	BK0RDY		PFREEZE		CURBK		DTGL
0x0166	PSTATUS3	7:0	BK1RDY	BK0RDY		PFREEZE		CURBK		DTGL
0x0167	PINTFLAG3	7:0			STALL	TXSTP	PERR	TRFAIL	TRCPT1	TRCPT0
0x0168	PINTENCLR3	7:0			STALL	TXSTP	PERR	TRFAIL	TRCPT1	TRCPT0
0x0169	PINTENSET3	7:0			STALL	TXSTP	PERR	TRFAIL	TRCPT1	TRCPT0
0x016A ... 0x017F	Reserved									
0x0180	PCFG4	7:0			PTYPE[2:0]			BK	PTOKEN[1:0]	
0x0181 ... 0x0182	Reserved									
0x0183	BINTERVAL4	7:0	BINTERVAL[7:0]							
0x0184	PSTATUSCLR4	7:0	BK1RDY	BK0RDY		PFREEZE		CURBK		DTGL
0x0185	PSTATUSSET4	7:0	BK1RDY	BK0RDY		PFREEZE		CURBK		DTGL

.....continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0186	PSTATUS4	7:0	BK1RDY	BK0RDY		PFREEZE		CURBK		DTGL
0x0187	PINTFLAG4	7:0			STALL	TXSTP	PERR	TRFAIL	TRCPT1	TRCPT0
0x0188	PINTENCLR4	7:0			STALL	TXSTP	PERR	TRFAIL	TRCPT1	TRCPT0
0x0189	PINTENSET4	7:0			STALL	TXSTP	PERR	TRFAIL	TRCPT1	TRCPT0
0x018A ... 0x019F	Reserved									
0x01A0	PCFG5	7:0			PTYPE[2:0]			BK	PTOKEN[1:0]	
0x01A1 ... 0x01A2	Reserved									
0x01A3	BINTERVAL5	7:0	BINTERVAL[7:0]							
0x01A4	PSTATUSCLR5	7:0	BK1RDY	BK0RDY		PFREEZE		CURBK		DTGL
0x01A5	PSTATUSSET5	7:0	BK1RDY	BK0RDY		PFREEZE		CURBK		DTGL
0x01A6	PSTATUS5	7:0	BK1RDY	BK0RDY		PFREEZE		CURBK		DTGL
0x01A7	PINTFLAG5	7:0			STALL	TXSTP	PERR	TRFAIL	TRCPT1	TRCPT0
0x01A8	PINTENCLR5	7:0			STALL	TXSTP	PERR	TRFAIL	TRCPT1	TRCPT0
0x01A9	PINTENSET5	7:0			STALL	TXSTP	PERR	TRFAIL	TRCPT1	TRCPT0
0x01AA ... 0x01BF	Reserved									
0x01C0	PCFG6	7:0			PTYPE[2:0]			BK	PTOKEN[1:0]	
0x01C1 ... 0x01C2	Reserved									
0x01C3	BINTERVAL6	7:0	BINTERVAL[7:0]							
0x01C4	PSTATUSCLR6	7:0	BK1RDY	BK0RDY		PFREEZE		CURBK		DTGL
0x01C5	PSTATUSSET6	7:0	BK1RDY	BK0RDY		PFREEZE		CURBK		DTGL
0x01C6	PSTATUS6	7:0	BK1RDY	BK0RDY		PFREEZE		CURBK		DTGL
0x01C7	PINTFLAG6	7:0			STALL	TXSTP	PERR	TRFAIL	TRCPT1	TRCPT0
0x01C8	PINTENCLR6	7:0			STALL	TXSTP	PERR	TRFAIL	TRCPT1	TRCPT0
0x01C9	PINTENSET6	7:0			STALL	TXSTP	PERR	TRFAIL	TRCPT1	TRCPT0
0x01CA ... 0x01DF	Reserved									
0x01E0	PCFG7	7:0			PTYPE[2:0]			BK	PTOKEN[1:0]	
0x01E1 ... 0x01E2	Reserved									
0x01E3	BINTERVAL7	7:0	BINTERVAL[7:0]							
0x01E4	PSTATUSCLR7	7:0	BK1RDY	BK0RDY		PFREEZE		CURBK		DTGL
0x01E5	PSTATUSSET7	7:0	BK1RDY	BK0RDY		PFREEZE		CURBK		DTGL
0x01E6	PSTATUS7	7:0	BK1RDY	BK0RDY		PFREEZE		CURBK		DTGL
0x01E7	PINTFLAG7	7:0			STALL	TXSTP	PERR	TRFAIL	TRCPT1	TRCPT0
0x01E8	PINTENCLR7	7:0			STALL	TXSTP	PERR	TRFAIL	TRCPT1	TRCPT0
0x01E9	PINTENSET7	7:0			STALL	TXSTP	PERR	TRFAIL	TRCPT1	TRCPT0

38.10.1 Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized Bits

Bit	7	6	5	4	3	2	1	0
	MODE					RUNSTDBY	ENABLE	SWRST
Access	R/W					R/W	R/W	R/W
Reset	0					0	0	0

Bit 7 – MODE Operating Mode
This bit defines the operating mode of the USB.

Value	Description
0	USB Device mode
1	USB Host mode

Bit 2 – RUNSTDBY Run in Standby Mode
This bit is Enable-Protected.

Value	Description
0	USB clock is stopped in standby mode.
1	USB clock is running in standby mode

Bit 1 – ENABLE Enable
Note: This bit is write-synchronized: SYNCBUSY.ENABLE must be checked to ensure the CTRLA.ENABLE synchronization is complete.

Value	Description
0	The peripheral is disabled or being disabled.
1	The peripheral is enabled or being enabled.

Bit 0 – SWRST Software Reset
Writing a zero to this bit has no effect.
Writing a '1' to this bit resets all registers in the USB, to their initial state, and the USB will be disabled.
Writing a '1' to CTRLA.SWRST will always take precedence, meaning that all other writes in the same write-operation will be discarded.
Notes:

1. When the CTRLA.SWRST is written, the user must poll SYNCBUSY.SWRST bit to know when the reset operation is complete.
2. 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.

38.10.2 Synchronization Busy

Name: SYNCBUSY
Offset: 0x02
Reset: 0x00
Property: -

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

Bit 1 – ENABLE Synchronization Enable status bit
This bit is cleared when the synchronization of ENABLE register between the clock domains is complete.
This bit is set when the synchronization of ENABLE register between clock domains is started.

Bit 0 – SWRST Synchronization Software Reset status bit
This bit is cleared when the synchronization of SWRST register between the clock domains is complete.
This bit is set when the synchronization of SWRST register between clock domains is started.

38.10.3 QOS Control

Name: QOSCTRL
Offset: 0x03
Reset: 0x0F
Property: PAC Write-Protection

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

Bits 3:2 – DQOS[1:0] Data Quality of Service
These bits define the memory priority access during the endpoint or pipe read/write data operation. Refer to SRAM Quality of Service.

Bits 1:0 – CQOS[1:0] Configuration Quality of Service
These bits define the memory priority access during the endpoint or pipe read/write configuration operation. Refer to SRAM Quality of Service.

38.10.4 Control B

Name: CTRLB
Offset: 0x08
Reset: 0x0000
Property: PAC Write-Protection

Bit	15	14	13	12	11	10	9	8
					L1RESUME	VBUSOK	BUSRESET	SOFE
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

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

Bit 11 – L1RESUME Send USB L1 Resume
Writing 0 to this bit has no effect.
1: Generates a USB L1 Resume on the USB bus. This bit should only be set when the Start-of-Frame generation is enabled (SOFE bit set). The duration of the USB L1 Resume is defined by the EXTREG.VARIABLE[7:4] bits field also known as BESL (See LPM ECN).See also 38.11.3. EXTREG Register.
This bit is cleared when the USB L1 Resume has been sent or when a USB reset is requested.

Bit 10 – VBUSOK VBUS is OK
This notifies the USB HOST that USB operations can be started. When this bit is zero and even if the USB HOST is configured and enabled, HOST operation is halted. Setting this bit will allow HOST operation when the USB is configured and enabled.

Value	Description
0	The USB module is notified that the VBUS on the USB line is not powered.
1	The USB module is notified that the VBUS on the USB line is powered.

Bit 9 – BUSRESET Send USB Reset

Value	Description
0	Reset generation is disabled. It is written to zero when the USB reset is completed or when a device disconnection is detected. Writing zero has no effect.
1	Generates a USB Reset on the USB bus.

Bit 8 – SOFE Start-of-Frame Generation Enable

Value	Description
0	The SOF generation is disabled and the USB bus is in suspend state.
1	Generates SOF on the USB bus in full speed and keep it alive in low speed mode. This bit is automatically set at the end of a USB reset (INTFLAG.RST) or at the end of a downstream resume (INTFLAG.DNRSM) or at the end of L1 resume.

Bit 4 – AUTORESUME Auto Resume Enable

Value	Description
0	The Auto Resume is disabled.
1	Enable Auto Resume

Bits 3:2 – SPDCONF[1:0] Speed Configuration for Host
These bits select the host speed configuration as shown below

Value	Description
0x0	Low and Full Speed capable
0x1	Reserved
0x2	Reserved

Value	Description
0x3	Reserved

Bit 1 – RESUME Send USB Resume
Writing 0 to this bit has no effect.
1: Generates a USB Resume on the USB bus.
This bit is cleared when the USB Resume has been sent or when a USB reset is requested.

38.10.5 Host Start-of-Frame Control

Name: HSOFC
Offset: 0x0A
Reset: 0x00
Property: PAC Write-Protection

During a very short period just before transmitting a Start-of-Frame, this register is locked. Thus, after writing, it is recommended to check the register value, and write this register again if necessary. This register is cleared upon a USB reset.

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

Bit 7 – FLENCE Frame Length Control Enable
When this bit is '1', the time between Start-of-Frames can be tuned by up to +/-0.06% using FLENC[3:0].
Note: In Low Speed mode, FLENCE must be '0'.

Value	Description
0	Start-of-Frame is generated every 1ms.
1	Start-of-Frame generation depends on the signed value of FLENC[3:0]. USB Start-of-Frame period equals 1ms + (FLENC[3:0]/12000)ms

Bits 3:0 – FLENC[3:0] Frame Length Control
These bits define the signed value of the 4-bit FLENC that is added to the Internal Frame Length when FLENCE is '1'. The internal Frame length is the top value of the frame counter when FLENCE is zero.

38.10.6 Status

Name: STATUS
Offset: 0x0C
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
	LINESTATE[1:0]				SPEED[1:0]			
Access	R	R			R	R		
Reset	0	0			0	0		

Bits 7:6 – LINESTATE[1:0] USB Line State Status
These bits define the current line state DP/DM.

LINESTATE[1:0]	USB Line Status
0x0	SE0/RESET
0x1	FS-J or LS-K State
0x2	FS-K or LS-J State

Bits 3:2 – SPEED[1:0] Speed Status
These bits define the current speed used by the host.

SPEED[1:0]	Speed Status
0x0	Full-speed mode
0x1	Low-speed mode
0x2	Reserved
0x3	Reserved

38.10.7 Finite State Machine Status

Name: FSMSTATUS
Offset: 0x0D
Reset: 0x01
Property: -

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

Bits 6:0 – FSMSTATE[6:0] Fine State Machine Status
These bits indicate the state of the finite state machine of the USB controller.

Value	Name	Description
0x01	OFF	Corresponds to the powered-off, disconnected, and disabled state.
0x02	ON	Corresponds to the Idle and Active states.
0x04	SUSPEND	
0x08	SLEEP	
0x10	DNRESUME	Down Stream Resume.
0x20	UPRESUME	Up Stream Resume.
0x40	RESET	USB lines Reset.
Others		Reserved

38.10.8 Host Frame Number

Name: FNUM
Offset: 0x10
Reset: 0x0000
Property: PAC Write-Protection

Bit	15	14	13	12	11	10	9	8
			FNUM[10:5]					
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
	FNUM[4:0]							
Access	R/W	R/W	R/W	R/W	R/W			
Reset	0	0	0	0	0			

Bits 13:3 – FNUM[10:0] Frame Number
These bits contains the current SOF number.
These bits can be written by software to initialize a new frame number value. In this case, at the next SOF, the FNUM field takes its new value.
As the FNUM register lies across two consecutive byte addresses, writing byte-wise (8-bits) to the FNUM register may produce incorrect frame number generation. It is recommended to write FNUM register word-wise (32-bits) or half-word-wise (16-bits).

38.10.9 Host Frame Length

Name: FLENHIGH
Offset: 0x12
Reset: 0x00
Property: -

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

Bits 7:0 – FLENHIGH[7:0] Frame Length
These bits contains the 8 high-order bits of the internal frame counter.

Table 38-2. Counter Description vs. Speed

Host Register STATUS.SPEED	Description
Full Speed	With a USB clock running at 12MHz, counter length is 12000 to ensure a SOF generation every 1 ms.

38.10.10 Host Interrupt Enable Register Clear

Name: INTENCLR
Offset: 0x14
Reset: 0x0000
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	15	14	13	12	11	10	9	8
							DDISC	DCONN
Access							R/W	R/W
Reset							0	0

Bit	7	6	5	4	3	2	1	0
	RAMACER	UPRSM	DNRSM	WAKEUP	RST	HSOF		
Access	R/W	R/W	R/W	R/W	R/W	R/W		
Reset	0	0	0	0	0	0		

Bit 9 – DDISC Device Disconnection Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will clear the Device Disconnection interrupt Enable bit and disable the corresponding interrupt request.

Value	Description
0	The Device Disconnection interrupt is disabled.
1	The Device Disconnection interrupt is enabled and an interrupt request will be generated when the Device Disconnection interrupt Flag is set.

Bit 8 – DCONN Device Connection Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will clear the Device Connection interrupt Enable bit and disable the corresponding interrupt request.

Value	Description
0	The Device Connection interrupt is disabled.
1	The Device Connection interrupt is enabled and an interrupt request will be generated when the Device Connection interrupt Flag is set.

Bit 7 – RAMACER RAM Access Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will clear the RAM Access interrupt Enable bit and disable the corresponding interrupt request.

Value	Description
0	The RAM Access interrupt is disabled.
1	The RAM Access interrupt is enabled and an interrupt request will be generated when the RAM Access interrupt Flag is set.

Bit 6 – UPRSM Upstream Resume from Device Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will clear the Upstream Resume interrupt Enable bit and disable the corresponding interrupt request.

Value	Description
0	The Upstream Resume interrupt is disabled.
1	The Upstream Resume interrupt is enabled and an interrupt request will be generated when the Upstream Resume interrupt Flag is set.

Bit 5 – DNRSM Down Resume Interrupt Enable

Writing a zero to this bit has no effect.
Writing a one to this bit will clear the Down Resume interrupt Enable bit and disable the corresponding interrupt request.

Value	Description
0	The Down Resume interrupt is disabled.
1	The Down Resume interrupt is enabled and an interrupt request will be generated when the Down Resume interrupt Flag is set.

Bit 4 – WAKEUP Wake Up Interrupt Enable

Writing a zero to this bit has no effect.
Writing a one to this bit will clear the Wake Up interrupt Enable bit and disable the corresponding interrupt request.

Value	Description
0	The Wake Up interrupt is disabled.
1	The Wake Up interrupt is enabled and an interrupt request will be generated when the Wake Up interrupt Flag is set.

Bit 3 – RST BUS Reset Interrupt Enable

Writing a zero to this bit has no effect.
Writing a one to this bit will clear the Bus Reset interrupt Enable bit and disable the corresponding interrupt request.

Value	Description
0	The Bus Reset interrupt is disabled.
1	The Bus Reset interrupt is enabled and an interrupt request will be generated when the Bus Reset interrupt Flag is set.

Bit 2 – HSOF Host Start-of-Frame Interrupt Enable

Writing a zero to this bit has no effect.
Writing a one to this bit will clear the Host Start-of-Frame interrupt Enable bit and disable the corresponding interrupt request.

Value	Description
0	The Host Start-of-Frame interrupt is disabled.
1	The Host Start-of-Frame interrupt is enabled and an interrupt request will be generated when the Host Start-of-Frame interrupt Flag is set.

38.10.11 Host Interrupt Enable Register Set

Name: INTENSET
Offset: 0x18
Reset: 0x0000
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	15	14	13	12	11	10	9	8
							DDISC	DCONN
Access							R/W	R/W
Reset							0	0

Bit	7	6	5	4	3	2	1	0
	RAMACER	UPRSM	DNRSM	WAKEUP	RST	HSOF		
Access	R/W	R/W	R/W	R/W	R/W	R/W		
Reset	0	0	0	0	0	0		

Bit 9 – DDISC Device Disconnection Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will set the Device Disconnection interrupt Enable bit and enable the corresponding interrupt request.

Value	Description
0	The Device Disconnection interrupt is disabled.
1	The Device Disconnection interrupt is enabled.

Bit 8 – DCONN Device Connection Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will set the Device Connection interrupt Enable bit and enable the corresponding interrupt request.

Value	Description
0	The Device Connection interrupt is disabled.
1	The Device Connection interrupt is enabled.

Bit 7 – RAMACER RAM Access Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will set the RAM Access interrupt Enable bit and enable the corresponding interrupt request.

Value	Description
0	The RAM Access interrupt is disabled.
1	The RAM Access interrupt is enabled.

Bit 6 – UPRSM Upstream Resume from the device Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will set the Upstream Resume interrupt Enable bit and enable the corresponding interrupt request.

Value	Description
0	The Upstream Resume interrupt is disabled.
1	The Upstream Resume interrupt is enabled.

Bit 5 – DNRSM Down Resume Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will set the Down Resume interrupt Enable bit and enable the corresponding interrupt request.

Value	Description
0	The Down Resume interrupt is disabled.

Value	Description
1	The Down Resume interrupt is enabled.

Bit 4 – WAKEUP Wake Up Interrupt Enable

Writing a zero to this bit has no effect.
Writing a one to this bit will set the Wake Up interrupt Enable bit and enable the corresponding interrupt request.

Value	Description
0	The WakeUp interrupt is disabled.
1	The WakeUp interrupt is enabled.

Bit 3 – RST Bus Reset Interrupt Enable

Writing a zero to this bit has no effect.
Writing a one to this bit will set the Bus Reset interrupt Enable bit and enable the corresponding interrupt request.

Value	Description
0	The Bus Reset interrupt is disabled.
1	The Bus Reset interrupt is enabled.

Bit 2 – HSOF Host Start-of-Frame Interrupt Enable

Writing a zero to this bit has no effect.
Writing a one to this bit will set the Host Start-of-Frame interrupt Enable bit and enable the corresponding interrupt request.

Value	Description
0	The Host Start-of-Frame interrupt is disabled.
1	The Host Start-of-Frame interrupt is enabled.

38.10.12 Host Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x1C
Reset: 0x0000
Property: -

Bit	15	14	13	12	11	10	9	8
							DDISC	DCONN
Access							R/W	R/W
Reset							0	0

Bit	7	6	5	4	3	2	1	0
	RAMACER	UPRSM	DNRSM	WAKEUP	RST	HSOF		
Access	R/W	R/W	R/W	R/W	R/W	R/W		
Reset	0	0	0	0	0	0		

Bit 9 – DDISC Device Disconnection Interrupt Flag
This flag is cleared by writing a one to the flag.
This flag is set when the device has been removed from the USB Bus and will generate an interrupt if INTENCLR/SET.DDISC is one.
Writing a zero to this bit has no effect.
Writing a one to this bit clears the DDISC Interrupt Flag.

Bit 8 – DCONN Device Connection Interrupt Flag
This flag is cleared by writing a one to the flag.
This flag is set when a new device has been connected to the USB BUS and will generate an interrupt if INTENCLR/SET.DCONN is one.
Writing a zero to this bit has no effect.
Writing a one to this bit clears the DCONN Interrupt Flag.

Bit 7 – RAMACER RAM Access Interrupt Flag
This flag is cleared by writing a one to the flag.
This flag is set when a RAM access error occurs during an OUT stage and will generate an interrupt if INTENCLR/SET.RAMACER is one.
Writing a zero to this bit has no effect.

Bit 6 – UPRSM Upstream Resume from the Device Interrupt Flag
This flag is cleared by writing a one to the flag.
This flag is set when the USB has received an Upstream Resume signal from the Device and will generate an interrupt if INTENCLR/SET.UPRSM is one.
Writing a zero to this bit has no effect.

Bit 5 – DNRSM Down Resume Interrupt Flag
This flag is cleared by writing a one to the flag.
This flag is set when the USB has sent a Down Resume and will generate an interrupt if INTENCLR/SET.DRSM is one.
Writing a zero to this bit has no effect.

Bit 4 – WAKEUP Wake Up Interrupt Flag
This flag is cleared by writing a one.
This flag is set when:

- The host controller is in suspend mode (SOFE is zero) and an upstream resume from the device is detected
- The host controller is in suspend mode (SOFE is zero) and an device disconnection is detected
- The host controller is in operational state (VBUSOK is one) and an device connection is detected

In all cases it will generate an interrupt if INTENCLR/SET.WAKEUP is one.
Writing a zero to this bit has no effect.

Bit 3 – RST Bus Reset Interrupt Flag
This flag is cleared by writing a one to the flag.
This flag is set when a Bus “Reset” has been sent to the Device and will generate an interrupt if INTENCLR/SET.RST is one.
Writing a zero to this bit has no effect.

Bit 2 – HSOF Host Start-of-Frame Interrupt Flag
This flag is cleared by writing a one to the flag.
This flag is set when a USB “Host Start-of-Frame” in Full Speed or a keep-alive in Low Speed has been sent (every 1 ms) and will generate an interrupt if INTENCLR/SET.HSOF is one.
The value of the FNUM register is updated.
Writing a zero to this bit has no effect.

38.10.13 Pipe Interrupt Summary

Name: PINTSMRY
Offset: 0x20
Reset: 0x0000
Property: -

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	PINT7	PINT6	PINT5	PINT4	PINT3	PINT2	PINT1	PINT0
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 – PINT

The flag PINT[n] is set when an interrupt is triggered by the pipe n. See 38.10.21. PINTFLAGn register in the Host Pipe Register section.
This bit will be cleared when there are no interrupts pending for Pipe n.
Writing to this bit has no effect.

38.10.14 Descriptor Address

Name: DESCADD
Offset: 0x24
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
	DESCADD[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
	DESCADD[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
	DESCADD[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
	DESCADD[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 – DESCADD[31:0] Descriptor Address Value
These bits define the base address of the main USB descriptor in RAM. The two least significant bits must be written to zero.

38.10.15 Pad Calibration

Name: PADCAL
Offset: 0x28
Reset: 0x0000
Property: PAC Write-Protection

The Pad Calibration values must be loaded from the NVM Software Calibration Area into the USB Pad Calibration register by software, before enabling the USB, to achieve the specified accuracy.

Bit	15	14	13	12	11	10	9	8
		TRIM[2:0]				TRANSN[4:2]		
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
	TRANSN[1:0]			TRANSP[4:0]				
Access	R/W	R/W		R/W	R/W	R/W	R/W	R/W
Reset	0	0		0	0	0	0	0

Bits 14:12 – TRIM[2:0] Trim bits for DP/DM
These bits calibrate the matching of rise/fall of DP/DM.

Bits 10:6 – TRANSN[4:0] Trimmable Output Driver Impedance N
These bits calibrate the NMOS output impedance of DP/DM drivers.

Bits 4:0 – TRANSP[4:0] Trimmable Output Driver Impedance P
These bits calibrate the PMOS output impedance of DP/DM drivers.

38.10.16 Host Pipe n Configuration

Name: PCFGn
Offset: 0x0100 + n*0x20 [n=0..7]
Reset: 0x00
Property: PAC Write-Protection

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

Bits 5:3 – PTYPE[2:0] Type of the Pipe
These bits contains the pipe type.

PTYPE[2:0]	Description
0x0	Pipe is disabled
0x1	Pipe is enabled and configured as CONTROL
0x2	Pipe is enabled and configured as ISO
0x3	Pipe is enabled and configured as BULK
0x4	Pipe is enabled and configured as INTERRUPT
0x5	Pipe is enabled and configured as EXTENDED
0x06-0x7	Reserved

These bits are cleared upon sending a USB reset.

Bit 2 – BK Pipe Bank
This bit selects the number of banks for the pipe.
For control endpoints writing a zero to this bit is required as only Bank0 is used for Setup/In/Out transactions.
This bit is cleared when a USB reset is sent.

BK	Description
0x0	Single-bank endpoint
0x1	Dual-bank endpoint

1. Bank field is ignored when PTYPE is configured as EXTENDED.

Bits 1:0 – PTOKEN[1:0] Pipe Token
These bits contains the pipe token.

PTOKEN[1:0](1)	Description
0x0	SETUP(2)
0x1	IN
0x2	OUT
0x3	Reserved

1. The PTOKEN field is ignored when PTYPE is configured as EXTENDED.
2. Available only when PTYPE is configured as CONTROL

Theses bits are cleared upon sending a USB reset.

38.10.17 Interval for the Bulk-Out/Ping Transaction

Name: BINTERVALn
Offset: 0x0103 + n*0x20 [n=0..7]
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
	BINTERVAL[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 – BINTERVAL[7:0] BINTERVAL
These bits contains the Ping/Bulk-out period.
These bits are cleared when a USB reset is sent or when PEN[n] is zero.

BINTERVAL	Description
=0	Multiple consecutive OUT token is sent in the same frame until it is ACKed by the peripheral
>0	One OUT token is sent every BINTERVAL frame until it is ACKed by the peripheral

Depending from the type of pipe the desired period is defined as:

PTYPE	Description
Interrupt	1 ms to 255 ms
Isochronous	2^(BINTERVAL) * 1 ms
Bulk or control	1 ms to 255 ms
EXT LPM	BINTERVAL ignored. Always 1 ms when a NYET is received.

38.10.18 Pipe Status Clear n

Name: PSTATUSCLRn
Offset: 0x0104 + n*0x20 [n=0..7]
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
	BK1RDY	BK0RDY		PFREEZE		CURBK		DTGL
Access	W	W		W		W		W
Reset	0	0		0		0		0

Bit 7 – BK1RDY Bank 1 Ready Clear
Writing a zero to this bit has no effect.
Writing a one to this bit will clear PSTATUS.BK1RDY bit.

Bit 6 – BK0RDY Bank 0 Ready Clear
Writing a zero to this bit has no effect.
Writing a one to this bit will clear PSTATUS.BK0RDY bit.

Bit 4 – PFREEZE Pipe Freeze Clear
Writing a zero to this bit has no effect.
Writing a one to this bit will clear PSTATUS.PFREEZE bit.

Bit 2 – CURBK Current Bank Clear
Writing a zero to this bit has no effect.
Writing a one to this bit will clear PSTATUS.CURBK bit.

Bit 0 – DTGL Data Toggle Clear
Writing a zero to this bit has no effect.
Writing a one to this bit will clear PSTATUS.DTGL bit.

38.10.19 Pipe Status Set Register n

Name: PSTATUSSETn
Offset: 0x0105 + n*0x20 [n=0..7]
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
	BK1RDY	BK0RDY		PFREEZE		CURBK		DTGL
Access	W	W		W		W		W
Reset	0	0		0		0		0

Bit 7 – BK1RDY Bank 1 Ready Set
Writing a zero to this bit has no effect.
Writing a one to this bit will set the bit PSTATUS.BK1RDY.

Bit 6 – BK0RDY Bank 0 Ready Set
Writing a zero to this bit has no effect.
Writing a one to this bit will set the bit PSTATUS.BK0RDY.

Bit 4 – PFREEZE Pipe Freeze Set
Writing a zero to this bit has no effect.
Writing a one to this bit will set PSTATUS.PFREEZE bit.

Bit 2 – CURBK Current Bank Set
Writing a zero to this bit has no effect.
Writing a one to this bit will set PSTATUS.CURBK bit.

Bit 0 – DTGL Data Toggle Set
Writing a zero to this bit has no effect.
Writing a one to this bit will set PSTATUS.DTGL bit.

38.10.20 Pipe Status Register n

Name: PSTATUSn
Offset: 0x0106 + n*0x20 [n=0..7]
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
	BK1RDY	BK0RDY		PFREEZE		CURBK		DTGL
Access	R	R		R		R		R
Reset	0	0		0		0		0

Bit 7 – BK1RDY Bank 1 is ready
Writing a one to the bit EPSTATUSCLR.BK1RDY will clear this bit.
Writing a one to the bit EPSTATUSSET.BK1RDY will set this bit.
This bank is not used for Control pipe.

Value	Description
0	The bank number 1 is not ready: For IN the bank is empty. For Control/OUT the bank is not yet fill in.
1	The bank number 1 is ready: For IN the bank is filled full. For Control/OUT the bank is filled in.

Bit 6 – BK0RDY Bank 0 is ready
Writing a one to the bit EPSTATUSCLR.BK0RDY will clear this bit.
Writing a one to the bit EPSTATUSSET.BK0RDY will set this bit.
This bank is the only one used for Control pipe.

Value	Description
0	The bank number 0 is not ready: For IN the bank is not empty. For Control/OUT the bank is not yet fill in.
1	The bank number 0 is ready: For IN the bank is filled full. For Control/OUT the bank is filled in.

Bit 4 – PFREEZE Pipe Freeze
Writing a one to the bit EPSTATUSCLR.PFREEZE will clear this bit.
Writing a one to the bit EPSTATUSSET.PFREEZE will set this bit.
This bit is also set by the hardware:

- When a STALL handshake has been received.
- After a PIPE has been enabled (rising of bit PEN.N).
- When an LPM transaction has completed whatever handshake is returned or the transaction was timed-out.
- When a pipe transfer was completed with a pipe error. See 38.10.21. PINTFLAGn register.

When PFREEZE bit is set while a transaction is in progress on the USB bus, this transaction will be properly completed. PFREEZE bit will be read as “1” only when the ongoing transaction will have been completed.

Value	Description
0	The Pipe operates in normal operation.
1	The Pipe is frozen and no additional requests will be sent to the device on this pipe address.

Bit 2 – CURBK Current Bank

Value	Description
0	The bank0 is the bank that will be used in the next single/multi USB packet.
1	The bank1 is the bank that will be used in the next single/multi USB packet.

Bit 0 – DTGL Data Toggle Sequence
Writing a one to the bit EPSTATUSCLR.DTGL will clear this bit.
Writing a one to the bit EPSTATUSSET.DTGL will set this bit.
This bit is toggled automatically by hardware after a data transaction.
This bit will reflect the data toggle in regards of the token type (IN/OUT/SETUP).

Value	Description
0	The PID of the next expected transaction will be zero: data 0.

Value	Description
1	The PID of the next expected transaction will be one: data 1.

38.10.21 Host Pipe Interrupt Flag Register

Name: PINTFLAGn
Offset: 0x0107 + n*0x20 [n=0..7]
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
			STALL	TXSTP	PERR	TRFAIL	TRCPT1	TRCPT0
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit 5 – STALL STALL Received Interrupt Flag
This flag is cleared by writing a one to the flag.
This flag is set when a stall occurs and will generate an interrupt if PINTENCLR/SET.STALL is one.
Writing a zero to this bit has no effect.
Writing a one to this bit clears the STALL Interrupt Flag.

Bit 4 – TXSTP Transmitted Setup Interrupt Flag
This flag is cleared by writing a one to the flag.
This flag is set when a Setup Packet Transaction Complete occurs and will generate an interrupt if PINTENCLR/SET.TXSTP is one.
Writing a zero to this bit has no effect.
Writing a one to this bit clears the TXSTP Interrupt Flag.

Bit 3 – PERR Pipe Error Interrupt Flag
This flag is cleared by writing a one to the flag.
This flag is set when a pipe error occurs and will generate an interrupt if PINTENCLR/SET.PERR is one.
Writing a zero to this bit has no effect.
Writing a one to this bit clears the PERR Interrupt Flag.

Bit 2 – TRFAIL Transfer Fail Interrupt Flag
This flag is cleared by writing a one to the flag.
This flag is set when a Transfer Fail occurs and will generate an interrupt if PINTENCLR/SET.TRFAIL is one.
Writing a zero to this bit has no effect.
Writing a one to this bit clears the TRFAIL Interrupt Flag.

Bits 0, 1 – TRCPT Transfer Complete x interrupt Flag
This flag is cleared by writing a one to the flag.
This flag is set when a Transfer complete occurs and will generate an interrupt if PINTENCLR/SET.TRCPT is one.
PINTFLAG.TRCPT is set for a single bank IN/OUT pipe or a double bank IN/OUT pipe when current bank is 0.
Writing a zero to this bit has no effect.
Writing a one to this bit clears the TRCPT Interrupt Flag.

38.10.22 Host Pipe Interrupt Clear Register

Name: PINTENCLRn
Offset: 0x0108 + n*0x20 [n=0..7]
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 Pipe Interrupt Enable Set (PINTENSET) register.

This register is cleared by USB reset or when PEN[n] is zero.

Bit	7	6	5	4	3	2	1	0
			STALL	TXSTP	PERR	TRFAIL	TRCPT1	TRCPT0
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit 5 – STALL Received Stall Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will clear the Received Stall interrupt Enable bit and disable the corresponding interrupt request.

Value	Description
0	The received Stall interrupt is disabled.
1	The received Stall interrupt is enabled and an interrupt request will be generated when the received Stall interrupt Flag is set.

Bit 4 – TXSTP Transmitted Setup Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will clear the Transmitted Setup interrupt Enable bit and disable the corresponding interrupt request.

Value	Description
0	The Transmitted Setup interrupt is disabled.
1	The Transmitted Setup interrupt is enabled and an interrupt request will be generated when the Transmitted Setup interrupt Flag is set.

Bit 3 – PERR Pipe Error Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will clear the Pipe Error interrupt Enable bit and disable the corresponding interrupt request.

Value	Description
0	The Pipe Error interrupt is disabled.
1	The Pipe Error interrupt is enabled and an interrupt request will be generated when the Pipe Error interrupt Flag is set.

Bit 2 – TRFAIL Transfer Fail Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will clear the Transfer Fail interrupt Enable bit and disable the corresponding interrupt request.

Value	Description
0	The Transfer Fail interrupt is disabled.
1	The Transfer Fail interrupt is enabled and an interrupt request will be generated when the Transfer Fail interrupt Flag is set.

Bits 0, 1 – TRCPT Transfer Complete Bank x interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will clear the Transfer Complete interrupt Enable bit x and disable the corresponding interrupt request.

Value	Description
0	The Transfer Complete Bank x interrupt is disabled.

Value	Description
1	The Transfer Complete Bank x interrupt is enabled and an interrupt request will be generated when the Transfer Complete interrupt x Flag is set.

38.10.23 Host Interrupt Pipe Set Register

Name: PINTENSETn
Offset: 0x0109 + n*0x20 [n=0..7]
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 Pipe Interrupt Enable Set (PINTENCLR) register.

This register is cleared by USB reset or when PEN[n] is zero.

Bit	7	6	5	4	3	2	1	0
			STALL	TXSTP	PERR	TRFAIL	TRCPT1	TRCPT0
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit 5 – STALL Stall Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will enable the Stall interrupt.

Value	Description
0	The Stall interrupt is disabled.
1	The Stall interrupt is enabled.

Bit 4 – TXSTP Transmitted Setup Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will enable the Transmitted Setup interrupt.

Value	Description
0	The Transmitted Setup interrupt is disabled.
1	The Transmitted Setup interrupt is enabled.

Bit 3 – PERR Pipe Error Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will enable the Pipe Error interrupt.

Value	Description
0	The Pipe Error interrupt is disabled.
1	The Pipe Error interrupt is enabled.

Bit 2 – TRFAIL Transfer Fail Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will enable the Transfer Fail interrupt.

Value	Description
0	The Transfer Fail interrupt is disabled.
1	The Transfer Fail interrupt is enabled.

Bits 0, 1 – TRCPT Transfer Complete x interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will enable the Transfer Complete interrupt Enable bit x.
0.2.7 Host Registers - Pipe RAM

Value	Description
0	The Transfer Complete x interrupt is disabled.
1	The Transfer Complete x interrupt is enabled.

38.11 Register Summary - USB Host Pipe n Descriptor Bank 0

This Register Description section is valid if the USB is in Host mode (CTRLA.MODE=1).

Refer to the Registers Description section for more details on register properties and access permissions.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	ADDR	31:24	ADDR[31:24]							
		23:16	ADDR[23:16]							
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0x04	PCKSIZE	31:24	AUTO_ZLP	SIZE[2:0]			MULTI_PACKET_SIZE[13:10]			
		23:16	MULTI_PACKET_SIZE[9:2]							
		15:8	MULTI_PACKET_SIZE[1:0]		BYTE_COUNT[5:0]					
		7:0								
0x08	EXTREG	15:8		VARIABLE[10:4]						
		7:0	VARIABLE[3:0]				SUBPID[3:0]			
0x0A	STATUS_BK	7:0							ERRORFLOW	CRCERR
0x0B	Reserved									
0x0C	CTRL_PIPE	15:8	PERMAX[3:0]				PEPNUM[3:0]			
		7:0		PDADDR[6:0]						
0x0E	STATUS_PIPE	15:8								
		7:0	ERCNT[2:0]			CRC16ER	TOUTER	PIDER	DAPIDER	DTGLER

38.11.1 Address of the Data Buffer

Name: ADDR
Offset: 0x00
Reset: -
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	x

Bits 31:0 – ADDR[31:0] Data Pointer Address Value
These bits define the data pointer address as an absolute double word address in RAM. The two least significant bits must be zero to ensure the descriptor is 32-bit aligned.

38.11.2 Packet Size

Name: PCKSIZE
Offset: 0x04
Reset: -
Property: -

Bit	31	30	29	28	27	26	25	24
	AUTO_ZLP	SIZE[2:0]			MULTI_PACKET_SIZE[13:10]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	x	0	0	x	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	MULTI_PACKET_SIZE[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	15	14	13	12	11	10	9	8
	MULTI_PACKET_SIZE[1:0]		BYTE_COUNT[5:0]					
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	x	0	0	0	0	0	x

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bit 31 – AUTO_ZLP Automatic Zero Length Packet
This bit defines the automatic Zero Length Packet mode of the pipe.
When enabled, the USB module will manage the ZLP handshake by hardware. This bit is for OUT pipes only. When disabled the handshake should be managed by firmware.

Value	Description
0	Automatic Zero Length Packet is disabled.
1	Automatic Zero Length Packet is enabled.

Bits 30:28 – SIZE[2:0] Pipe size
These bits contains the size of the pipe.
Theses bits are cleared upon sending a USB reset.

SIZE[2:0]	Description
0x0	8 Byte
0x1	16 Byte
0x2	32 Byte
0x3	64 Byte
0x4	128 Byte(1)
0x5	256 Byte(1)
0x6	512 Byte(1)
0x7	1024 Byte in HS mode(1) 1023 Byte in FS mode(1)

1. For Isochronous pipe only.

Bits 27:14 – MULTI_PACKET_SIZE[13:0] Multi Packet IN or OUT size
These bits define the 14-bit value that is used for multi-packet transfers.
For IN pipes, MULTI_PACKET_SIZE holds the total number of bytes sent. MULTI_PACKET_SIZE should be written to zero when setting up a new transfer.

For OUT pipes, MULTI_PACKET_SIZE holds the total data size for the complete transfer. This value must be a multiple of the maximum packet size.

Bits 13:8 – BYTE_COUNT[5:0] Byte Count
These bits define the 14-bit value that contains number of bytes sent in the last OUT or SETUP transaction for an OUT pipe, or of the number of bytes to be received in the next IN transaction for an input pipe.

38.11.3 Extended Register

Name: EXTREG
Offset: 0x08
Reset: -
Property: -

Bit	15	14	13	12	11	10	9	8
		VARIABLE[10:4]						
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
	VARIABLE[3:0]				SUBPID[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	x	0	0	0	x

Bits 14:4 – VARIABLE[10:0] Variable field send with extended token
These bits define the VARIABLE field sent with extended token. See “Section 2.1.1 Protocol Extension Token in the reference document ENGINEERING CHANGE NOTICE, USB 2.0 Link Power Management Addendum.”
To support the USB2.0 Link Power Management addition the VARIABLE field should be set as described below.

VARIABLE	Description
VARIABLE[3:0]	bLinkState(1)
VARIABLE[7:4]	BESL (See LPM ECN)(2)
VARIABLE[8]	bRemoteWake(1)
VARIABLE[10:9]	Reserved

(1) for a definition of LPM Token bRemoteWake and bLinkState fields, refer to "Table 2-3 in the reference document ENGINEERING CHANGE NOTICE, USB 2.0 Link Power Management Addendum"
(2) for a definition of LPM Token BESL field, refer to "Table 2-3 in the reference document ENGINEERING CHANGE NOTICE, USB 2.0 Link Power Management Addendum" and "Table X-X1 in Errata for ECN USB 2.0 Link Power Management.

Bits 3:0 – SUBPID[3:0] SUBPID field send with extended token
These bits define the SUBPID field sent with extended token. See “Section 2.1.1 Protocol Extension Token in the reference document ENGINEERING CHANGE NOTICE, USB 2.0 Link Power Management Addendum”.
To support the USB2.0 Link Power Management addition the SUBPID field should be set as described in “Table 2.2 SubPID Types in the reference document ENGINEERING CHANGE NOTICE, USB 2.0 Link Power Management Addendum”.

38.11.4 Host Status Bank

Name: STATUS_BK
Offset: 0x0A
Reset: -
Property: -

Bit	7	6	5	4	3	2	1	0
							ERRORFLOW	CRCERR
Access							R/W	R/W
Reset							x	x

Bit 1 – ERRORFLOW Error Flow Status

This bit defines the Error Flow Status.
This bit is set when a Error Flow has been detected during transfer from/towards this bank.
For IN transfer, a NAK handshake has been received. For OUT transfer, a NAK handshake has been received. For Isochronous IN transfer, an overrun condition has occurred. For Isochronous OUT transfer, an underflow condition has occurred.

Value	Description
0	No Error Flow detected.
1	A Error Flow has been detected.

Bit 0 – CRCERR CRC Error

This bit defines the CRC Error Status.
This bit is set when a CRC error has been detected in an isochronous IN endpoint bank.

Value	Description
0	No CRC Error.
1	CRC Error detected.

38.11.5 Host Control Pipe

Name: CTRL_PIPE
Offset: 0x0C
Reset: -
Property: -

Bit	15	14	13	12	11	10	9	8
	PERMAX[3:0]				PEPNUM[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	x	0	0	0	x

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

Bits 15:12 – PERMAX[3:0] Pipe Error Max Number
These bits define the maximum number of error for this Pipe before freezing the pipe automatically.

Bits 11:8 – PEPNUM[3:0] Pipe EndPoint Number
These bits define the number of endpoint for this Pipe.

Bits 6:0 – PDADDR[6:0] Pipe Device Address
These bits define the Device Address for this pipe.

38.11.6 Host Status Pipe

Name: STATUS_PIPE
Offset: 0x0E
Reset: -
Property: -

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	ERCNT[2:0]			CRC16ER	TOUTER	PIDER	DAPIDER	DTGLER
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	x	x	x	x	x	x

Bits 7:5 – ERCNT[2:0] Pipe Error Counter
These bits define the number of errors detected on the pipe.

Bit 4 – CRC16ER CRC16 ERROR
This bit defines the CRC16 Error Status.
This bit is set when a CRC 16 error has been detected during a IN transactions.

Value	Description
0	No CRC 16 Error detected.
1	A CRC 16 error has been detected.

Bit 3 – TOUTER TIME OUT ERROR
This bit defines the Time Out Error Status.
This bit is set when a Time Out error has been detected during a USB transaction.

Value	Description
0	No Time Out Error detected.
1	A Time Out error has been detected.

Bit 2 – PIDER PID ERROR
This bit defines the PID Error Status.
This bit is set when a PID error has been detected during a USB transaction.

Value	Description
0	No PID Error detected.
1	A PID error has been detected.

Bit 1 – DAPIDER Data PID ERROR
This bit defines the PID Error Status.
This bit is set when a Data PID error has been detected during a USB transaction.

Value	Description
0	No Data PID Error detected.
1	A Data PID error has been detected.

Bit 0 – DTGLER Data Toggle Error
This bit defines the Data Toggle Error Status.
This bit is set when a Data Toggle Error has been detected.

Value	Description
0	No Data Toggle Error.
1	Data Toggle Error detected.

38.12 Register Summary - USB Host Pipe n Descriptor Bank 1

This Register Description section is valid if the USB is in Host mode (CTRLA.MODE=1).

Refer to the Registers Description section for more details on register properties and access permissions.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00 ... 0x0F	Reserved									
0x10	ADDR	31:24	ADDR[31:24]							
		23:16	ADDR[23:16]							
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0x14	PCKSIZE	31:24	AUTO_ZLP	SIZE[2:0]			MULTI_PACKET_SIZE[13:10]			
		23:16	MULTI_PACKET_SIZE[9:2]							
		15:8	MULTI_PACKET_SIZE[1:0]		BYTE_COUNT[5:0]					
		7:0								
0x18 ... 0x19	Reserved									
0x1A	STATUS_BK	7:0							ERRORFLOW	CRCERR
0x1B ... 0x1D	Reserved									
0x1E	STATUS_PIPE	15:8								
		7:0	ERCNT[2:0]			CRC16ER	TOUTER	PIDER	DAPIDER	DTGLER

38.12.1 Address of the Data Buffer

Name: ADDR
Offset: 0x10
Reset: -
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	x

Bits 31:0 – ADDR[31:0] Data Pointer Address Value
These bits define the data pointer address as an absolute double word address in RAM. The two least significant bits must be zero to ensure the descriptor is 32-bit aligned.

38.12.2 Packet Size

Name: PCKSIZE
Offset: 0x14
Reset: -
Property: -

Bit	31	30	29	28	27	26	25	24
	AUTO_ZLP	SIZE[2:0]			MULTI_PACKET_SIZE[13:10]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	x	0	0	x	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	MULTI_PACKET_SIZE[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	15	14	13	12	11	10	9	8
	MULTI_PACKET_SIZE[1:0]		BYTE_COUNT[5:0]					
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	x	0	0	0	0	0	x

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bit 31 – AUTO_ZLP Automatic Zero Length Packet
This bit defines the automatic Zero Length Packet mode of the pipe.
When enabled, the USB module will manage the ZLP handshake by hardware. This bit is for OUT pipes only. When disabled the handshake should be managed by firmware.

Value	Description
0	Automatic Zero Length Packet is disabled.
1	Automatic Zero Length Packet is enabled.

Bits 30:28 – SIZE[2:0] Pipe size
These bits contains the size of the pipe.
Theses bits are cleared upon sending a USB reset.

SIZE[2:0]	Description
0x0	8 Byte
0x1	16 Byte
0x2	32 Byte
0x3	64 Byte
0x4	128 Byte(1)
0x5	256 Byte(1)
0x6	512 Byte(1)
0x7	1024 Byte in HS mode(1) 1023 Byte in FS mode(1)

1. For Isochronous pipe only.

Bits 27:14 – MULTI_PACKET_SIZE[13:0] Multi Packet IN or OUT size
These bits define the 14-bit value that is used for multi-packet transfers.
For IN pipes, MULTI_PACKET_SIZE holds the total number of bytes sent. MULTI_PACKET_SIZE should be written to zero when setting up a new transfer.

For OUT pipes, MULTI_PACKET_SIZE holds the total data size for the complete transfer. This value must be a multiple of the maximum packet size.

Bits 13:8 – BYTE_COUNT[5:0] Byte Count
These bits define the 14-bit value that contains number of bytes sent in the last OUT or SETUP transaction for an OUT pipe, or of the number of bytes to be received in the next IN transaction for an input pipe.

38.12.3 Host Status Bank

Name: STATUS_BK
Offset: 0x1A
Reset: -
Property: -

Bit	7	6	5	4	3	2	1	0
							ERRORFLOW	CRCERR
Access							R/W	R/W
Reset							x	x

Bit 1 – ERRORFLOW Error Flow Status

This bit defines the Error Flow Status.
This bit is set when a Error Flow has been detected during transfer from/towards this bank.
For IN transfer, a NAK handshake has been received. For OUT transfer, a NAK handshake has been received. For Isochronous IN transfer, an overrun condition has occurred. For Isochronous OUT transfer, an underflow condition has occurred.

Value	Description
0	No Error Flow detected.
1	A Error Flow has been detected.

Bit 0 – CRCERR CRC Error

This bit defines the CRC Error Status.
This bit is set when a CRC error has been detected in an isochronous IN endpoint bank.

Value	Description
0	No CRC Error.
1	CRC Error detected.

38.12.4 Host Status Pipe

Name: STATUS_PIPE
Offset: 0x1E
Reset: -
Property: -

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	ERCNT[2:0]			CRC16ER	TOUTER	PIDER	DAPIDER	DTGLER
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	x	x	x	x	x	x

Bits 7:5 – ERCNT[2:0] Pipe Error Counter
These bits define the number of errors detected on the pipe.

Bit 4 – CRC16ER CRC16 ERROR
This bit defines the CRC16 Error Status.
This bit is set when a CRC 16 error has been detected during a IN transactions.

Value	Description
0	No CRC 16 Error detected.
1	A CRC 16 error has been detected.

Bit 3 – TOUTER TIME OUT ERROR
This bit defines the Time Out Error Status.
This bit is set when a Time Out error has been detected during a USB transaction.

Value	Description
0	No Time Out Error detected.
1	A Time Out error has been detected.

Bit 2 – PIDER PID ERROR
This bit defines the PID Error Status.
This bit is set when a PID error has been detected during a USB transaction.

Value	Description
0	No PID Error detected.
1	A PID error has been detected.

Bit 1 – DAPIDER Data PID ERROR
This bit defines the PID Error Status.
This bit is set when a Data PID error has been detected during a USB transaction.

Value	Description
0	No Data PID Error detected.
1	A Data PID error has been detected.

Bit 0 – DTGLER Data Toggle Error
This bit defines the Data Toggle Error Status.
This bit is set when a Data Toggle Error has been detected.

Value	Description
0	No Data Toggle Error.
1	Data Toggle Error detected.

39. Timer/Counter (TC)

39.1 Overview

There are up to 3 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 timestamp input events or IO pin edges, allowing for capturing of frequency and/or pulse width.

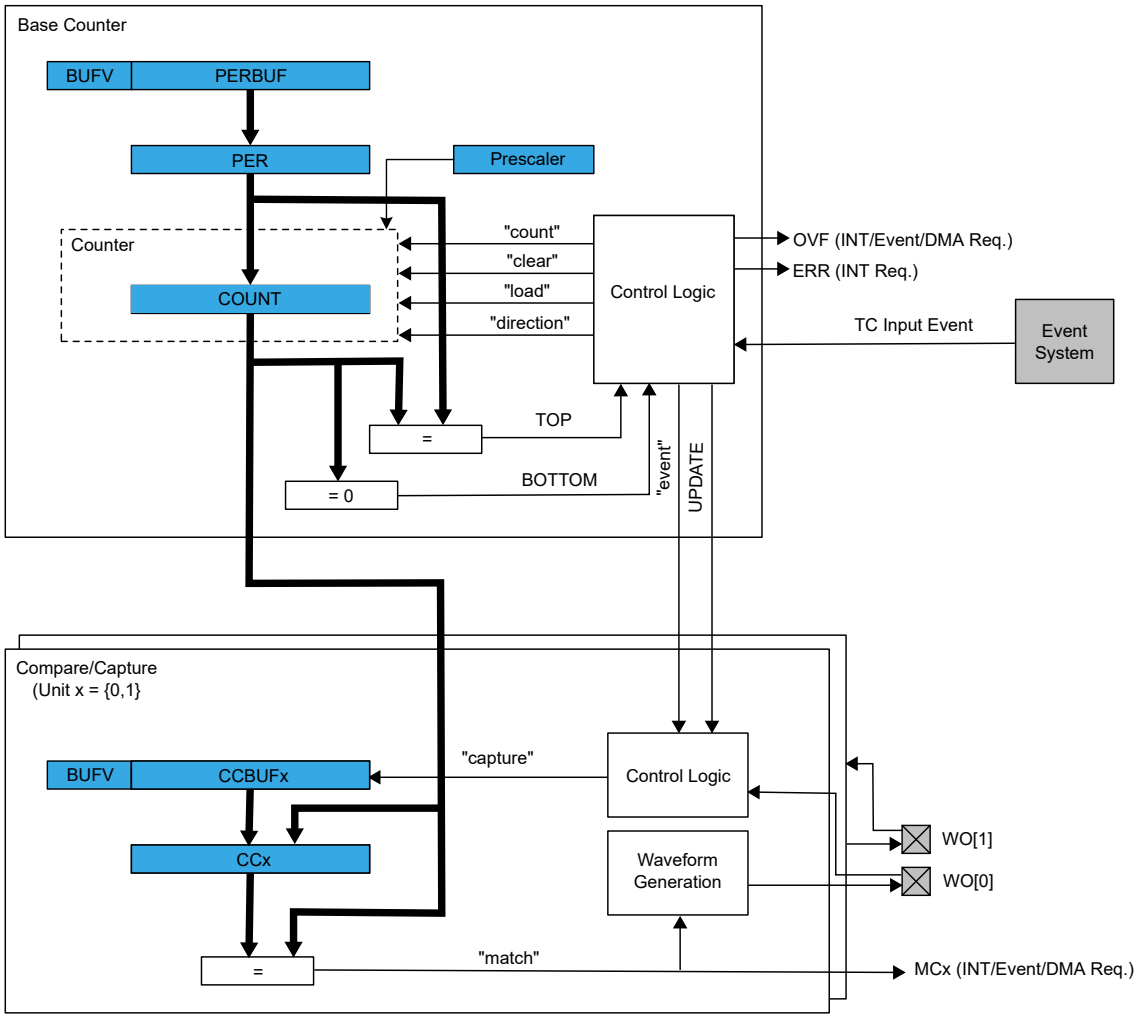
A TC can also perform waveform generation, such as frequency generation and pulse-width modulation.

39.2 Features

- Selectable configuration
  - 8-bit, 16-bit or 32-bit TC operation, with compare/capture channels
- 2 compare/capture channels (CC) with:
  - Double buffered timer period setting
  - 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

39.3 Block Diagram

Figure 39-1. Timer/Counter Block Diagram



39.4 Signal Description

Table 39-1. Signal Description for TC.

Signal Name	Type	Description
WO[1:0]	Digital output	Waveform output
	Digital input	Capture input

Refer to the Pinout for details on the pin mapping for this peripheral. One signal can be mapped on several pins.

39.5 Peripheral Dependencies

Table 39-2. TC Configuration Summary

Peripheral name	Base address	NVIC IRQ Index	MCLK AHB/APB Clocks	GCLK Peripheral Channel Index (GCLK.PCHCTRL)	PAC Peripheral Identifier Index (PAC.WRCTRL)	DMA Trigger Source Index (DMAC.CHCTRLB)	Events (EVSYS)		Power Domain (PM.STDBYCFG)
							Users (EVSYS.USER)	Generators (EVSYS.CHANNEL)	
TC0	0x42001C00	55: ERR, MC0, MC1, OVF	CLK_TC0_APB	23: GCLK_TC0_TC1	71	16: OVF 17-18: MC0-1	15: EVU	41: OVF 42-43: MC0-1	PDSW
TC1	0x42002000	56: ERR, MC0, MC1, OVF	CLK_TC1_APB	23: GCLK_TC0_TC1	72	19: OVF 20-21: MC0-1	16: EVU	44: OVF 45-46: MC0-1	PDSW
TC2	0x42002400	57: ERR, MC0, MC1, OVF	CLK_TC2_APB	24: GCLK_TC2	73	22: OVF 23-24: MC0-1	17: EVU	47: OVF 48-49: MC0-1	PDSW

39.6 Functional Description

39.6.1 Principle of Operation

The following definitions are used throughout the documentation:

Table 39-3. 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 39.6.2.6.1. Waveform Output Operations.
ZERO	The counter is ZERO when it contains all zeroes
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	TC mode where increment/decrement/clear/reload steps are done on each prescaled clock
Counter	TC mode where increment/decrement/clear/reload steps are done on each detected 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.

The CCx registers are using buffer registers (CCBUFx) for optimized timing. 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 according MAX values. Mode settings (CTRLA.MODE) determine the maximum range of the Counter register.

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 compare operation, the counter value is continuously compared to the values in the CCx registers. In 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.

39.6.2 Basic Operation

39.6.2.1 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 (CLK_TCx_APB).
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 IO pin input signal for individual channels via the Invert Enable bit group in the Drive Control register (DRVCTRL.INVEN).

Note: Two instances of the TC may share a peripheral clock channel. In this case, they cannot be set to different clock frequencies. Refer to the peripheral clock channel mapping of the Generic Clock Controller (GCLK.PCHTRLm) to identify shared peripheral clocks.

39.6.2.2 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. Refer to the 39.7.1. CTRLA register for details.

The TC should be disabled before the TC is reset in order to avoid undefined behavior.

39.6.2.3 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.

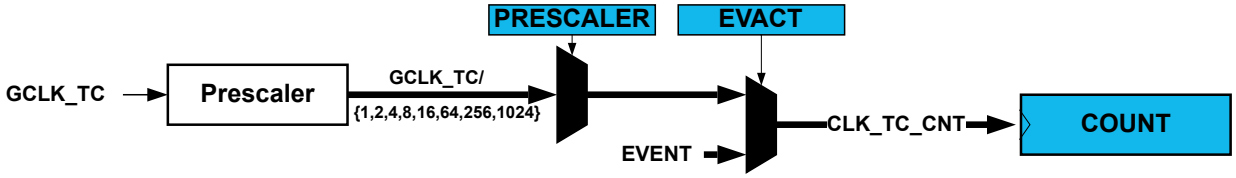
If the prescaler value is higher than one, the counter update condition can be optionally executed on the next GCLK_TCx clock pulse or the next prescaled clock pulse. For further details, refer to Prescaler (CTRLA.PRESCALER) and Counter Synchronization (CTRLA.PRESYNC) description.

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_TC_CNT.

Figure 39-2. Prescaler



39.6.2.4 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: 8-bit counter mode
- COUNT16: 16-bit counter mode
- COUNT32: This mode is achieved by pairing two 16-bit TC peripherals
  When paired, the TC peripherals are configured using the registers of the even-numbered TC. The odd-numbered partner will act as a client, and the Client bit in the Status register (STATUS.SLAVE) will be set. The register values of a client 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.

Table 39-4. 32-bit Resolution Mode Support

Timer Counter	32-bit Resolution Mode
TC0	Yes: Host
TC1	Yes: Client to TC0
TC2	No

39.6.2.5 Counter Operations

Depending on the mode of operation, the counter is cleared, reloaded, incremented, or decremented at each TC clock input (CLK_TC_CNT). A counter clear or reload marks the end of the current counter cycle and the start of a new one.

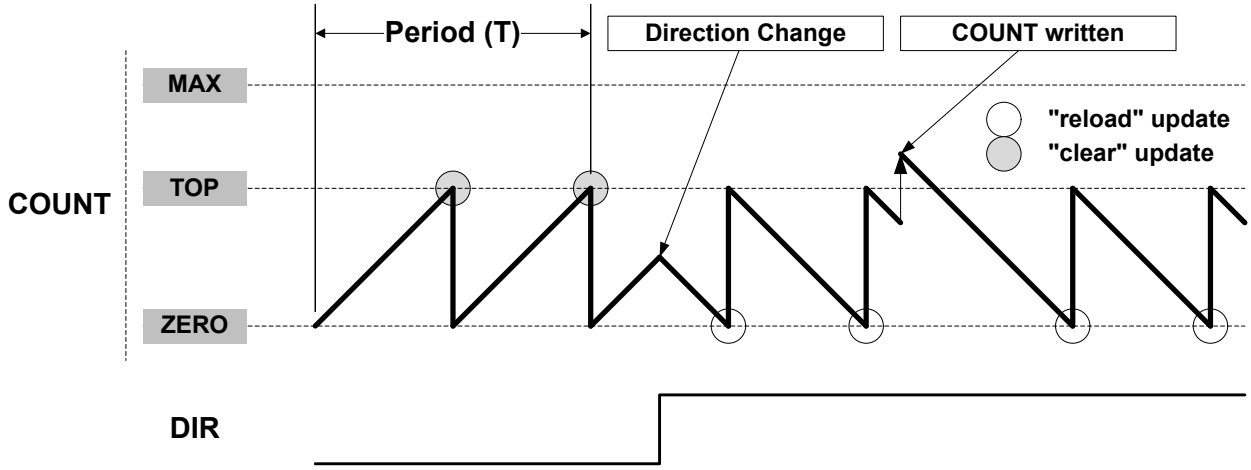
The counting direction is set by the Direction bit in the Control B register (CTRLB.DIR). If this bit is zero the counter is counting up, and counting down if CTRLB.DIR=1. The counter will count up or down for each tick (clock or event) until it reaches TOP or ZERO. When it is counting up and TOP is 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 39-3. 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.

39.6.2.5.1 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 will be loaded with the starting value (ZERO or TOP, depending on direction set by CTRLBSET.DIR or CTRLBCLR.DIR). All waveforms are cleared and the Stop bit in the Status register is set (STATUS.STOP).

39.6.2.5.2 Retrigger Command and Event Action

A retrigger command can be issued from software by writing the Command bits in the Control B Set register (CTRLBSET.CMD = 0x1, RETRIGGER), or from event when a retrigger event action is configured in the Event Control register (EVCTRL.EVACT = 0x1, RETRIGGER).

When the command is detected during counting operation, the counter will be reloaded or cleared, depending on the counting direction (CTRLBSET.DIR or CTRLBCLR.DIR). When the retrigger command is detected while the counter is stopped, the counter will resume counting from the current value in the COUNT register.

Note: When a retrigger 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 corresponding following event.

39.6.2.5.3 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.

39.6.2.5.4 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 retrigger 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).

39.6.2.6 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). For further details, refer to 39.6.2.7. Double Buffering. The synchronization prevents the occurrence of odd-length, non-symmetrical pulses and ensures glitch-free output.

39.6.2.6.1 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 Pin Controller. Refer to PORT - I/O Pin Controller for details.

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 (see Normal Frequency Operation). An interrupt/and or event can be generated on 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 will influence how the waveform is generated and impose restrictions on the top value. The configurations are:

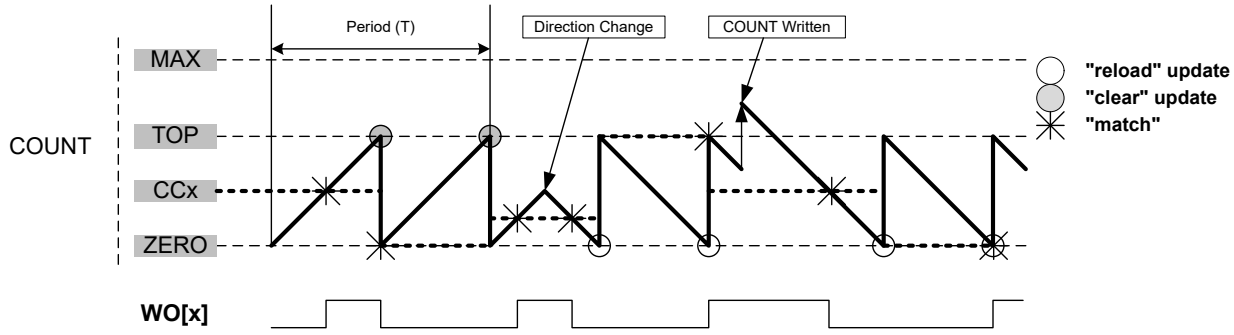
- Normal frequency (NFRQ)
- Match frequency (MFRQ)
- Normal pulse-width modulation (NPWM)
- Match pulse-width modulation (MPWM)

When using NPWM or NFRQ configuration, TOP is determined by the Period register (PER). TOP can be changed by writing to the PER register.

Normal Frequency Generation (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 (INTFLAG.MCx) will be set.

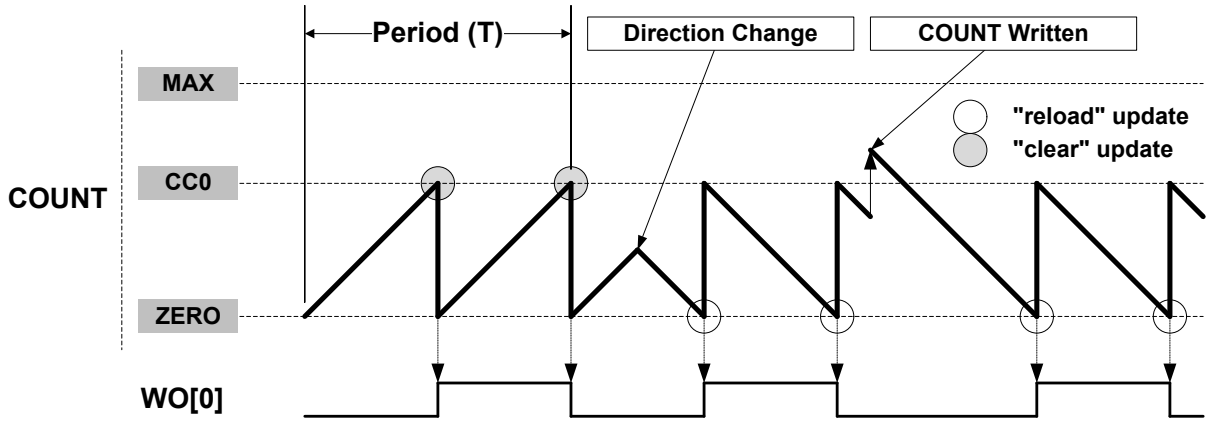
Figure 39-4. Normal Frequency Operation



Match Frequency Generation (MFRQ)

For Match Frequency Generation, the period time (T) is controlled by the CC0 register instead of PER. WO[0] toggles on each update condition.

Figure 39-5. 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 start or compare match between the COUNT and ZERO values, and set on compare match between COUNT and CCx register values.

The following equation calculates the exact resolution for a single-slope PWM (R_{PWM_SS}) waveform:

$$R_{PWM\_SS} = \frac{\log(TOP+1)}{\log(2)}$$

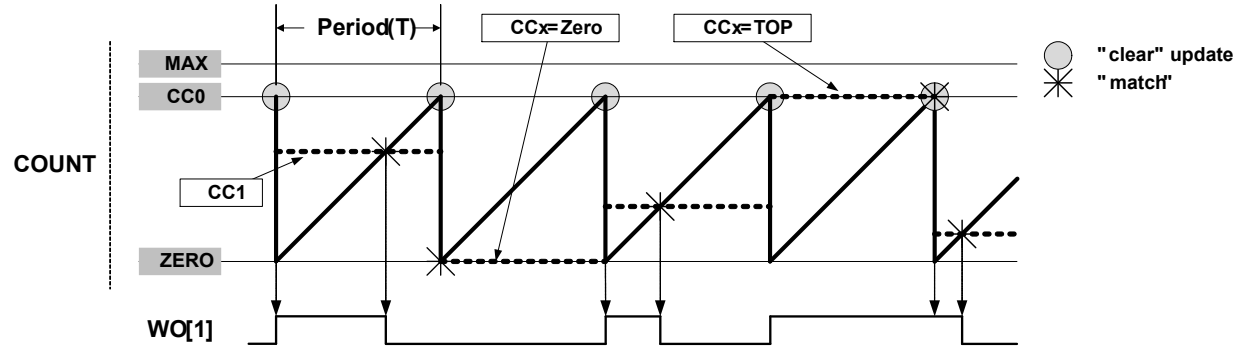
The PWM frequency (f_{PWM_SS}) depends on TOP value and the peripheral clock frequency (f_{GCLK_TC}), and can be calculated by the following equation:

$$f_{PWM\_SS} = \frac{f_{GCLK\_TC}}{N(TOP+1)}$$

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 shown in the figure below. On every overflow/underflow, a one-TC-clock-cycle negative pulse is put out on WO[0] (not shown in the figure).

Figure 39-6. Match PWM Operation



The table below shows the update counter and overflow event/interrupt generation conditions in different operation modes.

Table 39-5. 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	See description above		TOP	ZERO

39.6.2.7 Double Buffering

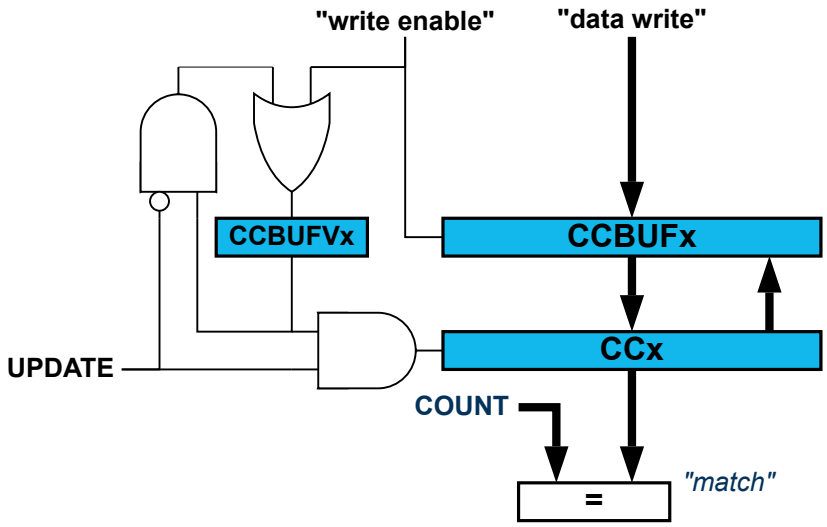
The Compare Channels (CCx) registers, and the Period (PER) register are double buffered. Each buffer register has a buffer valid bit (CCBUFVx or PERBUFV) in the STATUS register, which indicates that the buffer register contains a new valid value that can be copied into the corresponding register. As 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 39-7. Compare Channel Double Buffering

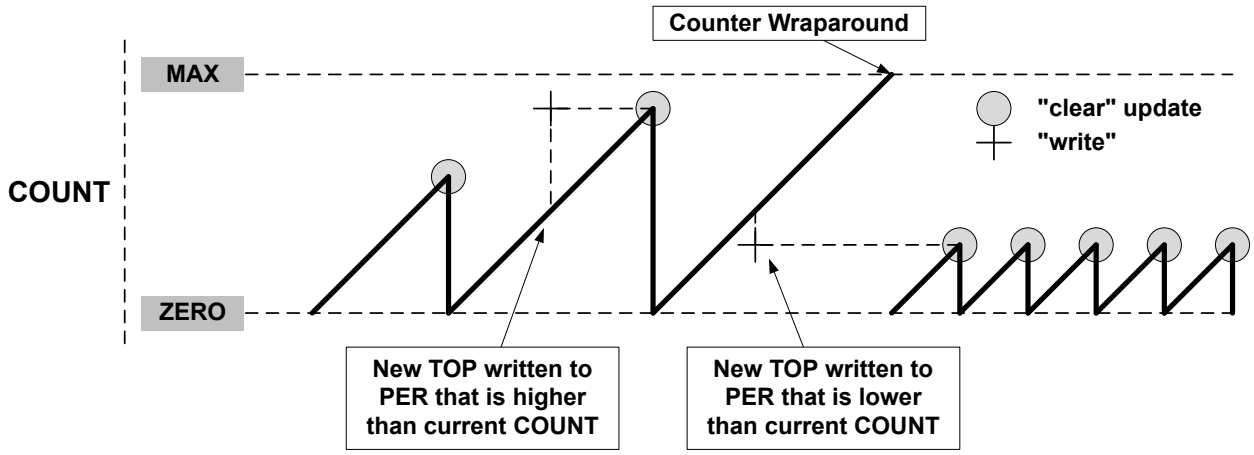


Both the registers (PER/CCx) and corresponding buffer registers (PERBUF/CCBUFx) are available in the I/O register map, and the double buffering feature is not mandatory. The double buffering is disabled by writing a '1' to CTRLBSET.LUPD.

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.

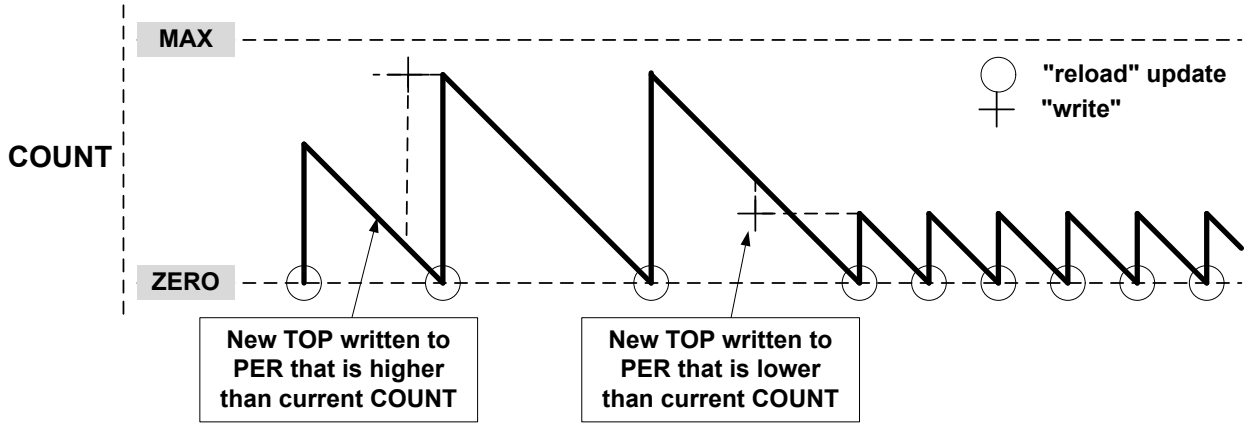
Figure 39-8. Unbuffered Single-Slope Up-Counting Operation



A counter wraparound can occur in any operation mode when up-counting without buffering, see Figure 39-8.

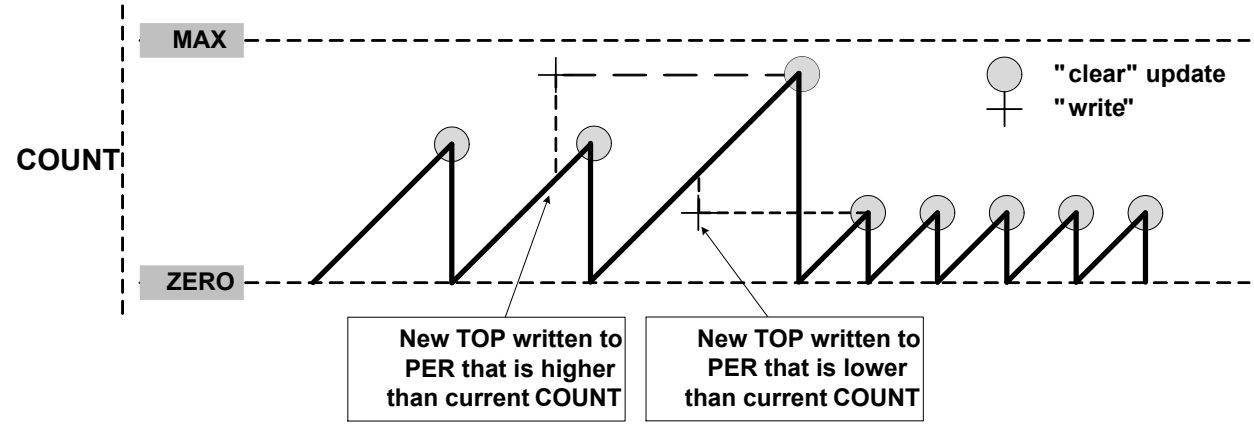
COUNT and TOP are continuously compared, so when a new TOP value that is lower than current COUNT is written to TOP, COUNT will wrap before a compare match.

Figure 39-9. 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 Figure 39-10. This prevents wraparound and the generation of odd waveforms.

Figure 39-10. Changing the Period Using Buffering



39.6.2.8 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 input event line TC_EV or by asynchronous IO pin WO[x] for each capture channel or by a TC event. To enable the capture from input event line, Event Input Enable bit in the Event Control register (EVCTRL.TCEI) must be written to '1'. To enable the capture from the IO pin, the Capture On Pin x Enable bit in CTRLA register (CTRLA.COPENx) must be written to '1'.

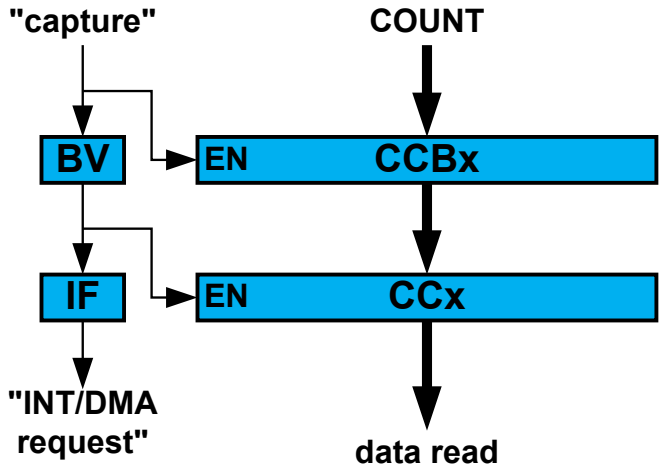
Notes:

1. Capture on I/Os is only possible in 'Event' and 'Time-Stamp' capture action modes. Other modes can only use internal events. (if I/Os toggling is needed in other modes, then the I/Os edge should 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 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/Os 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 falling edge is available, its activation is depending on the input source:

- When the channel is used with a 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 the Event Control register (EVCTRL.TCINV).

Figure 39-11. Capture Double Buffering

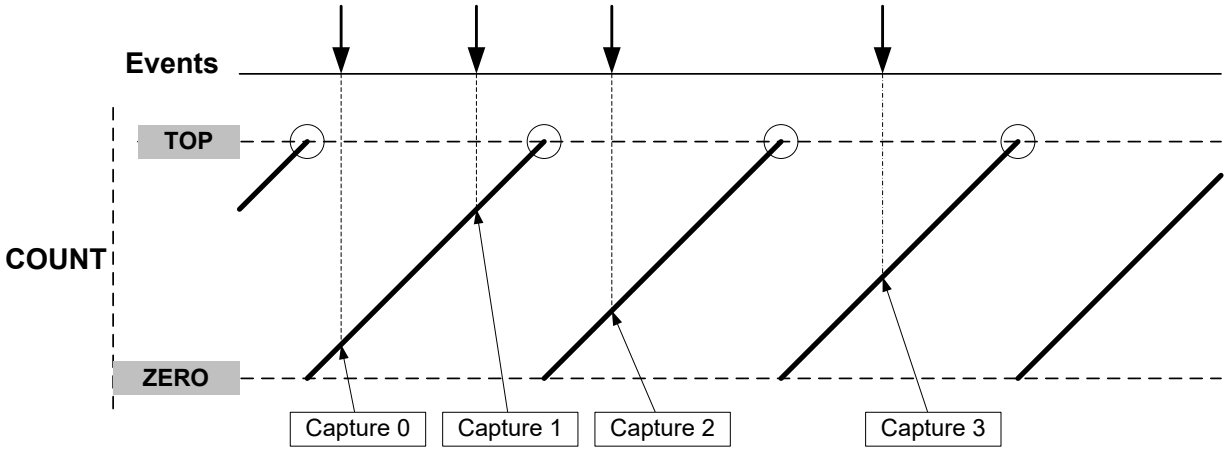


For input capture, the buffer register and the corresponding CCx act like a FIFO. When CCx is empty or read, any content in CCBUFx is transferred to CCx. The buffer valid flag is passed to set the CCx interrupt flag (IF) and generate the optional interrupt, event or DMA request. The CCBUFx register value can't be read, all captured data must be read from CCx register.

39.6.2.8.1 Event Capture Action on Events or I/Os

The compare/capture channels can be used as input capture channels to capture events from the Event System or I/Os and give them a timestamp. This mode is selected when EVTCTRL.EVACT is configured either as OOF, RETRIGGER, COUNT or START. The following figure shows four capture events for one capture channel.

Figure 39-12. 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.

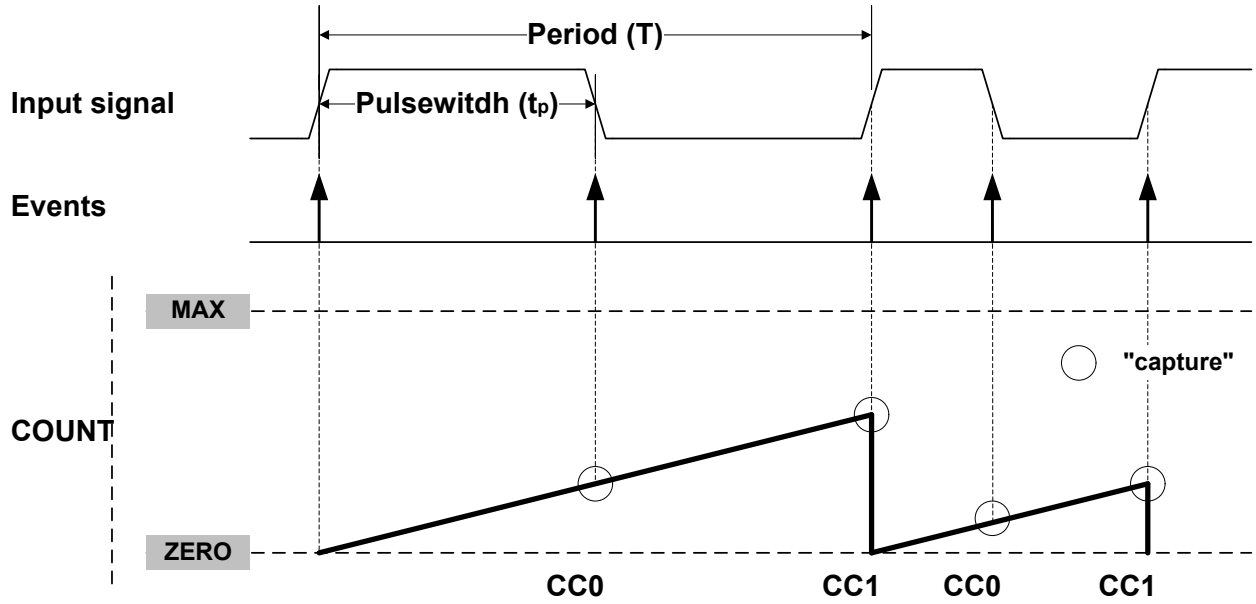
39.6.2.8.2 Period and Pulse-Width (PPW/PWP) Capture Action on Events

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 39-13. 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 the other 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 should occur on the rising edge or the falling edge. If EVCTRL.TCINV=1, the wraparound will happen on the falling edge.

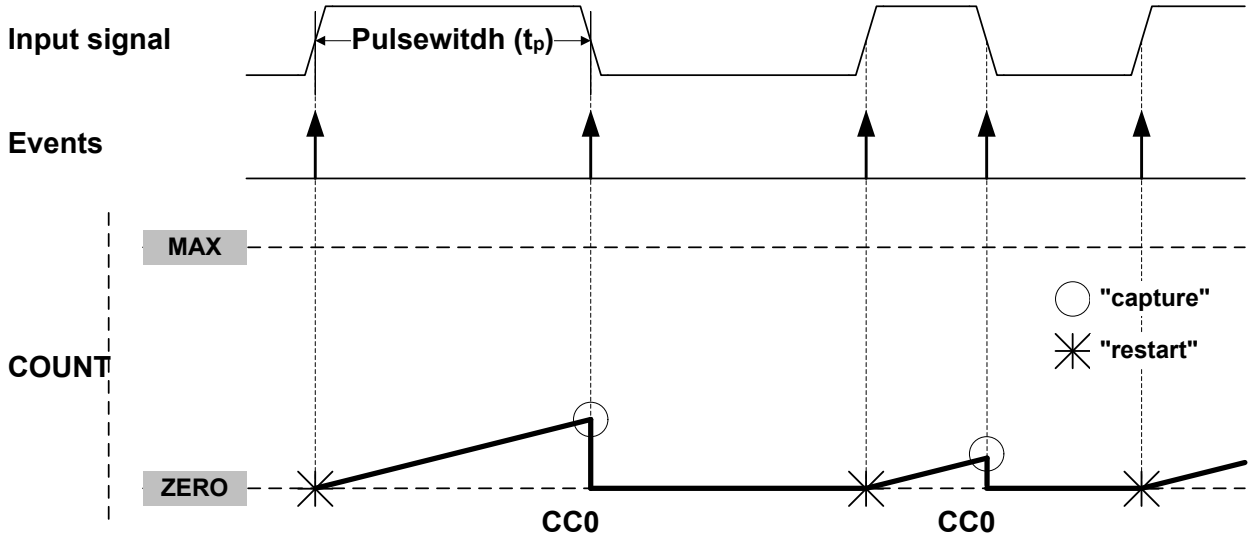
The TC can detect capture overflow of the input capture channels: When a new capture event is detected while the Capture Interrupt flag (INTFLAG.MCx) is still set, the new timestamp will not be stored and INTFLAG.ERR will be set.

Note: The corresponding capture is working only if the channel is enabled in capture mode (CTRLA.CAPTENx=1). Consequently, both channels must be enabled in order to fully characterize the input.

39.6.2.8.3 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 DRVCTRL.INVEN or EVCTRL.TCEINV).

Figure 39-14. 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.

39.6.3 Additional Features

39.6.3.1 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 zero.

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 will count until an overflow or underflow occurs and stops counting operation. The one-shot operation can be restarted by a re-trigger software command, a retrigger event, or a start event. When the counter restarts its operation, STATUS.STOP is automatically cleared.

39.6.3.2 Time-Stamp Capture on Events or I/Os

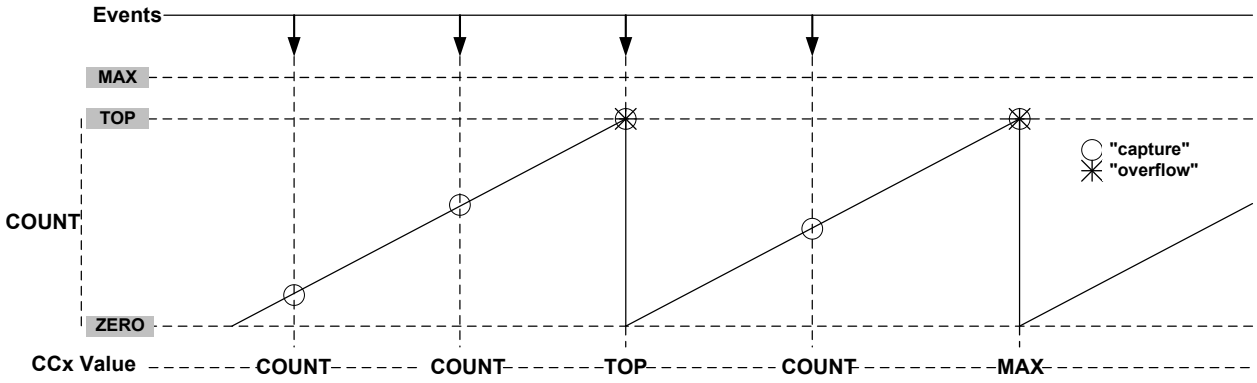
This feature is enabled when the Capture Time Stamp (STAMP) Event Action in Event Control register (EVCTRL.EVACT) is selected. The counter TOP value must be smaller than MAX.

When a capture event 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 39-15. Time-Stamp



39.6.3.3 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 captures will be performed using the corresponding channel.

MCx Behaviour:

In CAPTMIN operation, capture is performed only when on capture event time, the counter value is lower than the last captured value. The MCx interrupt flag is set only when on capture event time, the counter value is upper or equal to the value captured on the previous event. So interrupt flag is set when a new absolute local Minimum value has been detected.

39.6.3.4 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).

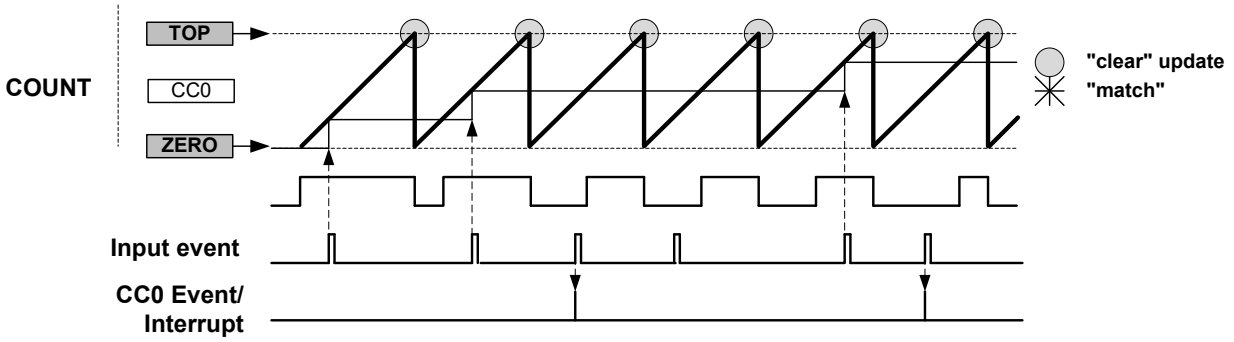
CCx Content:

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 captures will be performed using the corresponding channel.

MCx Behaviour:

In CAPTMAX operation, capture is performed only when on capture event time, the counter value is upper than the last captured value. The MCx interrupt flag is set only when on capture event time, the counter value is lower or equal to the value captured on the previous event. So interrupt flag is set when a new absolute local Maximum value has been detected.

Figure 39-16. Maximum Capture Operation with CC0 Initialized with ZERO Value



39.6.4 DMA Operation

The TC can generate the following DMA requests:

- Overflow (OVF): the request is set when an update condition (overflow, underflow or retrigger) 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.

39.6.5 Interrupts

The TC has the following interrupt sources:

- Overflow/Underflow (OVF)
- Match or Capture Channel x (MC0-1)
- 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).

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. Refer to the INTFLAG register for details on how to clear 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.

Interrupts must be globally enabled for interrupt requests to be generated. Refer to the Nested Vector Interrupt Controller for additional information.

39.6.6 Events

The TC can generate the following output events:

- Overflow/Underflow (OVF)
- Match or Capture Channel x (MC0-1)

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)
- Retrigger 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 (EVU) to the TC. Writing a '0' to this bit disables input events to the TC.

39.6.7 Sleep Mode Operation

The TC can be configured to operate in any sleep mode. To be able to run in standby, 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 STATUS register (STATUS.STOP) is set to '1'. When a retrigger or start condition is detected, the TC requests the clock before the operation starts.

39.6.8 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 - refer to the Debug Control (DBGCTRL) register for details.

39.6.9 Synchronization

Some registers (or bit fields within a register) require synchronization when read and/or written.

Synchronization is denoted by the "Read-Synchronized" (or "Read-Synchronized Bits”) and/or "Write-Synchronized" (or "Write-Synchronized Bits”) property in each individual register description.

For more details, refer to Register Synchronization.

39.7 Register Summary - 8-bit Mode

This Register Description section is valid if the TC is in 8-bit mode (CTRLA.MODE=1).

Refer to the Registers Description section for more details on register properties and access permissions.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRLA	31:24				CAPTMODE1[1:0]			CAPTMODE0[1:0]	
		23:16			COPEN1	COPEN0			CAPTEN1	CAPTEN0
		15:8					ALOCK	PRESCALER[2:0]		
		7:0	ONDEMAND	RUNSTDBY	PRESCSYNC[1:0]		MODE[1:0]		ENABLE	SWRST
0x04	CTRLBCLR	7:0	CMD[2:0]					ONESHOT	LUPD	DIR
0x05	CTRLBSET	7:0	CMD[2:0]					ONESHOT	LUPD	DIR
0x06	EVCTRL	15:8			MCEO1	MCEO0				OVFEO
		7:0			TCEI	TCINV		EVACT[2:0]		
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		SLAVE	STOP
0x0C	WAVE	7:0							WAVEGEN[1:0]	
0x0D	DRVCTRL	7:0							INVEN1	INVEN0
0x0E	Reserved									
0x0F	DBGCTRL	7:0								DBGRUN
0x10	SYNCBUSY	31:24								
		23:16								
		15:8								
		7:0	CC1	CC0	PER	COUNT	STATUS	CTRLB	ENABLE	SWRST
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]							

39.7.1 Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00000000
Property: PAC Write-Protection, Write-Synchronized Bits, Enable-Protected Bits

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
					ALOCK	PRESCALER[2:0]		
Access					R/W	R/W	R/W	R/W
Reset					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.
Note: This bit field is enable-protected. This bit field is not synchronized.

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.
Note: This bit field is enable-protected. This bit field is not synchronized.

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
Bit x of COPEN[1:0] selects the trigger source for capture operation, either events or I/O pin input.
Note: This bit is enable-protected. 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
Bit x of CAPTEN[1:0] selects whether channel x is a capture or a compare channel.

Note: This bit is enable-protected. This bit is not synchronized.

Value	Description
0	CAPTEN disables capture on channel x.
1	CAPTEN enables capture on channel x.

Bit 11 – ALOCK Auto Lock

When this bit is set, Lock bit update (LUPD) is set to '1' on each overflow/underflow or retrigger event.

Note: This bit is enable-protected. This bit is not synchronized.

Value	Description
0	The LUPD bit is not affected on overflow/underflow, and retrigger event.
1	The LUPD bit is set on each overflow/underflow or retrigger event.

Bits 10:8 – PRESCALER[2:0] Prescaler

These bits select the counter prescaler factor.

Note: This bit field is enable-protected. This bit field is 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'.

Note: This bit is enable-protected. This bit is not synchronized.

Value	Description
0	The On Demand is disabled. If On Demand is disabled, the TC will continue 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 retrigger 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.

Note: This bit is enable-protected. 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 should wrap around on the next GCLK_TCx clock or the next prescaled GCLK_TCx clock. It also makes it possible to reset the prescaler.

Note: This bit field is enable-protected. This bit field is not synchronized.

Value	Name	Description
0x0	GCLK	Reload or reset the counter on next generic clock
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.

Note: This bit field is enable-protected. This bit field is 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

Notes:

1. This bit is write-synchronized: SYNCBUSY.ENABLE must be checked to ensure the CTRLA.ENABLE synchronization is complete.
2. 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 will be disabled.

Writing a '1' to CTRLA.SWRST will always take precedence; all other writes in the same write-operation will be discarded.

Notes:

1. When the CTRLA.SWRST is written, the user must poll SYNCBUSY.SWRST bit to know when the reset operation is complete.
2. During a SWRST, access to registers/bits without SWRST are disallowed until SYNCBUSY.SWRST is cleared by hardware.
3. This bit is not enable protected.

39.7.2 Control B Clear

Name: CTRLBCLR
Offset: 0x04
Reset: 0x00
Property: PAC Write-Protection, Read-Synchronized, Write-Synchronized

This register allows the user to change this register without doing a read-modify-write operation. Changes in this register will also be reflected in the Control B Set register (CTRLBSET).
Note: This register is read- and write-synchronized: SYNCBUSY.CTRLB must be checked to ensure the CTRLBCLR register synchronization is complete.

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
Writing a `0' to these bits has no effect.
Writing a '1' to any of these bits will clear 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 will disable 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 any update of the registers with value of its buffered register is performed on hardware UPDATE condition. Locking the update ensures that all buffer registers are valid before an hardware update is performed. After all the buffer registers are loaded correctly, the buffered registers can be unlocked.
This bit has no effect when input capture operation is enabled.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear 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 will clear the bit and make the counter count up.

Value	Description
0	The timer/counter is counting up (incrementing).
1	The timer/counter is counting down (decrementing).

39.7.3 Control B Set

Name: CTRLBSET
Offset: 0x05
Reset: 0x00
Property: PAC Write-Protection, Read-Synchronized, Write-Synchronized

This register allows the user to change this register without doing a read-modify-write operation. Changes in this register will also be reflected in the Control B Clear register (CTRLBCLR).
Note: This register is read- and write-synchronized: SYNCBUSY.CTRLB must be checked to ensure the CTRLBCLR register synchronization is complete.

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 has been executed, the CMD bit group will be read back as zero.
Writing 0x0 to these bits has no effect.
Writing a value different from 0x0 from the following table will issue 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 will enable 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 any update of the registers with value of its buffered register is performed on hardware UPDATE condition. Locking the update ensures that all buffer registers are valid before an 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 will set 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 will set the bit and make the counter count down.

Value	Description
0	The timer/counter is counting up (incrementing).

Value	Description
1	The timer/counter is counting down (decrementing).

39.7.4 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

39.7.5 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
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 Enable
Writing a '0' to this bit 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 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 request.

Value	Description
0	The Overflow interrupt is disabled.
1	The Overflow interrupt is enabled.

39.7.6 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
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 this bit 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 this bit 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.

39.7.7 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
This flag is set on a comparison match, or when the corresponding CCx register contains a valid capture value. This flag 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 one of 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 nowhere 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 0 – OVF Overflow Interrupt Flag
This flag is set after an overflow condition occurs, and will generate an interrupt request if INTENSET.OVF is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Overflow interrupt flag.

39.7.8 Status

Name: STATUS
Offset: 0x0B
Reset: 0x01
Property: Read-Synchronized, Write-Synchronized

Note: This register is read- and write-synchronized: SYNCBUSY.STATUS must be checked to ensure the STATUS register synchronization is complete.

Bit	7	6	5	4	3	2	1	0
			CCBUFV1	CCBUFV0	PERBUFV		SLAVE	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
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 UPDATE condition.

Bit 1 – SLAVE Client Status Flag
This bit is only available in 32-bit mode on the client (TC1). The bit is set when the associated host (TC0) 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.

39.7.9 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 39.6.2.6.1. Waveform Output Operations. They also control whether frequency or PWM waveform generation should be used. The waveform generation operations are explained in 39.6.2.6.1. Waveform Output Operations.
These bits are not synchronized.

Value	Name	Operation	Top Value	Output Waveform on Match	Output Waveform on Wraparound
0x0	NFRQ	Normal frequency	PER	Toggle	No action
0x1	MFRQ	Match frequency	CC0	Toggle	No action
0x2	NPWM	Normal PWM	PER	Set	Clear
0x3	MPWM	Match PWM	CC0	Set	Clear

39.7.10 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
INVENx bit 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.

39.7.11 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 should 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.

39.7.12 Synchronization Busy

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

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

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 CC channels number, refer to each TC feature list.
This bit is cleared when the synchronization of CCx between the clock domains is complete.
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 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 PERBUF is written, and cleared on 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 CTRLBSET or CTRLBCLR between the clock domains is complete.
This bit is set when the synchronization of CTRLBSET or CTRLBCLR 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.

39.7.13 Counter Value, 8-bit Mode

Name: COUNT
Offset: 0x14
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized

Note: Prior to any read access, this register must be synchronized by the user by writing the TC Command value to the Control B Set register (CTRLBSET.CMD=READSYNC).

Note: This register is write-synchronized: SYNCBUSY.COUNT must be checked to ensure the COUNT register synchronization is complete.

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.

39.7.14 Period Value, 8-bit Mode

Name: PER
Offset: 0x1B
Reset: 0xFF
Property: Write-Synchronized

Note: This register is write-synchronized: SYNCBUSY.PER must be checked to ensure the PER register synchronization is complete.

Important: In 8-bit Mode, PER register updates using the DMA are not possible in standby mode.

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	1	1	1	1	1	1	1	1

Bits 7:0 – PER[7:0] Period Value
These bits hold the value of the TC period count.

39.7.15 Channel x Compare/Capture Value, 8-bit Mode

Name: CCx
Offset: 0x1C + x*0x01 [x=0..1]
Reset: 0x00
Property: Write-Synchronized

Note: This register is write-synchronized: SYNCBUSY.CCx must be checked to ensure the CCx register synchronization is complete.

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.

39.7.16 Period Buffer Value, 8-bit Mode

Name: PERBUF
Offset: 0x2F
Reset: 0xFF
Property: Write-Synchronized

Note: This register is write-synchronized: SYNCBUSY.PER must be checked to ensure the PER register synchronization is complete.

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 PER register on UPDATE condition.

39.7.17 Channel x Compare Buffer Value, 8-bit Mode

Name: CCBUFx
Offset: 0x30 + x*0x01 [x=0..1]
Reset: 0x00
Property: Write-Synchronized

Note: This register is write-synchronized: SYNCBUSY.CCx must be checked to ensure the CCx register synchronization is complete.

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
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 will be copied into the corresponding CCx register under UPDATE condition, including the software update command (CTRLBSET.CMD=0x3).

39.8 Register Summary - 16-bit Mode

This Register Description section is valid if the TC is in 16-bit mode (CTRLA.MODE=0).

Refer to the Registers Description section for more details on register properties and access permissions.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRLA	31:24				CAPTMODE1[1:0]			CAPTMODE0[1:0]	
		23:16			COPEN1	COPEN0			CAPTEN1	CAPTEN0
		15:8					ALOCK	PRESCALER[2:0]		
		7:0	ONDEMAND	RUNSTDBY	PRESCSYNC[1:0]		MODE[1:0]		ENABLE	SWRST
0x04	CTRLBCLR	7:0	CMD[2:0]					ONESHOT	LUPD	DIR
0x05	CTRLBSET	7:0	CMD[2:0]					ONESHOT	LUPD	DIR
0x06	EVCTRL	15:8			MCEO1	MCEO0				OVFEO
		7:0			TCEI	TCINV		EVACT[2:0]		
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		SLAVE	STOP
0x0C	WAVE	7:0							WAVEGEN[1:0]	
0x0D	DRVCTRL	7:0							INVEN1	INVEN0
0x0E	Reserved									
0x0F	DBGCTRL	7:0								DBGRUN
0x10	SYNCBUSY	31:24								
		23:16								
		15:8								
		7:0	CC1	CC0	PER	COUNT	STATUS	CTRLB	ENABLE	SWRST
0x14	COUNT	15:8	COUNT[15:8]							
		7:0	COUNT[7:0]							
0x16 ... 0x19	Reserved									
0x1A	PER	15:8	PER[15:8]							
		7:0	PER[7:0]							
0x1C	CC0	15:8	CC[15:8]							
		7:0	CC[7:0]							
0x1E	CC1	15:8	CC[15:8]							
		7:0	CC[7:0]							
0x20 ... 0x2D	Reserved									
0x2E	PERBUF	15:8	PERBUF[15:8]							
		7:0	PERBUF[7:0]							
0x30	CCBUF0	15:8	CCBUF[15:8]							
		7:0	CCBUF[7:0]							
0x32	CCBUF1	15:8	CCBUF[15:8]							
		7:0	CCBUF[7:0]							

39.8.1 Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00000000
Property: PAC Write-Protection, Write-Synchronized Bits, Enable-Protected Bits

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
					ALOCK	PRESCALER[2:0]		
Access					R/W	R/W	R/W	R/W
Reset					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.
Note: This bit field is enable-protected. This bit field is not synchronized.

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.
Note: This bit field is enable-protected. This bit field is not synchronized.

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
Bit x of COPEN[1:0] selects the trigger source for capture operation, either events or I/O pin input.
Note: This bit is enable-protected. 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
Bit x of CAPTEN[1:0] selects whether channel x is a capture or a compare channel.

Note: This bit is enable-protected. This bit is not synchronized.

Value	Description
0	CAPTEN disables capture on channel x.
1	CAPTEN enables capture on channel x.

Bit 11 – ALOCK Auto Lock

When this bit is set, Lock bit update (LUPD) is set to '1' on each overflow/underflow or retrigger event.

Note: This bit is enable-protected. This bit is not synchronized.

Value	Description
0	The LUPD bit is not affected on overflow/underflow, and retrigger event.
1	The LUPD bit is set on each overflow/underflow or retrigger event.

Bits 10:8 – PRESCALER[2:0] Prescaler

These bits select the counter prescaler factor.

Note: This bit field is enable-protected. This bit field is 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'.

Note: This bit is enable-protected. This bit is not synchronized.

Value	Description
0	The On Demand is disabled. If On Demand is disabled, the TC will continue 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 retrigger 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.

Note: This bit is enable-protected. 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 should wrap around on the next GCLK_TCx clock or the next prescaled GCLK_TCx clock. It also makes it possible to reset the prescaler.

Note: This bit field is enable-protected. This bit field is not synchronized.

Value	Name	Description
0x0	GCLK	Reload or reset the counter on next generic clock
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.

Note: This bit field is enable-protected. This bit field is 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

Notes:

1. This bit is write-synchronized: SYNCBUSY.ENABLE must be checked to ensure the CTRLA.ENABLE synchronization is complete.
2. 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 will be disabled.

Writing a '1' to CTRLA.SWRST will always take precedence; all other writes in the same write-operation will be discarded.

Notes:

1. When the CTRLA.SWRST is written, the user must poll SYNCBUSY.SWRST bit to know when the reset operation is complete.
2. During a SWRST, access to registers/bits without SWRST are disallowed until SYNCBUSY.SWRST is cleared by hardware.
3. This bit is not enable protected.

39.8.2 Control B Clear

Name: CTRLBCLR
Offset: 0x04
Reset: 0x00
Property: PAC Write-Protection, Read-Synchronized, Write-Synchronized

This register allows the user to change this register without doing a read-modify-write operation. Changes in this register will also be reflected in the Control B Set register (CTRLBSET).
Note: This register is read- and write-synchronized: SYNCBUSY.CTRLB must be checked to ensure the CTRLBCLR register synchronization is complete.

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
Writing a `0' to these bits has no effect.
Writing a '1' to any of these bits will clear 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 will disable 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 any update of the registers with value of its buffered register is performed on hardware UPDATE condition. Locking the update ensures that all buffer registers are valid before an hardware update is performed. After all the buffer registers are loaded correctly, the buffered registers can be unlocked.
This bit has no effect when input capture operation is enabled.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear 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 will clear the bit and make the counter count up.

Value	Description
0	The timer/counter is counting up (incrementing).
1	The timer/counter is counting down (decrementing).

39.8.3 Control B Set

Name: CTRLBSET
Offset: 0x05
Reset: 0x00
Property: PAC Write-Protection, Read-Synchronized, Write-Synchronized

This register allows the user to change this register without doing a read-modify-write operation. Changes in this register will also be reflected in the Control B Clear register (CTRLBCLR).
Note: This register is read- and write-synchronized: SYNCBUSY.CTRLB must be checked to ensure the CTRLBCLR register synchronization is complete.

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 has been executed, the CMD bit group will be read back as zero.
Writing 0x0 to these bits has no effect.
Writing a value different from 0x0 from the following table will issue 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 will enable 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 any update of the registers with value of its buffered register is performed on hardware UPDATE condition. Locking the update ensures that all buffer registers are valid before an 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 will set 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 will set the bit and make the counter count down.

Value	Description
0	The timer/counter is counting up (incrementing).

Value	Description
1	The timer/counter is counting down (decrementing).

39.8.4 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

39.8.5 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
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 Enable
Writing a '0' to this bit 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 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 request.

Value	Description
0	The Overflow interrupt is disabled.
1	The Overflow interrupt is enabled.

39.8.6 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
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 this bit 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 this bit 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.

39.8.7 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
This flag is set on a comparison match, or when the corresponding CCx register contains a valid capture value. This flag 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 one of 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 nowhere 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 0 – OVF Overflow Interrupt Flag
This flag is set after an overflow condition occurs, and will generate an interrupt request if INTENSET.OVF is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Overflow interrupt flag.

39.8.8 Status

Name: STATUS
Offset: 0x0B
Reset: 0x01
Property: Read-Synchronized, Write-Synchronized

Note: This register is read- and write-synchronized: SYNCBUSY.STATUS must be checked to ensure the STATUS register synchronization is complete.

Bit	7	6	5	4	3	2	1	0
			CCBUFV1	CCBUFV0	PERBUFV		SLAVE	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
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 UPDATE condition.

Bit 1 – SLAVE Client Status Flag
This bit is only available in 32-bit mode on the client (TC1). The bit is set when the associated host (TC0) 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.

39.8.9 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 39.6.2.6.1. Waveform Output Operations. They also control whether frequency or PWM waveform generation should be used. The waveform generation operations are explained in 39.6.2.6.1. Waveform Output Operations.
These bits are not synchronized.

Value	Name	Operation	Top Value	Output Waveform on Match	Output Waveform on Wraparound
0x0	NFRQ	Normal frequency	PER	Toggle	No action
0x1	MFRQ	Match frequency	CC0	Toggle	No action
0x2	NPWM	Normal PWM	PER	Set	Clear
0x3	MPWM	Match PWM	CC0	Set	Clear

39.8.10 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
INVENx bit 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.

39.8.11 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 should 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.

39.8.12 Synchronization Busy

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

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

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 CC channels number, refer to each TC feature list.
This bit is cleared when the synchronization of CCx between the clock domains is complete.
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 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 PERBUF is written, and cleared on 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 CTRLBSET or CTRLBCLR between the clock domains is complete.
This bit is set when the synchronization of CTRLBSET or CTRLBCLR 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.

39.8.13 Counter Value, 16-bit Mode

Name: COUNT
Offset: 0x14
Reset: 0x0000
Property: PAC Write-Protection, Write-Synchronized

Note: Prior to any read access, this register must be synchronized by the user by writing the TC Command value to the Control B Set register (CTRLBSET.CMD=READSYNC).

Note: 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 contain the current counter value.

39.8.14 Period Value, 16-bit Mode

Name: PER
Offset: 0x1A
Reset: 0xFFFF
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	1	1	1	1	1	1	1	1

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	1	1	1	1	1	1	1	1

Bits 15:0 – PER[15:0] Period Value
These bits hold the value of the TC period count.

39.8.15 Channel x Compare/Capture Value, 16-bit Mode

Name: CCx
Offset: 0x1C + x*0x02 [x=0..1]
Reset: 0x0000
Property: Write-Synchronized

Note: This register is write-synchronized: SYNCBUSY.CCx must be checked to ensure the CCx register synchronization is complete.

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
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.

39.8.16 Period Buffer Value, 16-bit Mode

Name: PERBUF
Offset: 0x2E
Reset: 0xFFFF
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
	PERBUF[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
	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 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.

39.8.17 Channel x Compare Buffer Value, 16-bit Mode

Name: CCBUFx
Offset: 0x30 + x*0x02 [x=0..1]
Reset: 0x0000
Property: Write-Synchronized

Note: This register is write-synchronized: SYNCBUSY.CCx must be checked to ensure the CCx register synchronization is complete.

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
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 will be copied into the corresponding CCx register under UPDATE condition, including the software update command (CTRLBSET.CMD=0x3).

39.9 Register Summary - 32-bit Mode

This Register Description section is valid if the TC is in 32-bit mode (CTRLA.MODE=2).

Refer to the Registers Description section for more details on register properties and access permissions.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRLA	31:24				CAPTMODE1[1:0]			CAPTMODE0[1:0]	
		23:16			COPEN1	COPEN0			CAPTEN1	CAPTEN0
		15:8					ALOCK	PRESCALER[2:0]		
		7:0	ONDEMAND	RUNSTDBY	PRESCSYNC[1:0]		MODE[1:0]		ENABLE	SWRST
0x04	CTRLBCLR	7:0	CMD[2:0]					ONESHOT	LUPD	DIR
0x05	CTRLBSET	7:0	CMD[2:0]					ONESHOT	LUPD	DIR
0x06	EVCTRL	15:8			MCEO1	MCEO0				OVFEO
		7:0			TCEI	TCINV		EVACT[2:0]		
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		SLAVE	STOP
0x0C	WAVE	7:0							WAVEGEN[1:0]	
0x0D	DRVCTRL	7:0							INVEN1	INVEN0
0x0E	Reserved									
0x0F	DBGCTRL	7:0								DBGRUN
0x10	SYNCBUSY	31:24								
		23:16								
		15:8								
		7:0	CC1	CC0	PER	COUNT	STATUS	CTRLB	ENABLE	SWRST
0x14	COUNT	31:24	COUNT[31:24]							
		23:16	COUNT[23:16]							
		15:8	COUNT[15:8]							
		7:0	COUNT[7:0]							
0x18	PER	31:24	PER[31:24]							
		23:16	PER[23:16]							
		15:8	PER[15:8]							
		7:0	PER[7:0]							
0x1C	CC0	31:24	CC[31:24]							
		23:16	CC[23:16]							
		15:8	CC[15:8]							
		7:0	CC[7:0]							
0x20	CC1	31:24	CC[31:24]							
		23:16	CC[23:16]							
		15:8	CC[15:8]							
		7:0	CC[7:0]							
0x24 ... 0x2B	Reserved									
0x2C	PERBUF	31:24	PERBUF[31:24]							
		23:16	PERBUF[23:16]							
		15:8	PERBUF[15:8]							
		7:0	PERBUF[7:0]							
0x30	CCBUF0	31:24	CCBUF[31:24]							
		23:16	CCBUF[23:16]							
		15:8	CCBUF[15:8]							
		7:0	CCBUF[7:0]							
0x34	CCBUF1	31:24	CCBUF[31:24]							
		23:16	CCBUF[23:16]							
		15:8	CCBUF[15:8]							
		7:0	CCBUF[7:0]							

39.9.1 Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00000000
Property: PAC Write-Protection, Write-Synchronized Bits, Enable-Protected Bits

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
					ALOCK	PRESCALER[2:0]		
Access					R/W	R/W	R/W	R/W
Reset					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.
Note: This bit field is enable-protected. This bit field is not synchronized.

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.
Note: This bit field is enable-protected. This bit field is not synchronized.

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
Bit x of COPEN[1:0] selects the trigger source for capture operation, either events or I/O pin input.
Note: This bit is enable-protected. 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
Bit x of CAPTEN[1:0] selects whether channel x is a capture or a compare channel.

Note: This bit is enable-protected. This bit is not synchronized.

Value	Description
0	CAPTEN disables capture on channel x.
1	CAPTEN enables capture on channel x.

Bit 11 – ALOCK Auto Lock

When this bit is set, Lock bit update (LUPD) is set to '1' on each overflow/underflow or retrigger event.

Note: This bit is enable-protected. This bit is not synchronized.

Value	Description
0	The LUPD bit is not affected on overflow/underflow, and retrigger event.
1	The LUPD bit is set on each overflow/underflow or retrigger event.

Bits 10:8 – PRESCALER[2:0] Prescaler

These bits select the counter prescaler factor.

Note: This bit field is enable-protected. This bit field is 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'.

Note: This bit is enable-protected. This bit is not synchronized.

Value	Description
0	The On Demand is disabled. If On Demand is disabled, the TC will continue 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 retrigger 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.

Note: This bit is enable-protected. 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 should wrap around on the next GCLK_TCx clock or the next prescaled GCLK_TCx clock. It also makes it possible to reset the prescaler.

Note: This bit field is enable-protected. This bit field is not synchronized.

Value	Name	Description
0x0	GCLK	Reload or reset the counter on next generic clock
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.

Note: This bit field is enable-protected. This bit field is 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

Notes:

1. This bit is write-synchronized: SYNCBUSY.ENABLE must be checked to ensure the CTRLA.ENABLE synchronization is complete.
2. 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 will be disabled.

Writing a '1' to CTRLA.SWRST will always take precedence; all other writes in the same write-operation will be discarded.

Notes:

1. When the CTRLA.SWRST is written, the user must poll SYNCBUSY.SWRST bit to know when the reset operation is complete.
2. During a SWRST, access to registers/bits without SWRST are disallowed until SYNCBUSY.SWRST is cleared by hardware.
3. This bit is not enable protected.

39.9.2 Control B Clear

Name: CTRLBCLR
Offset: 0x04
Reset: 0x00
Property: PAC Write-Protection, Read-Synchronized, Write-Synchronized

This register allows the user to change this register without doing a read-modify-write operation. Changes in this register will also be reflected in the Control B Set register (CTRLBSET).
Note: This register is read- and write-synchronized: SYNCBUSY.CTRLB must be checked to ensure the CTRLBCLR register synchronization is complete.

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
Writing a `0' to these bits has no effect.
Writing a '1' to any of these bits will clear 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 will disable 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 any update of the registers with value of its buffered register is performed on hardware UPDATE condition. Locking the update ensures that all buffer registers are valid before an hardware update is performed. After all the buffer registers are loaded correctly, the buffered registers can be unlocked.
This bit has no effect when input capture operation is enabled.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear 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 will clear the bit and make the counter count up.

Value	Description
0	The timer/counter is counting up (incrementing).
1	The timer/counter is counting down (decrementing).

39.9.3 Control B Set

Name: CTRLBSET
Offset: 0x05
Reset: 0x00
Property: PAC Write-Protection, Read-Synchronized, Write-Synchronized

This register allows the user to change this register without doing a read-modify-write operation. Changes in this register will also be reflected in the Control B Clear register (CTRLBCLR).
Note: This register is read- and write-synchronized: SYNCBUSY.CTRLB must be checked to ensure the CTRLBCLR register synchronization is complete.

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 has been executed, the CMD bit group will be read back as zero.
Writing 0x0 to these bits has no effect.
Writing a value different from 0x0 from the following table will issue 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 will enable 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 any update of the registers with value of its buffered register is performed on hardware UPDATE condition. Locking the update ensures that all buffer registers are valid before an 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 will set 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 will set the bit and make the counter count down.

Value	Description
0	The timer/counter is counting up (incrementing).

Value	Description
1	The timer/counter is counting down (decrementing).

39.9.4 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

39.9.5 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
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 Enable
Writing a '0' to this bit 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 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 request.

Value	Description
0	The Overflow interrupt is disabled.
1	The Overflow interrupt is enabled.

39.9.6 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
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 this bit 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 this bit 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.

39.9.7 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
This flag is set on a comparison match, or when the corresponding CCx register contains a valid capture value. This flag 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 one of 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 nowhere 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 0 – OVF Overflow Interrupt Flag
This flag is set after an overflow condition occurs, and will generate an interrupt request if INTENSET.OVF is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Overflow interrupt flag.

39.9.8 Status

Name: STATUS
Offset: 0x0B
Reset: 0x01
Property: Read-Synchronized, Write-Synchronized

Note: This register is read- and write-synchronized: SYNCBUSY.STATUS must be checked to ensure the STATUS register synchronization is complete.

Bit	7	6	5	4	3	2	1	0
			CCBUFV1	CCBUFV0	PERBUFV		SLAVE	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
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 UPDATE condition.

Bit 1 – SLAVE Client Status Flag
This bit is only available in 32-bit mode on the client (TC1). The bit is set when the associated host (TC0) 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.

39.9.9 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 39.6.2.6.1. Waveform Output Operations. They also control whether frequency or PWM waveform generation should be used. The waveform generation operations are explained in 39.6.2.6.1. Waveform Output Operations.
These bits are not synchronized.

Value	Name	Operation	Top Value	Output Waveform on Match	Output Waveform on Wraparound
0x0	NFRQ	Normal frequency	PER	Toggle	No action
0x1	MFRQ	Match frequency	CC0	Toggle	No action
0x2	NPWM	Normal PWM	PER	Set	Clear
0x3	MPWM	Match PWM	CC0	Set	Clear

39.9.10 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
INVENx bit 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.

39.9.11 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 should 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.

39.9.12 Synchronization Busy

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

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

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 CC channels number, refer to each TC feature list.
This bit is cleared when the synchronization of CCx between the clock domains is complete.
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 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 PERBUF is written, and cleared on 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 CTRLBSET or CTRLBCLR between the clock domains is complete.
This bit is set when the synchronization of CTRLBSET or CTRLBCLR 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.

39.9.13 Counter Value, 32-bit Mode

Name: COUNT
Offset: 0x14
Reset: 0x00000000
Property: PAC Write-Protection, Write-Synchronized

Note: Prior to any read access, this register must be synchronized by the user by writing the TC Command value to the Control B Set register (CTRLBSET.CMD=READSYNC).

Note: 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 contain the current counter value.

39.9.14 Period Value, 32-bit Mode

Name: PER
Offset: 0x18
Reset: 0xFFFFFFFF
Property: Write-Synchronized

Note: This register is write-synchronized: SYNCBUSY.PER must be checked to ensure the PER register synchronization is complete.

Bit	31	30	29	28	27	26	25	24
	PER[31:24]							
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	23	22	21	20	19	18	17	16
	PER[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
	PER[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
	PER[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 31:0 – PER[31:0] Period Value
These bits hold the value of the TC period count.

39.9.15 Channel x Compare/Capture Value, 32-bit Mode

Name: CCx
Offset: 0x1C + x*0x04 [x=0..1]
Reset: 0x00000000
Property: Write-Synchronized

Note: This register is write-synchronized: SYNCBUSY.CCx must be checked to ensure the CCx register synchronization is complete.

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
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.

39.9.16 Period Buffer Value, 32-bit Mode

Name: PERBUF
Offset: 0x2C
Reset: 0xFFFFFFFF
Property: Write-Synchronized

Note: This register is write-synchronized: SYNCBUSY.PER must be checked to ensure the PER register synchronization is complete.

Bit	31	30	29	28	27	26	25	24
	PERBUF[31:24]							
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	23	22	21	20	19	18	17	16
	PERBUF[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
	PERBUF[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
	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 31:0 – PERBUF[31:0] Period Buffer Value
These bits hold the value of the period buffer register. The value is copied to PER register on UPDATE condition.

39.9.17 Channel x Compare Buffer Value, 32-bit Mode

Name: CCBUFx
Offset: 0x30 + x*0x04 [x=0..1]
Reset: 0x00000000
Property: Write-Synchronized

Note: This register is write-synchronized: SYNCBUSY.CCx must be checked to ensure the CCx register synchronization is complete.

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
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 will be copied into the corresponding CCx register under UPDATE condition, including the software update command (CTRLBSET.CMD=0x3).

40. Timer/Counter for Control Applications (TCC)

40.1 Overview

The device provides 4 instances of the Timer/Counter for Control applications (TCC) peripheral.

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, may be reduced for some of the TCC instances.

40.2 Features

- Up to four 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, retrigger and overflow) are available for counter
  - One compare match/input capture event output for each channel
- Interrupts:
  - Overflow and retrigger interrupt

  - Compare match/input capture interrupt
  - Interrupt on fault detection
  - Counter cycle interrupt
  - Error condition interrupt
- Can be Used with DMA and can Trigger DMA Transactions

The following table lists the features for each TCC instance.

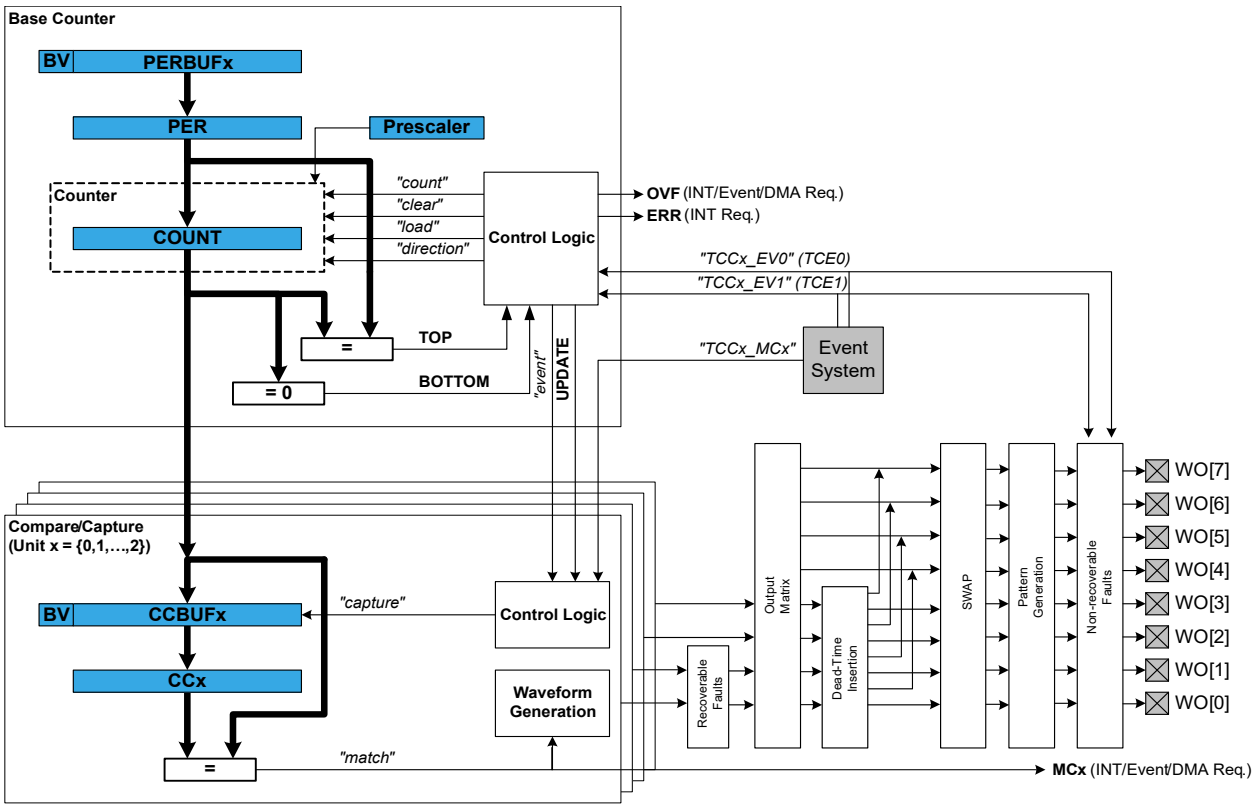
Table 40-1. TCC Configuration Summary

TCC#	Channels (CC_NUM)	Waveform Output (WO_NUM)	Counter size	Fault	Dithering	Output matrix	Dead Time Insertion (DTI)	SWAP	Pattern generation
0	4	8	24-bit	Yes	Yes	Yes	Yes	Yes	Yes
1	2	4	24-bit	Yes	Yes	-	-	-	Yes
2	2	2	16-bit	Yes	-	-	-	-	-
3	4	8	24-bit	Yes	Yes	Yes	Yes	Yes	Yes

Note: The number of CC registers (CC_NUM) for each TCC corresponds to the number of compare/capture channels, so that a TCC can have more Waveform Outputs (WO_NUM) than CC registers.

40.3 Block Diagram

Figure 40-1. Timer/Counter for Control Applications - Block Diagram



40.4 Signal Description

Pin Name	Type	Description
TCC/WO[0]	Digital output	Compare channel 0 waveform output
TCC/WO[1]	Digital output	Compare channel 1 waveform output
…	...	...
TCC/WO[WO_NUM-1]	Digital output	Compare channel (WO_NUM-1) waveform output

Refer to the Pinout for details on the pin mapping for this peripheral. One signal can be mapped on several pins.

40.5 Peripheral Dependencies

Table 40-2. TCC Configuration Summary

Peripheral name	Base address	NVIC IRQ Index	MCLK AHB/APB Clocks	GCLK Peripheral Channel Index (GCLK.PCHCTRL)	PAC Peripheral Identifier Index (PAC.WRCTRL)	DMA Trigger Source Index (DMAC.CHCTRLB)	Events (EVSYS)		Power Domain (PM.STDBYCFG)
							Users (EVSYS.USER)	Generators (EVSYS.CHANNEL)	
TCC0	0x42002800	58: ERR, MC0-3, OVF, TRG, CNT, DFS, UFS, FAULTA, FAULTB, FAULT0, FAULT1	CLK_TCC0_APB	25: GCLK_TCC0_TCC1	74	25: OVF 26-29: MC0-3	18-19: EVU0-1 20-23: MC0-3	50: TRG 51: CNT 52-55: MC0-3 56: OVF	PDSW
TCC1	0x42002C00	59: ERR, MC0-1, OVF, TRG, CNT, DFS, UFS, FAULTA, FAULTB, FAULT0, FAULT1	CLK_TCC1_APB	25: GCLK_TCC0_TCC1	75	30: OVF 31-32: MC0-1	24-25: EVU0-1 26-27: MC0-1	57: TRG 58: CNT 59-60: MC0-1 61: OVF	PDSW
TCC2	0x42003000	60: ERR, MC0-1, OVF, TRG, CNT, DFS, UFS, FAULTA, FAULTB, FAULT0, FAULT1	CLK_TCC2_APB	26: GCLK_TCC2	76	33: OVF 34-35: MC0-1	28-29: EVU0-1 30-31: MC0-1	62: TRG 63: CNT 64-65: MC0-1 66: OVF	PDSW
TCC3	0x42003400	61: ERR, MC0-3, OVF, TRG, CNT, DFS, UFS, FAULTA, FAULTB, FAULT0, FAULT1	CLK_TCC3_APB	27: GCLK_TCC3	77	36: OVF 37-40: MC0-3	32-33: EVU0-1 34-37: MC0-3	67: TRG 68: CNT 69-72: MC0-3 73: OVF	PDSW

40.6 Functional Description

40.6.1 Principle of Operation

The following definitions are used throughout the documentation:

Table 40-3. 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 40.6.2.5.1. Waveform Output Generation Operations.
ZERO	The counter reaches ZERO when it contains all zeroes.
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	TCC mode where increment/decrement/clear/reload steps are done on each prescaled clock.
Counter	TCC mode where increment/decrement/clear/reload steps are done on each detected 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 four 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, request DMA transactions or generate events for the Event System. In Waveform Generator mode, these comparisons are used to set the waveform period or pulse alignment.

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 or until the end of the current TCC cycle. This can typically be used for current sensing regulation, and zero-crossing and demagnetization retriggering.

The MCE0 and MCE1 asynchronous event sources are shared with the recoverable fault unit. Only asynchronous events must be used when fault unit extension is enabled. For further details on how to configure asynchronous events routing, refer to EVSYS – Event System.

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 also 40.6.3.5. Recoverable Faults.

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
- Dithering

See also Figure 40-1.

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.
Note: MCE0 and MCE1 can also be used as non-recoverable event source.

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. For further details on how to configure asynchronous events routing, refer to section EVSYS – Event System.

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 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 (CLK_TCCx_APB).
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).

Note: Two instances of the TCC (TCC0 and TCC1) may share a peripheral clock channel. In this case, they cannot be set to different clock frequencies. Refer to the peripheral clock channel mapping of the Generic Clock Controller (GCLK.PCHTRLm) to identify shared peripheral clocks.

40.6.2.2 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 zero 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, will be reset to their initial state, and the TCC will be disabled. Refer to Control A (40.7.1. CTRLA) register for details.

The TCC should be disabled before the TCC is reset to avoid undefined behavior.

40.6.2.3 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.

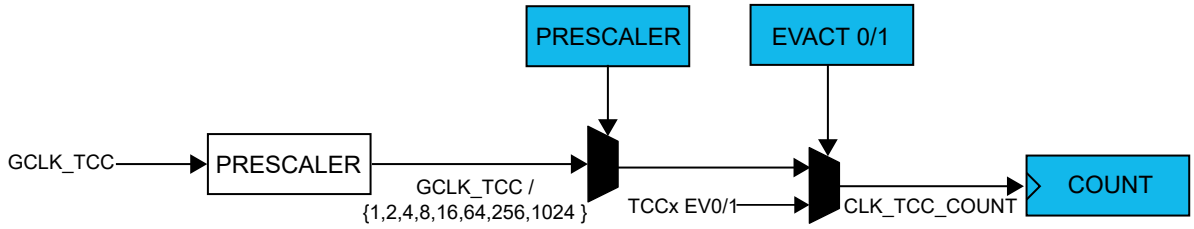
If the prescaler value is higher than one, the Counter Update condition can be optionally executed on the next GCLK_TCC clock pulse or the next prescaled clock pulse. For further details, refer to the Prescaler (CTRLA.PRESCALER) and Counter Synchronization (CTRLA.PRESYNC) descriptions.

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_TCC_COUNT.

Figure 40-2. Prescaler



40.6.2.4 Counter Operation

Depending on the mode of operation, the counter is cleared, reloaded, incremented, or decremented at each TCC clock input (CLK_TCC_COUNT). A counter clear or reload mark the end of current counter cycle and the start of a new one.

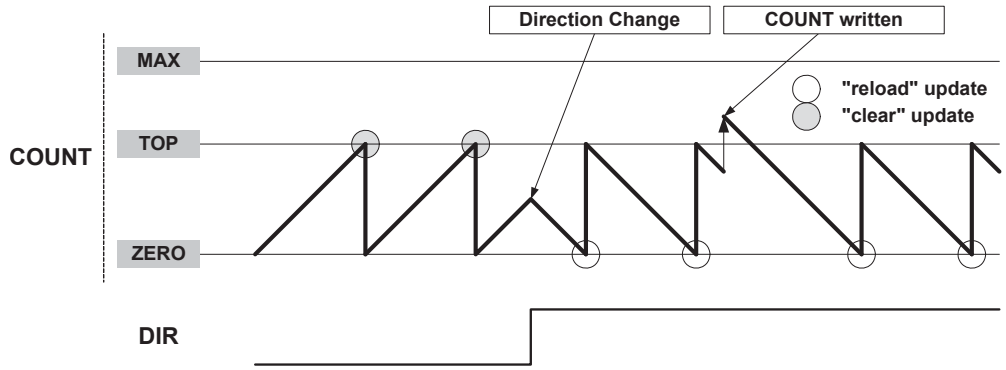
The counting direction is set by the Direction bit in the Control B register (CTRLB.DIR). If the bit is zero, it's counting up and one if counting down.

The counter will count up or down for each tick (clock or event) until it reaches TOP or ZERO. When it's 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 down-counting, the counter is reloaded with the TOP value when ZERO is reached (underflow), and INTFLAG.OVF is set.

Note: When the TCC is counting down, the COUNT register must be initialized to the TOP value (PER or CC0 value depending on the mode).

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). The One-Shot feature is explained in the Additional Features section.

Figure 40-3. 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 start from ZERO or TOP, depending on direction set by CTRLBSET.DIR or CTRLBCLR.DIR, when starting the TCC, unless a different value has been written to it, or the TCC 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 during normal operation.

Stop Command

A stop command can be issued from software by using TCC Command bits in Control B Set register (CTRLBSET.CMD=0x2, STOP).

When a stop is detected while the counter is running, the counter will maintain its current value. If the waveform generation (WG) is used, all waveforms are set to a state defined in Non-Recoverable State x Output Enable bit and Non- Recoverable State x Output Value bit in the Driver Control register (DRVCTRL.NREx and DRVCTRL.NRVx), and the Stop bit in the Status register is set (STATUS.STOP).

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, as long as a start event action is detected: Input Event Action 0 bits in Event Control register (EVCTRL.EVACT0=0x3, START).

ReTrigger Command and Event Action

A retrigger command can be issued from software by using TCC Command bits in Control B Set register (CTRLBSET.CMD=0x1, RETRIGGER), or from event when the retrigger 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 counting operation, the counter will be reloaded or cleared, depending on the counting direction (CTRLBSET.DIR or CTRLBCLR.DIR). The retrigger bit in the Interrupt Flag Status and Clear register will be set (INTFLAG.TRG). It is also possible to generate an event by writing a '1' to the retrigger Event Output Enable bit in the Event Control register (EVCTRL.TRGEO). If the retrigger command is detected when the counter is stopped, the counter will resume counting operation from the value in COUNT.

Note:
When a retrigger 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 will start on the next incoming event and restart on corresponding following event.

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 retrigger 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 will start on the next incoming event, but it will not restart on subsequent events.

Count Event Action

The TCC 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 is selected by the Event Action 0 bit group in the Event Control register (EVCTRL.EVACT0=0x2, COUNTEV).

Count on Active State of Asynchronous Event

The TCC counts during the active state of an asynchronous event (increment or decrement, depending on counter direction).

The count event action is selected by the Event Action 0 bit group in the Event Control register (EVCTRL.EVACT0=0x5, COUNT).

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.

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, whatever the direction setting (CTRLBSET.DIR or CTRLBCLR.DIR) is.

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, whatever the direction setting (CTRLBSET.DIR or CTRLBCLR.DIR) is.

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 will be 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.

Event Action Off

If the event action is disabled (EVCTRL.EVACTn=0x0, OFF), enabling the counter will also start the counter.

40.6.2.5 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). For further details, refer to 40.6.2.6. Double Buffering. The synchronization prevents the occurrence of odd-length, non-symmetrical pulses and ensures glitch-free output.

40.6.2.5.1 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. Refer to PORT - I/O Pin Controller for details.
   Note: Events MCx must not be used except as non-recoverable fault input 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 (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 on the same condition. The same condition generates a DMA request.

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 Compare PWM (DPWM)
- 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 retrigger.

The table below shows the update counter and overflow event/interrupt generation conditions in different operation modes.

Table 40-4. Counter Update and Overflow Event/interrupt Conditions

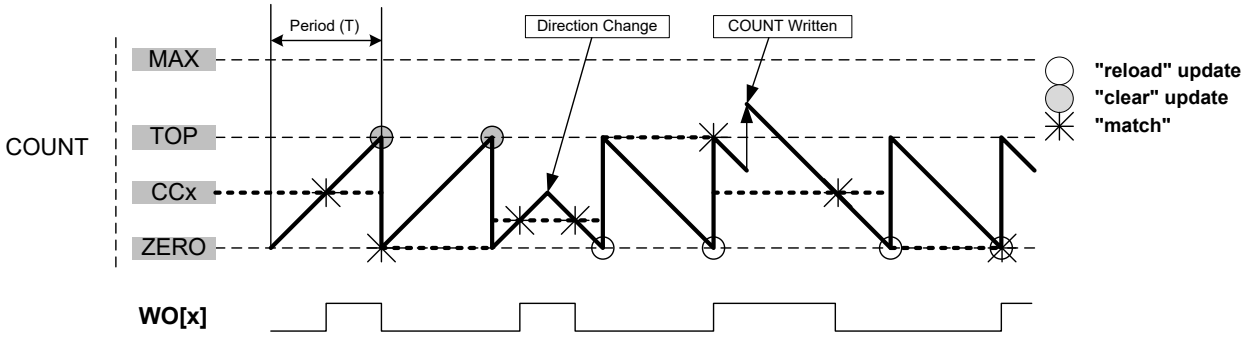
Name	Operation	TOP	Update	Output Waveform		OVFIF/Event	
				On Match	On Update	Up	Down
NFRQ	Normal Frequency	PER	TOP/ ZERO	Toggle	Stable	TOP	ZERO
MFRQ	Match Frequency	CC0	TOP/ ZERO	Toggle	Stable	TOP	ZERO
NPWM / DPWM	Single-slope PWM	PER	TOP/ ZERO	See section 'Output Polarity' below		TOP	ZERO
DSCRITICAL	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.

40.6.2.5.2 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 (INTFLAG.MCx) will be set.

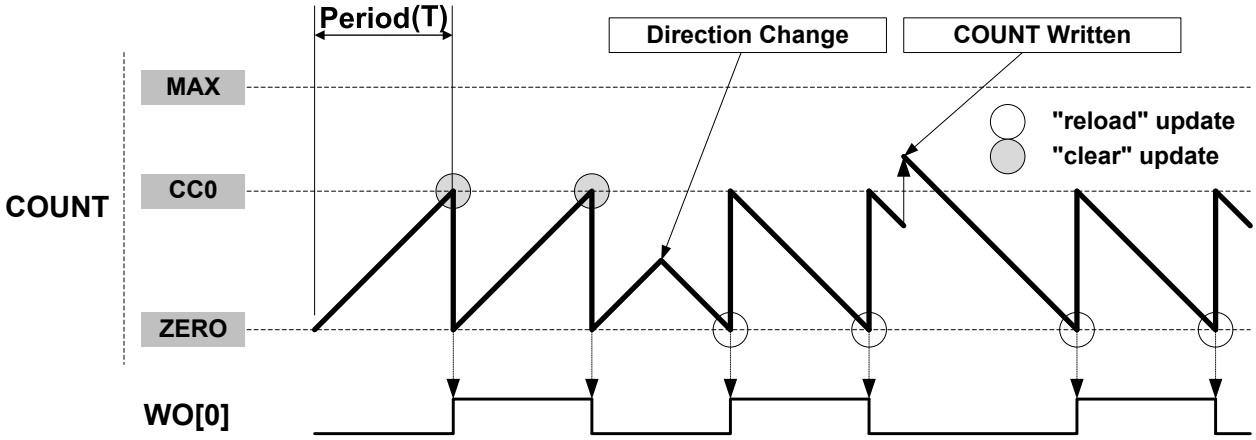
Figure 40-4. Normal Frequency Operation



40.6.2.5.3 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 40-5. Match Frequency Operation



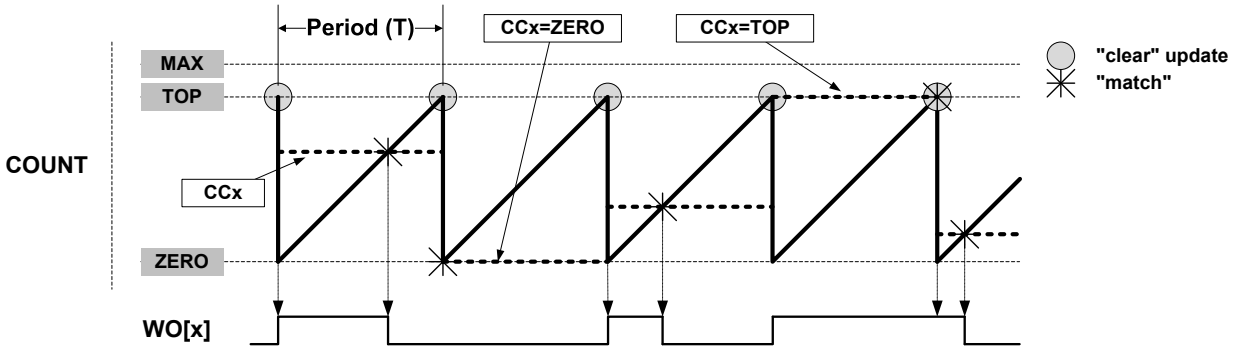
40.6.2.5.4 Normal Pulse-Width Modulation (NPWM)

NPWM uses single-slope PWM generation.

40.6.2.5.5 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 40-6. 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_TCC}), and can be calculated by the following equation:

$$f_{PWM\_SS} = \frac{f_{GCLK\_TCC}}{N(TOP+1)}$$

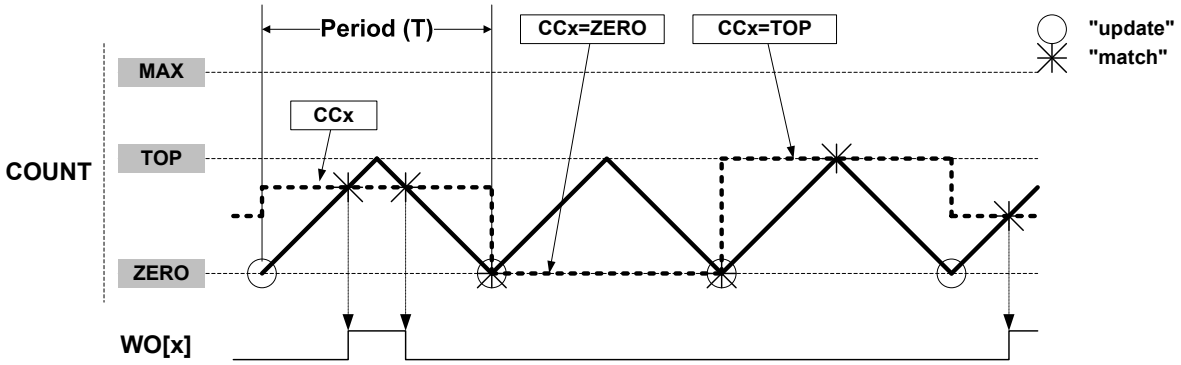
Where N represents the prescaler divider used (1, 2, 4, 8, 16, 64, 256, 1024).

40.6.2.5.6 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 synchronize the update condition on TOP.

Figure 40-7. 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}):

$$R_{PWM\_DS} = \frac{\log(TOP+1)}{\log(2)}.$$

The PWM frequency f_{PWM_DS} depends on the period setting (TOP) and the peripheral clock frequency f_{GCLK_TCC}, and can be calculated by the following equation (outside of DSBOTH mode):

$$f_{PWM\_DS} = \frac{f_{GCLK\_TCC}}{2N \cdot TOP}$$

N represents the prescaler divider used. The waveform generated will have a maximum frequency of half of the TCC clock frequency (f_{GCLK_TCC}) 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_TCC}), and can be calculated by the following equation:

$$P_{\mathrm{PWM\_DS}} = \frac{2N \cdot (\mathrm{TOP} - \mathrm{CCx})}{f_{\mathrm{GCLK\_TCC}}}$$

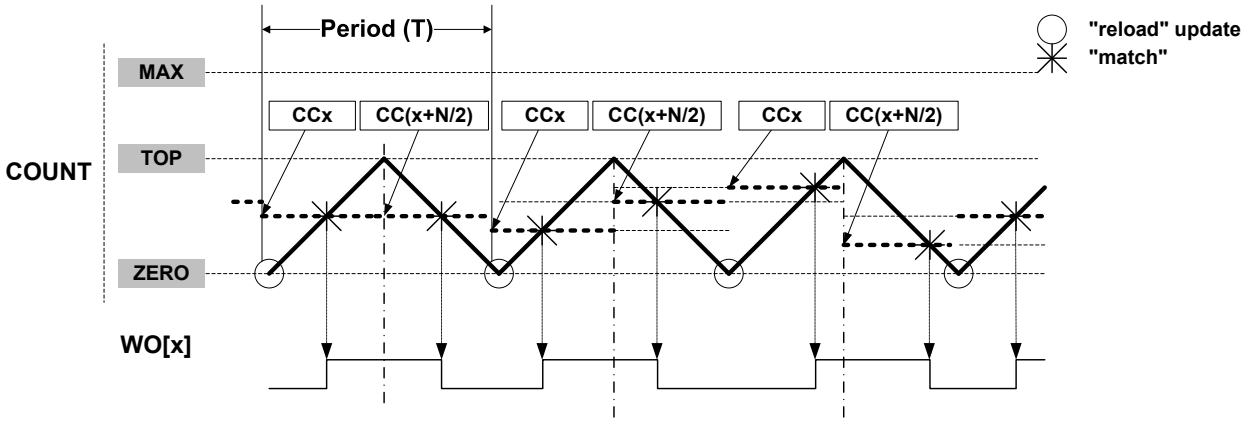
N represents the prescaler divider used.

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.)

40.6.2.5.7 Dual-Slope Critical PWM Generation

Critical mode generation allows generation of non-aligned centered pulses. In this mode, the period time TOP 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.

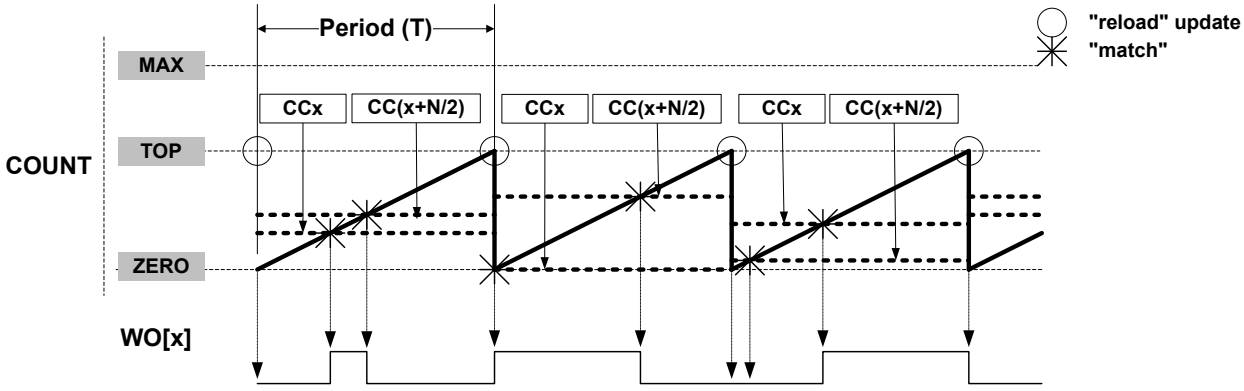
Figure 40-8. Dual-Slope Critical Pulse Width Modulation (N=CC_NUM)



40.6.2.5.8 Dual-Compare PWM Generation

Dual compare PWM generation allows generation of pulses unaligned on start or end of Period. In this mode, the period time is controlled by PER, while CCx controls the waveform output leading edge and CC(x+CC_NUM/2) controls the waveform output trailing edge.

Figure 40-9. Dual-Compare Pulse Width Modulation (N=CC_NUM)



40.6.2.5.9 Output Polarity

The polarity (WAVE.POLx) is available in all waveform output generation.

The table below shows the waveform output set/clear conditions, depending on the settings of timer/counter, direction, and polarity.

Table 40-5. 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 CCx	Timer/counter matches TOP
	1	0	Timer/counter matches CCx	Timer/counter matches ZERO
		1	Timer/counter matches ZERO	Timer/counter matches CCx
Dual-Slope PWM	x	0	Timer/counter matches CCx when counting up	Timer/counter matches CCx when counting down
		1	Timer/counter matches CCx when counting down	Timer/counter matches CCx when counting up
Dual Compare PWM	0	0	Timer/Counter match TOP Timer/ counter matches CC[x+WO_NUM/2]	Timer/counter matches CCx
		1	Timer/counter matches CCx	Timer/Counter match TOP Timer/ counter matches CC[x+WO_NUM/2]
	1	0	Timer/counter matches CCx	Timer/Counter match ZERO Timer/ counter matches CC[x+WO_NUM/2]
		1	Timer/Counter match ZERO Timer/ counter matches CC[x+WO_NUM/2]	Timer/counter matches CCx

In Normal and Match Frequency, the WAVE.POLx value represents the initial state of the waveform output.

40.6.2.6 Double Buffering

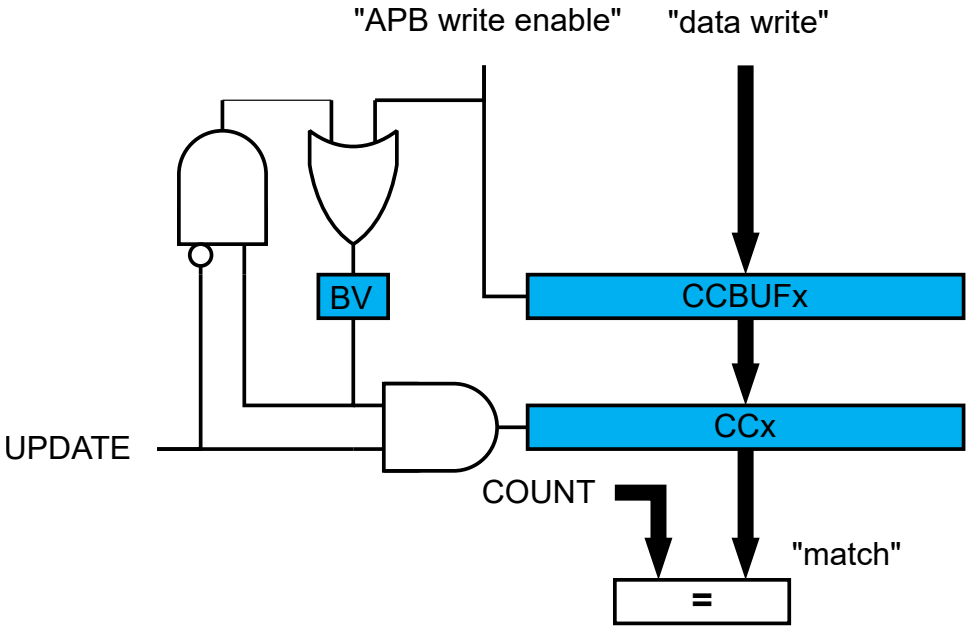
The Pattern (PATT), Period (PER) and Compare Channels (CCx) registers are all double buffered. Each buffer register has a buffer valid (PATTBUFV), PERBUFV) and CCBUFVx) bit in the STATUS register, which indicates that the Buffer register contains a valid value that can be copied into the corresponding register. As long as the respective Buffer Valid Status flag (PATTBUFV, PERBUFV or CCBUFVx) are set to '1', the related SYNCBUSY bits are set (SYNCBUSY.PATT, SYNCBUSY.PER or SYNCBUSY.CCx), a write to the respective PATT/PATTBUF, PER/PERBUF or CCx/CCBUFx registers will generate a PAC error, and read access to the respective PATT, 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: Software update command (CTRLBSET.CMD=0x3) act independently of LUPD value.

A compare register is double buffered as in the following figure.

Figure 40-10. Compare Channel Double Buffering



Both the registers (PATT/PER/CCx) and corresponding Buffer registers (PATTBUF/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 CTRLSET.LUPD.

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, whatever double buffering enabling is.

Figure 40-11. Unbuffered Single-Slope Up-Counting Operation

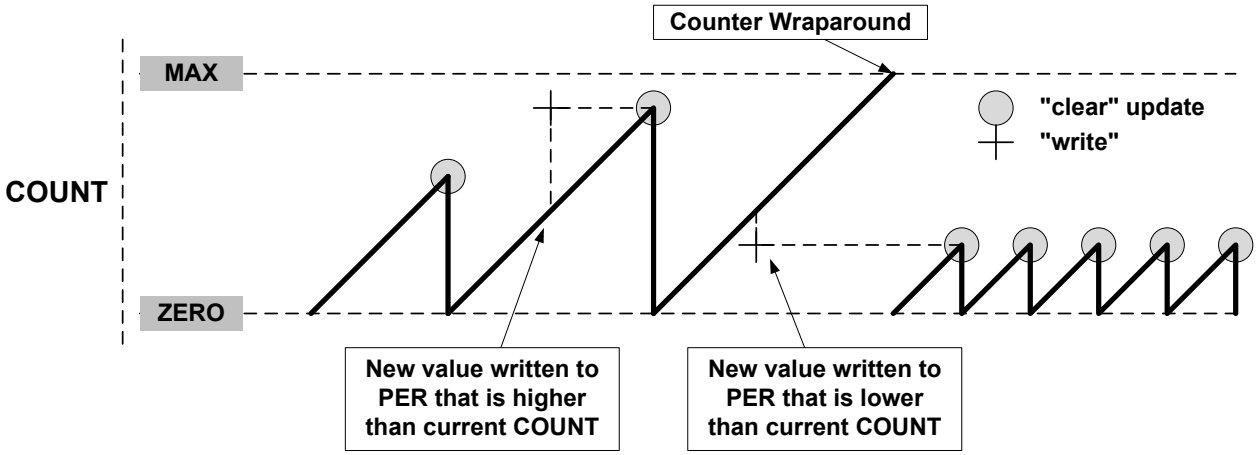
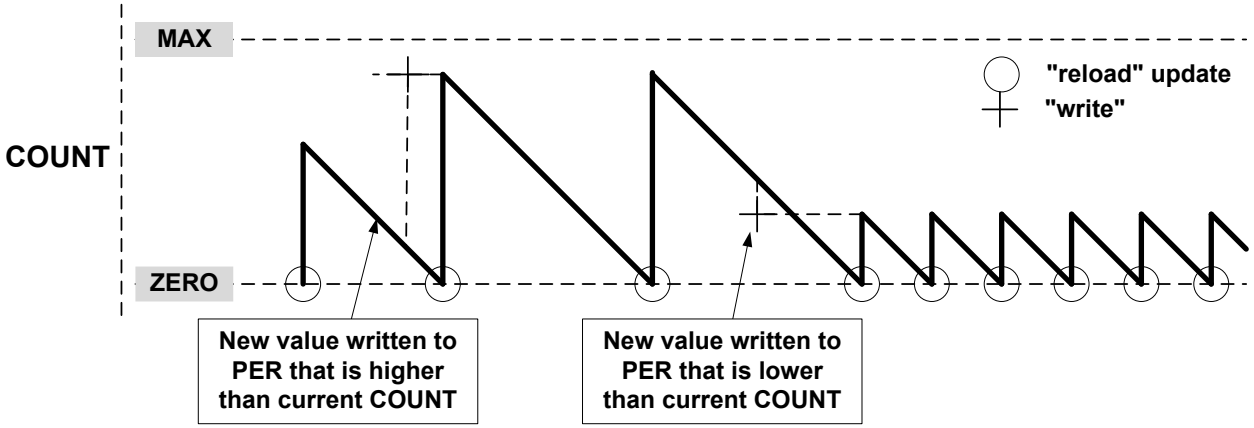
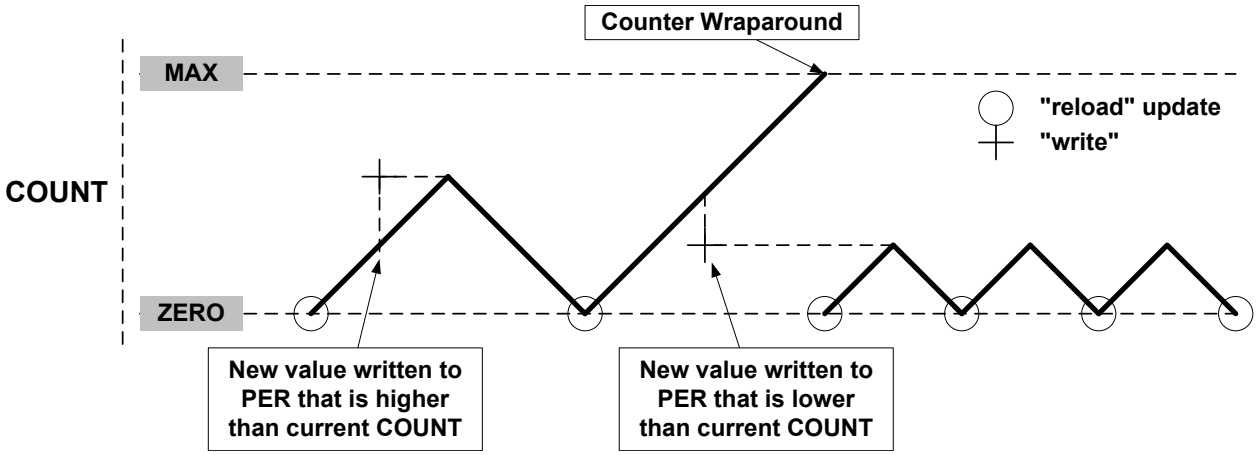


Figure 40-12. Unbuffered Single-Slope Down-Counting Operation



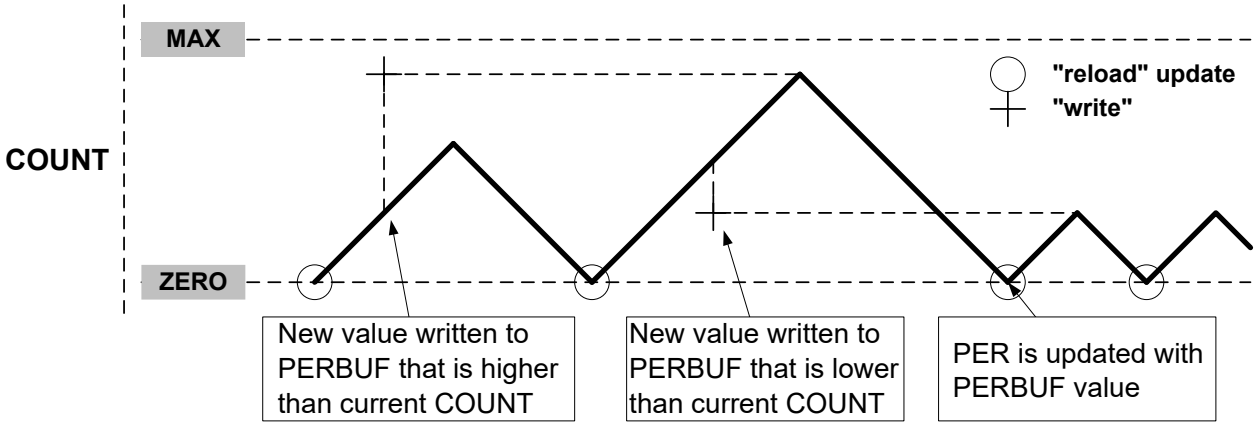
A counter wraparound can occur in any operation mode when up-counting without buffering, see Figure 40-11. COUNT and TOP are continuously compared, so when a new value that is lower than the current COUNT is written to TOP, COUNT will wrap before a compare match.

Figure 40-13. 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 shown in Figure 40-14. This prevents wraparound and the generation of odd waveforms.

Figure 40-14. Changing the Period Using Buffering



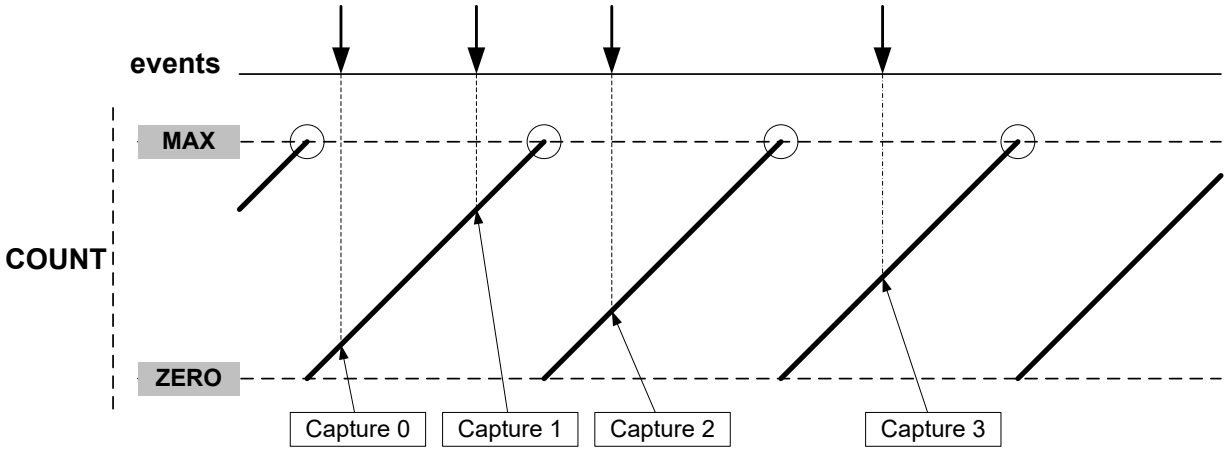
40.6.2.7 Capture Operations

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

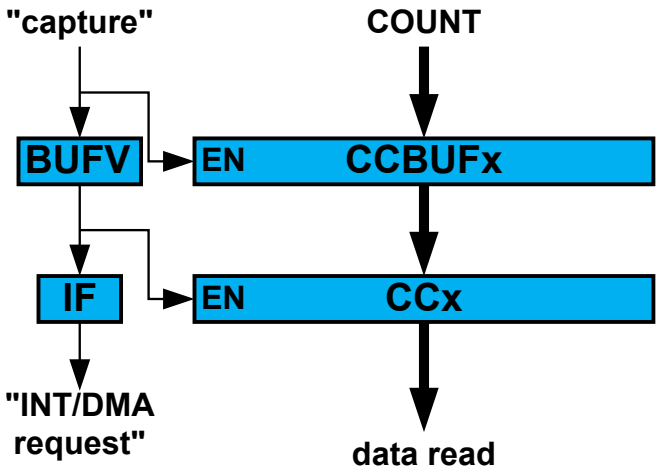
The compare/capture channels can be used as input capture channels to capture events from the Event System, and give them a timestamp. The following figure shows four capture events for one capture channel.

Figure 40-15. 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. CCBUFx register value can't be read, all captured data must be read from CCx register.

Figure 40-16. Capture Double Buffering



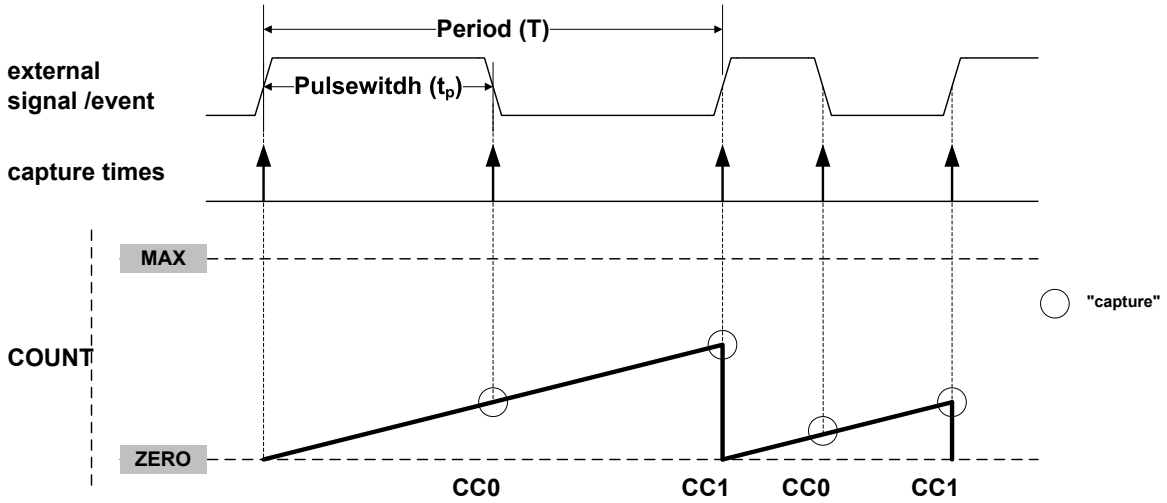
The TCC can detect capture overflow of the input capture channels: When a new capture event is detected while the Capture Buffer Valid flag (STATUS.CCBUFV) is still set, the new timestamp will not be stored and INTFLAG.ERR will be set.

Pulse-Width and Period (PWP) and Period and Pulse-Width (PPW) Capture Actions

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:

$$f = \frac{1}{T} \quad , \quad dutyCycle = \frac{t_p}{T}$$

Figure 40-17. 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 PPW (period and pulse-width) event action, period *T* will be 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* will be captured into CC1 and t_p into CC0.

The Timer/Counter Event x Invert Enable bit in Event Control register (EVCTRL.TCEINVx) is used for event source x to select whether the wraparound should 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 will be ignored and the channel will be enabled in compare mode of operation. When only one of these channel 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 will be set.

Note: When up-counting (CTRLBSET.DIR=0), counter values lower than 1 cannot be captured especially 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).

Note: 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 on which the capture has been done. For rising ramps CCx[MSB] is zero, for falling ramps CCx[MSB]=1.

40.6.3 Additional Features

40.6.3.1 One-Shot Operation

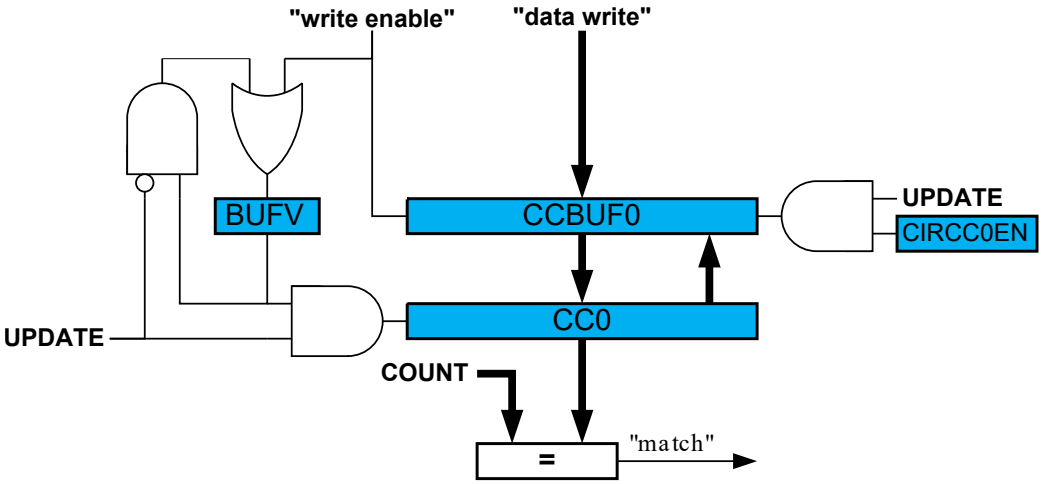
When one-shot is enabled, the counter automatically stops on the next Counter Overflow or Underflow condition. When the counter is stopped, the Stop bit in the Status register (STATUS.STOP) is set and the waveform outputs are set to the value defined by DRVCTRL.NREx and DRVCTRL.NRVx.

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 will count until an overflow or underflow occurs and stop counting. The one-shot operation can be restarted by a retrigger software command, a retrigger event or a start event. When the counter restarts its operation, STATUS.STOP is automatically cleared.

40.6.3.2 Circular Buffer

The Period register (PER) and the Compare Channels register (CC0 to CC3) 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 40-18. Circular Buffer on Channel 0



40.6.3.3 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 dithercy increment value and so 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:

$$PwmPeriod = \left(\frac{\mathrm{DITHERCY}}{16} + \mathrm{PER}\right)\left(\frac{1}{f_{\mathrm{GCLK\_TCC}}}\right)$$

Note: If DITH4 mode is enabled, the last 4 significant bits from PER/CCx register correspond to the DITHERCY value (the last 4 significant bits from COUNT are always read as 0), rest of the bits corresponds to PER/CCx or COUNT value.

DITH5 mode:

$$PwmPeriod = \left(\frac{\text{DITHERCY}}{32} + \text{PER}\right)\left(\frac{1}{f_{\text{GCLK\_TCC}}}\right)$$

DITH6 mode:

$$PwmPeriod = \left(\frac{\text{DITHERCY}}{64} + \text{PER}\right)\left(\frac{1}{f_{\text{GCLK\_TCC}}}\right)$$

Dithering on Pulse-Width

Writing DITHERCY in CCx will lead to an average PWM pulse width configured by the following formula.

DITH4 mode:

$$PwmPulseWidth = \left(\frac{\text{DITHERCY}}{16} + \text{CCx}\right)\left(\frac{1}{f_{\text{GCLK\_TCC}}}\right)$$

DITH5 mode:

$$PwmPulseWidth = \left(\frac{\text{DITHERCY}}{32} + \text{CCx}\right)\left(\frac{1}{f_{\text{GCLK\_TCC}}}\right)$$

DITH6 mode:

$$PwmPulseWidth = \left(\frac{\text{DITHERCY}}{64} + \text{CCx}\right)\left(\frac{1}{f_{\text{GCLK\_TCC}}}\right)$$

Note: The PWM period will remain static in this case.

40.6.3.4 Ramp Operations

Several ramp operation modes are supported. All of them require the timer/counter running in 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.

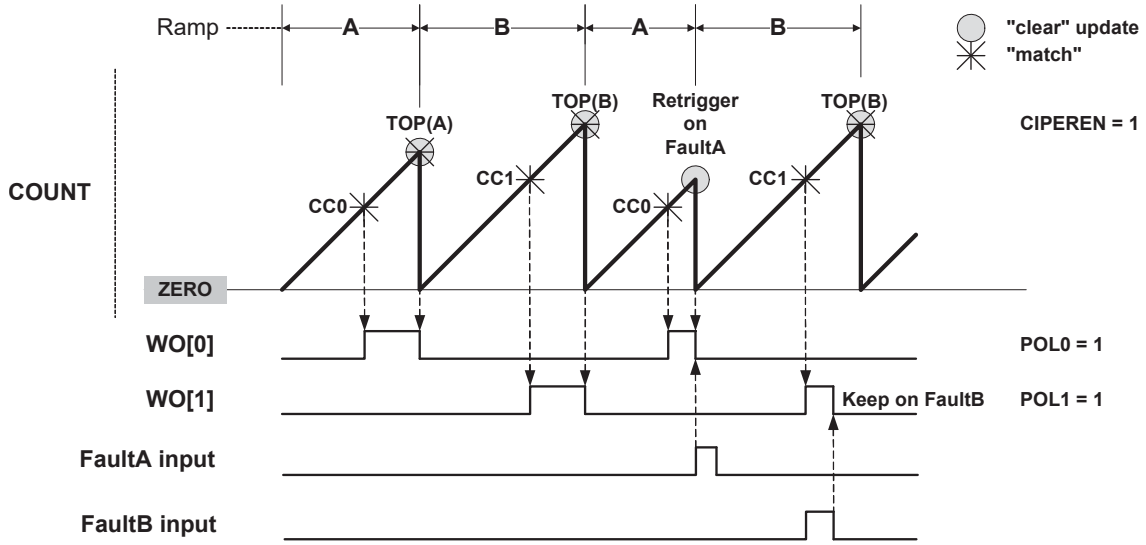
RAMP2 Operation

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 40-19. 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 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 outputs.

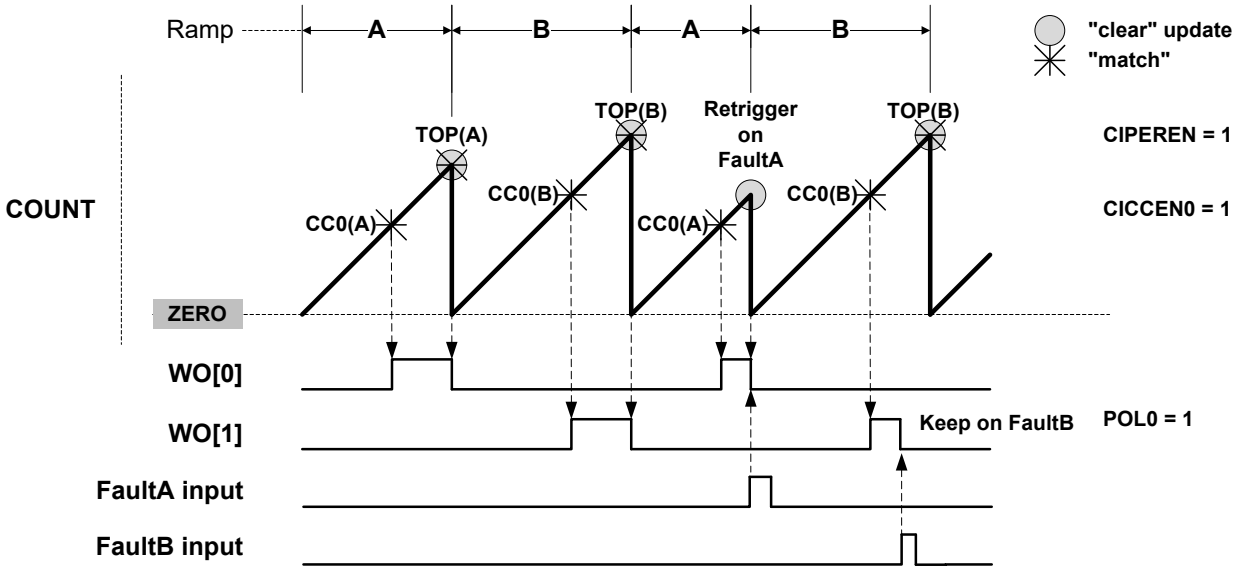
Figure 40-19. RAMP2 Standard Operation



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 40-20. 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 40-21. Critical RAMP2 Operation With More Than 2 Channels

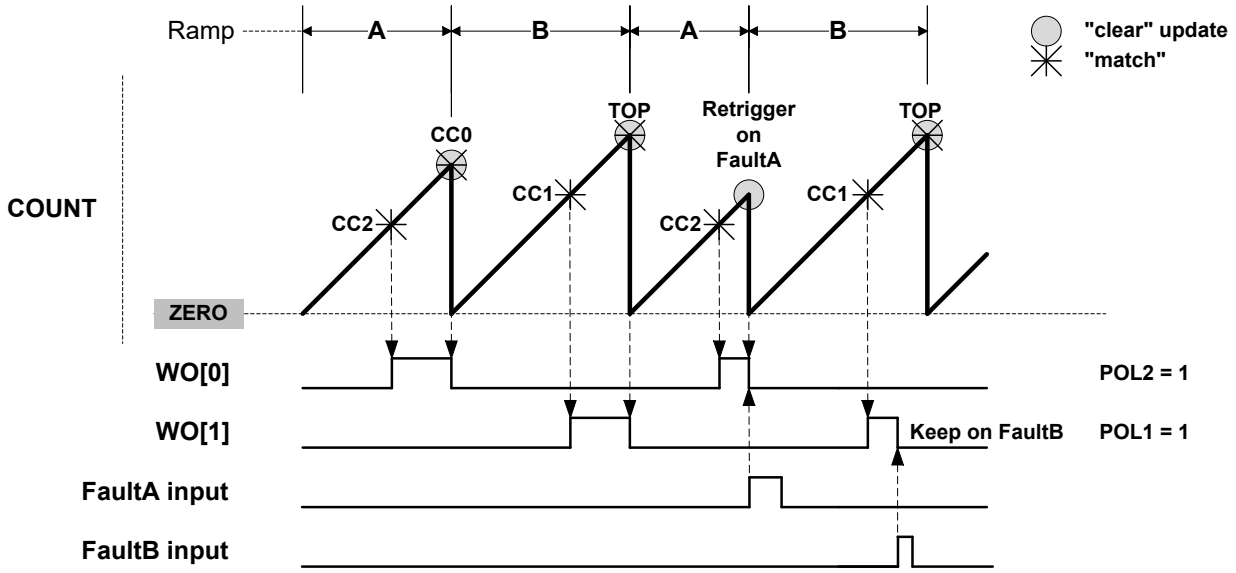
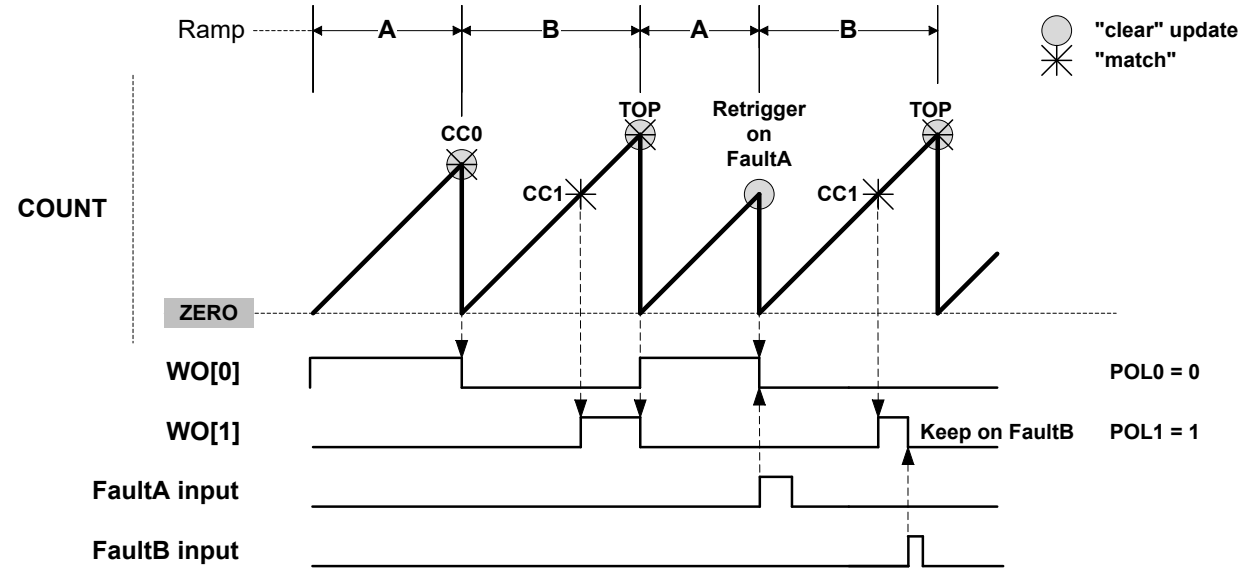


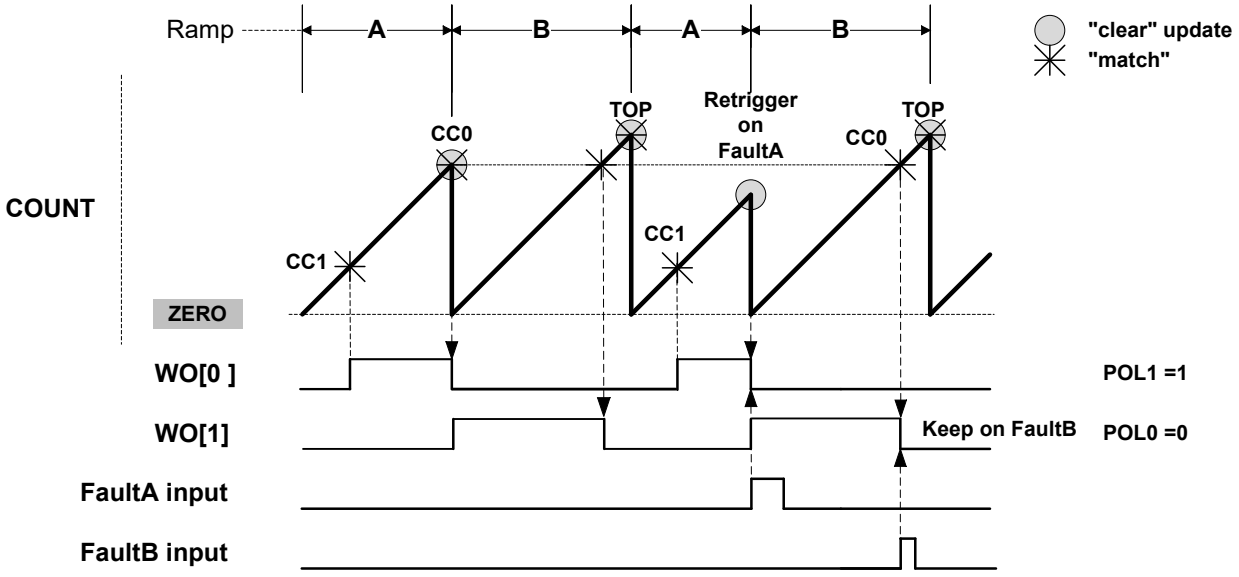
Figure 40-22. Critical RAMP2 Operation With 2 Channels



Critical Swapped RAMP2 (RAMP2CS) Operation

In RAMP2CS variant, WO[0] and W0[1] active ramps control is inverted: WO[0] active ramp is controlled by CC[1] POL1 and WO[1] active ramp is controlled by CC[0] POL0.

Figure 40-23. Critical Swapped RAMP2 Operation



40.6.3.5 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 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, also in device power modes where the clock is not available. To avoid false fault detection on external events (e.g. due to a glitch on an I/O port) a digital filter can be enabled and configured by the Fault Filter Value bits in the Fault n Configuration registers (FCTRLn.FILTERVAL). If the event width is less than FILTERVAL (in clock cycles), the event will be discarded. A valid event will be 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

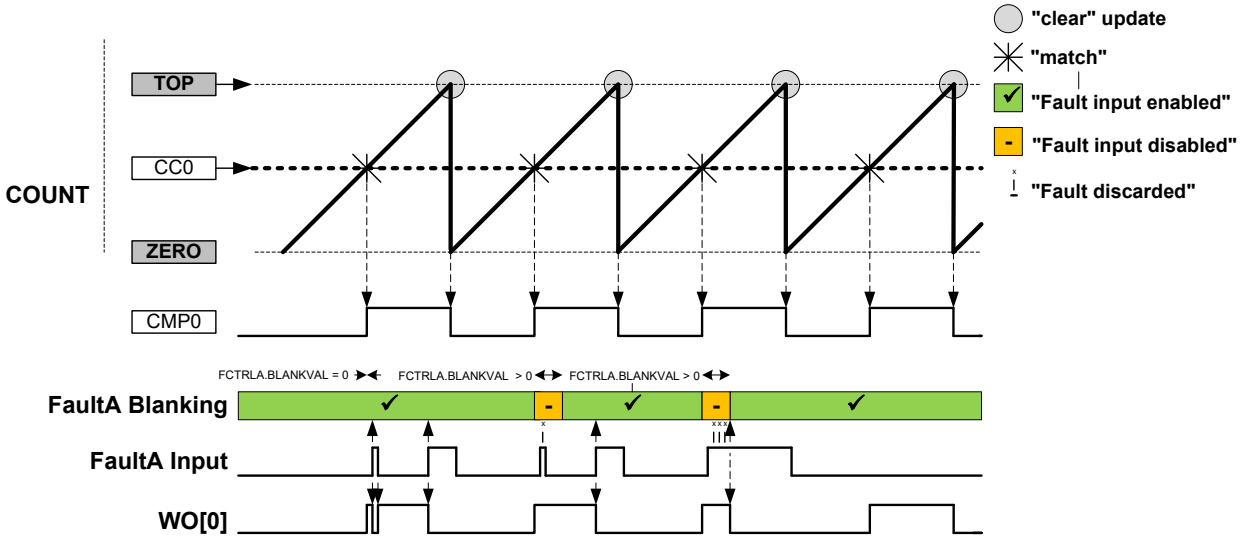
$$t_b = \frac{1 + \mathrm{BLANKVAL}}{f_{\mathrm{GCLK\_TCCx\_PRESC}}}$$

Here, $f_{GCLK\_TCCx\_PRESC}$ is the frequency of the prescaled peripheral clock frequency f_{GCLK_TCCx}.

The prescaler is enabled by writing '1' to the Fault n Blanking Prescaler bit (FCTRLn.BLANKPRESC). When disabled, $f_{GCLK\_TCCx\_PRESC}$=f_{GCLK_TCCx}. When enabled, $f_{GCLK\_TCCx\_PRESC}$=f_{GCLK_TCCx}/64.

The maximum blanking time (FCTRLn.BLANKVAL= 255) at f_{GCLK_TCCx}=96MHz is 2.67µs (no prescaler) or 170µs (prescaling). For f_{GCLK_TCCx}=1MHz, the maximum blanking time is either 170µs (no prescaling) or 10.9ms (prescaling enabled).

Figure 40-24. 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 shown in the figures below.

Figure 40-25. Fault Qualification in RAMP1 Operation

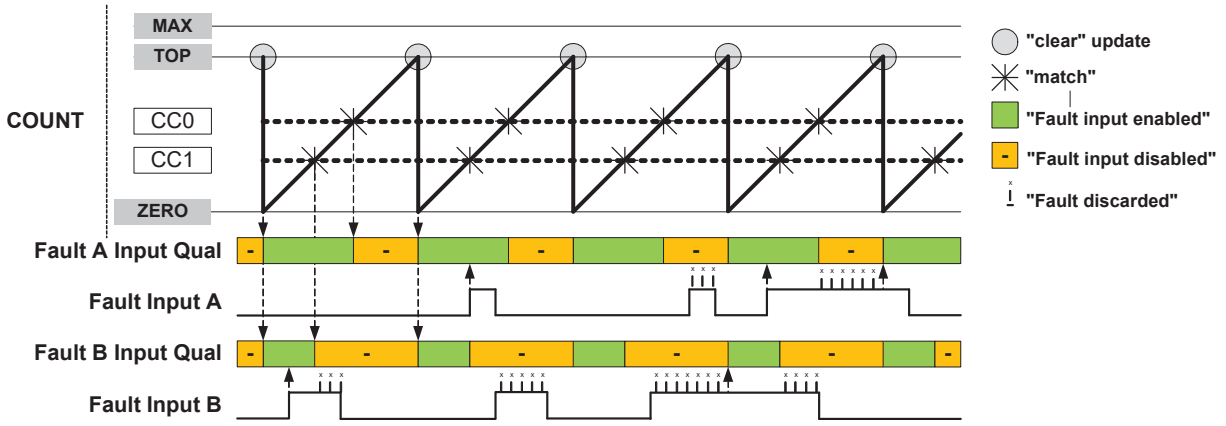
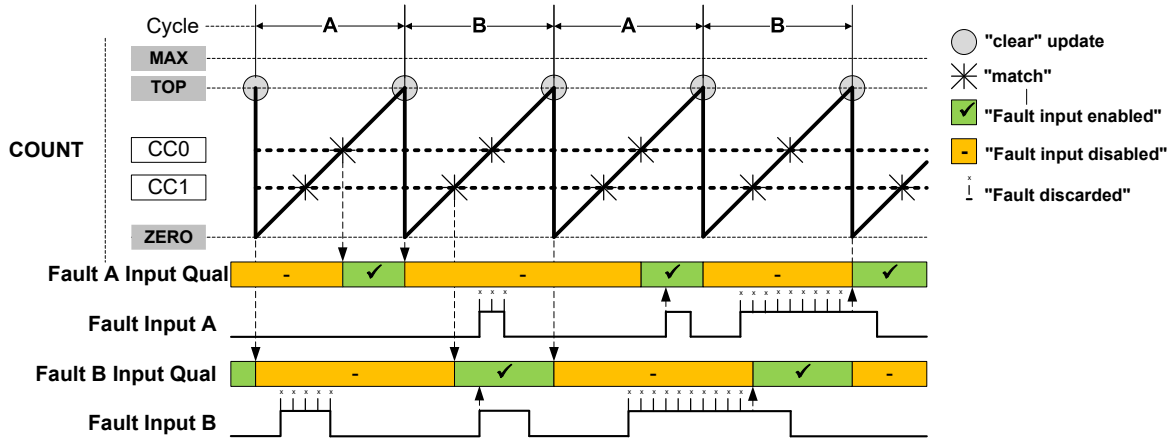


Figure 40-26. Fault Qualification in RAMP2 Operation

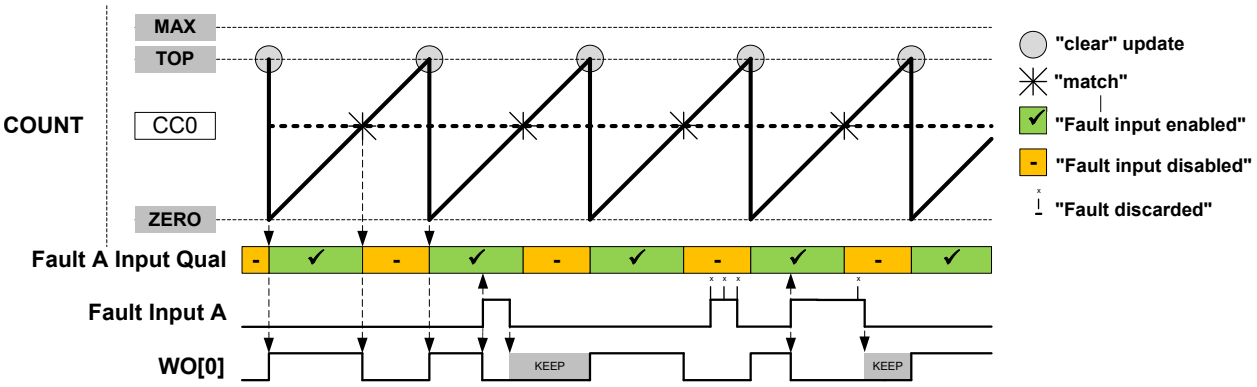


Fault Actions

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 will be clamped to zero as long as the fault condition is present. The clamp will be released on the start of the first cycle after the fault condition is no longer present, see next Figure.

Figure 40-27. 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 will be 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 40-28. In Ramp 1 mode, when the new cycle starts, the compare outputs will be clamped to inactive level as long as the fault condition is present.

Note: For RAMP2 operation, when a new timer/counter cycle starts the cycle index will change automatically, see Figure 40-29. Fault A and Fault B are qualified only during the cycle A and cycle B respectively: Fault A is disabled during cycle B, and Fault B is disabled during cycle A.

Figure 40-28. Waveform Generation in RAMP1 mode with Restart Action

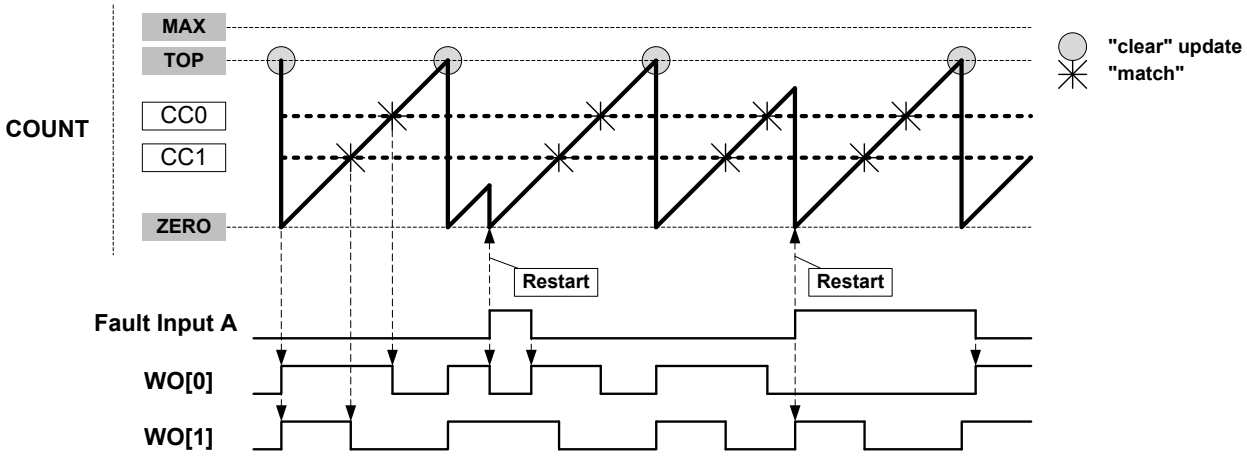
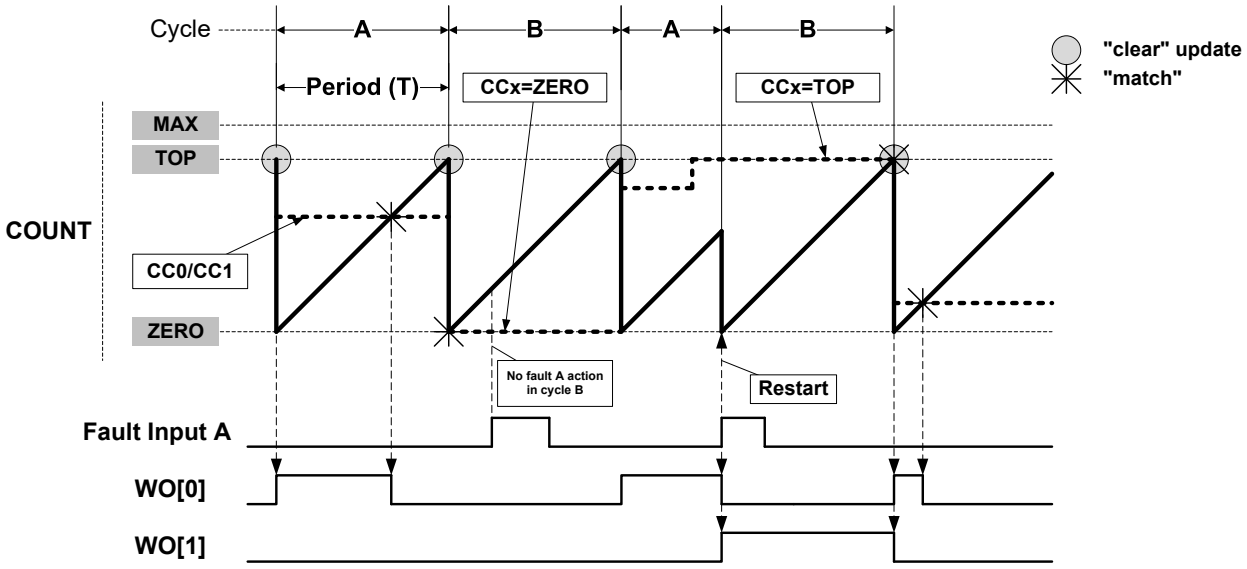


Figure 40-29. Waveform Generation in RAMP2 mode with Restart Action



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, for further details refer to 40.6.2.7. Capture Operations
- 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 40-30.
- 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 40-31.

CCx Content:

In CAPTMIN and CAPTMAX operations, CCx keeps the respective extremum captured values, see Figure 40-30. In LOCMIN, LOCMAX or DERIV0 operation, CCx follows the counter value at fault time, see Figure 40-31.

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) top (for CAPTMAX). If the CCx register initial value is zero (for CAPTMIN) top (for CAPTMAX), no captures will be performed using the corresponding channel.

MCx Behaviour:

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 interrupt flag is set when a new relative local Minimum (for CAPTMIN) or Maximum (for CAPTMAX) value has been 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 CAPTMIN and CAPTMAX operation, 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 interrupt flag is set when a new absolute local Minimum (for CAPTMIN) or Maximum (for CAPTMAX) value has been 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 40-30. Capture Action “CAPTMAX”

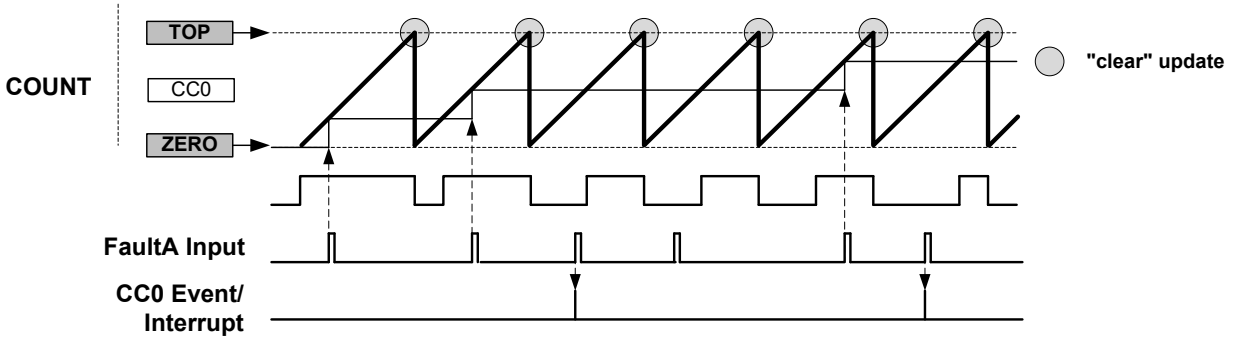
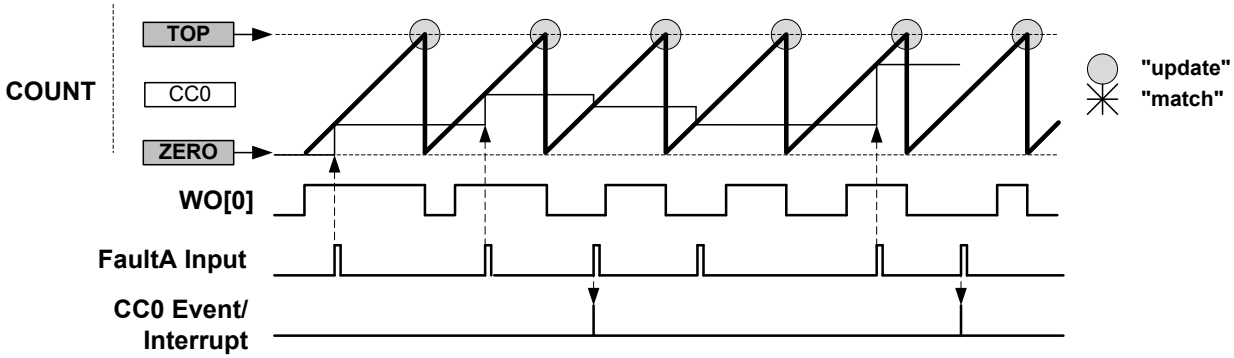


Figure 40-31. Capture Action “DERIV0”



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 next figure ('Waveform Generation with Halt and Restart Actions') shows 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 after that ('Waveform Generation with Fault Qualification, Halt, and Restart Actions') shows a similar example, but with additionally enabled fault qualification. Here, counting is resumed after the fault condition is no longer present.

Note that in RAMP2 and RAMP2A operations, when a new timer/counter cycle starts, the cycle index will automatically change.

Figure 40-32. Waveform Generation with Halt and Restart Actions

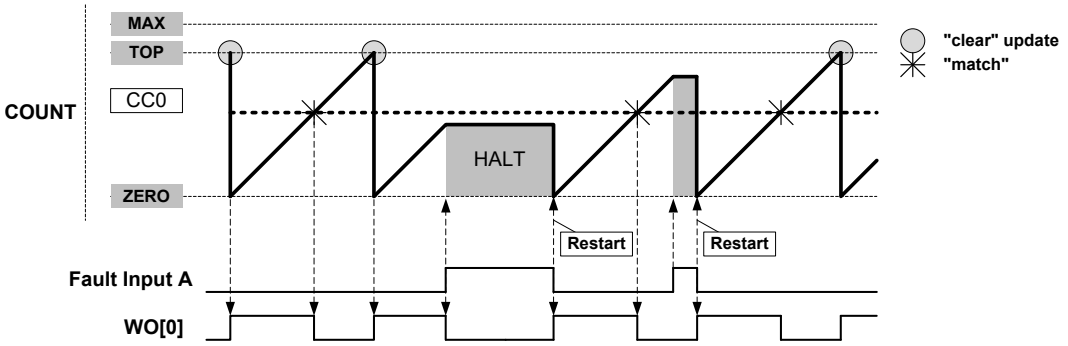
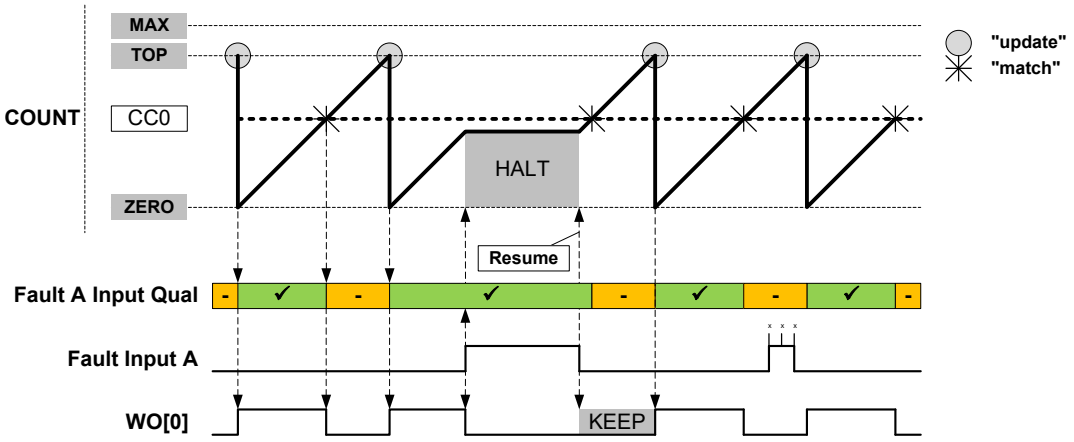


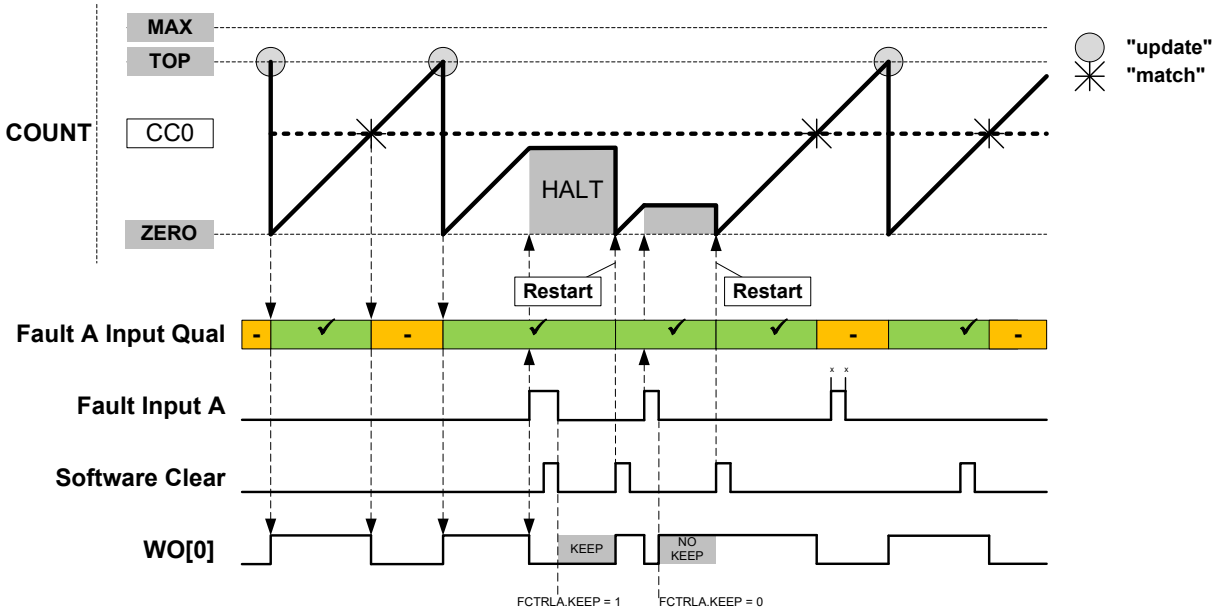
Figure 40-33. Waveform Generation with Fault Qualification, Halt, and Restart Actions



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 in order 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 40-34. Waveform Generation with Software Halt, Fault Qualification, Keep and Restart Actions



40.6.3.6 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 (e.g. 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.

40.6.3.7 Time-Stamp Capture

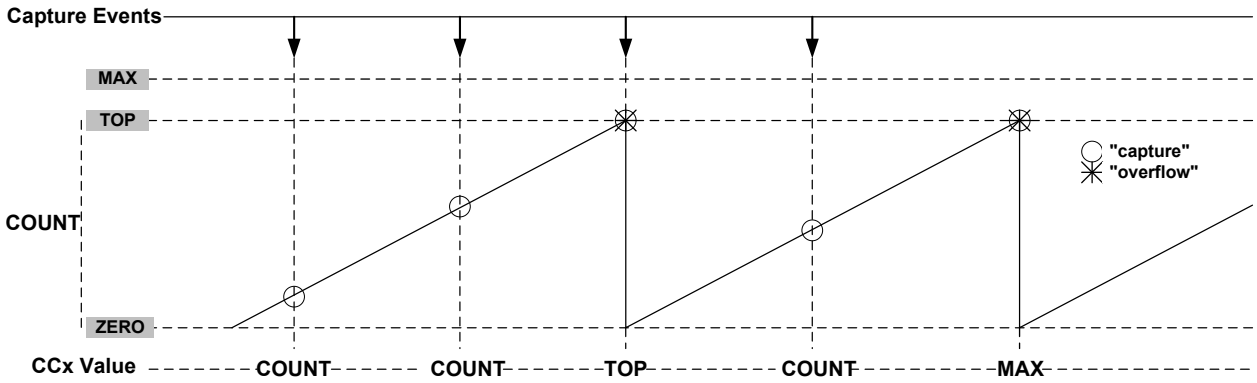
This feature is enabled when the Capture Time Stamp (STAMP) Event Action in Event Control register (EVCTRL.EVACT) is selected. The counter TOP value must be smaller than MAX.

When a capture event 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 40-35. Time-Stamp



40.6.3.8 Waveform Extension

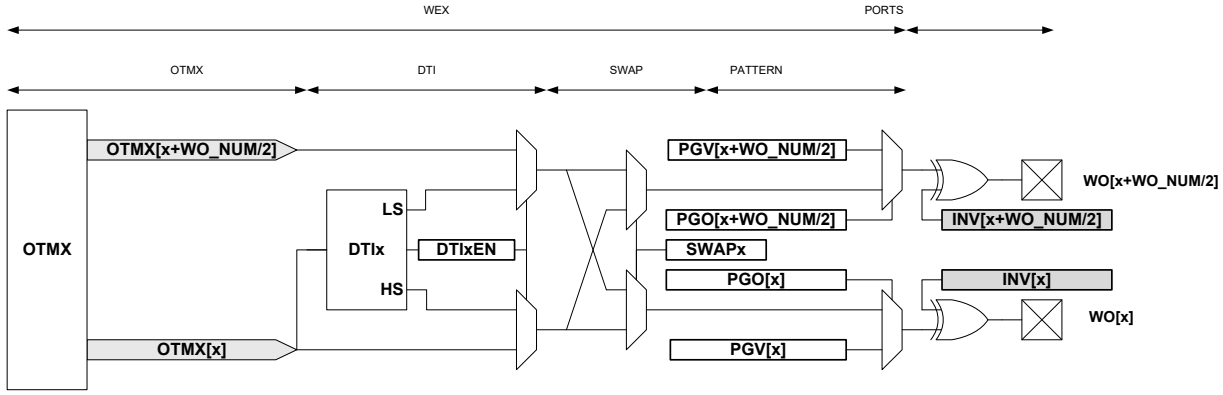
Figure 40-36 shows a schematic diagram of actions of the four optional units that follow the recoverable fault stage on a port pin pair: Output Matrix (OTMX), Dead-Time Insertion (DTI), SWAP and Pattern Generation. The DTI and SWAP units can be seen as a four port pair slices:

- 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 generally:

- Slice n DTIx/SWAPx acting on port pins (WO[x], WO[WO_NUM/2 +x])

Figure 40-36. Waveform Extension Stage Details



The output matrix (OTMX) unit distributes compare channels according to the selectable WEXCTRL.OTMX configuration:

- 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, then 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 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 tables below provide the different configurations depending the TCC instance.

Table 40-6. Output Matrix Channel Pin Routing Configuration (TCC0 and TCC3)

WEXCTRL.OTMX	OTMX[7]	OTMX[6]	OTMX[5]	OTMX[4]	OTMX[3]	OTMX[2]	OTMX[1]	OTMX[0]
0x0	CC3	CC2	CC1	CC0	CC3	CC2	CC1	CC0
0x1	CC1	CC0	CC1	CC0	CC1	CC0	CC1	CC0
0x2	CC0	CC0	CC0	CC0	CC0	CC0	CC0	CC0
0x3	CC1	CC1	CC1	CC1	CC1	CC1	CC1	CC0

Table 40-7. Output Matrix Channel Pin Routing Configuration (TCC2)

WEXCTRL.OTMX	OTMX[1]	OTMX[0]
-	CC1	CC0

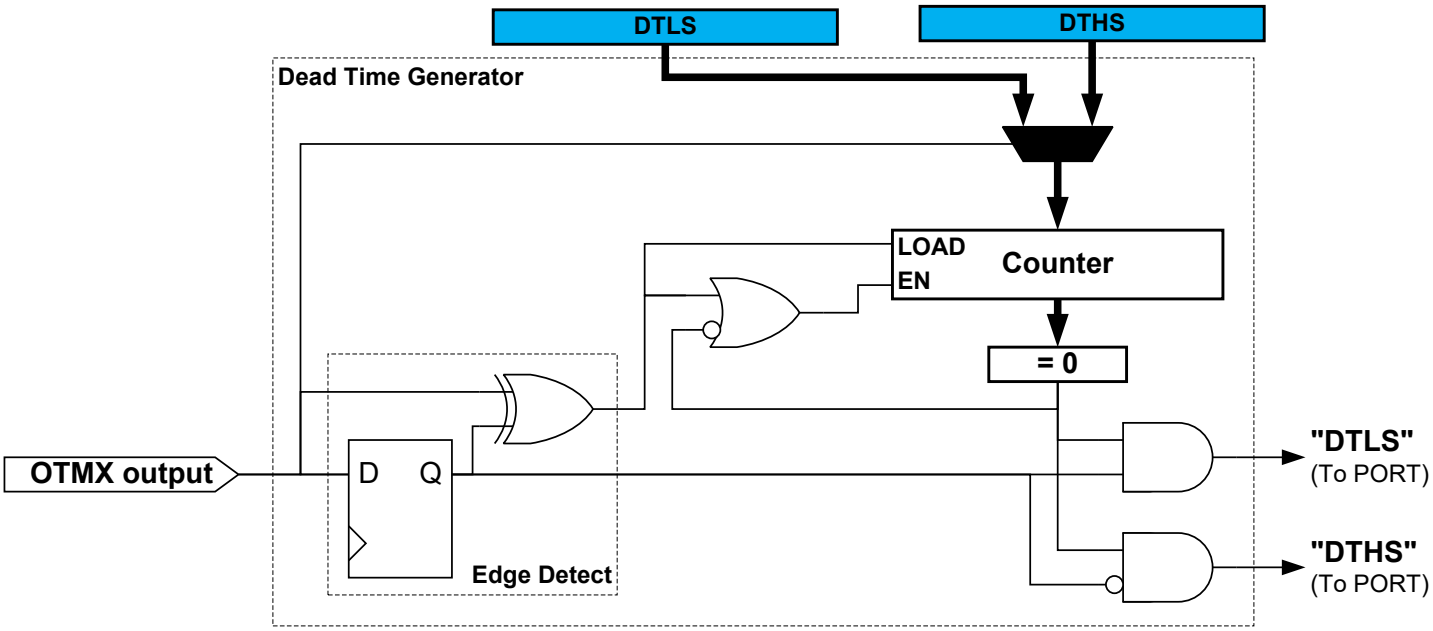
Table 40-8. Output Matrix Channel Pin Routing Configuration (TCC1)

WEXCTRL.OTMX	OTMX[3]	OTMX[2]	OTMX[1]	OTMX[0]
-	CC1	CC0	CC1	CC0

The dead-time insertion (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 be active simultaneously and active states of LS and HS are separated by a security time off.

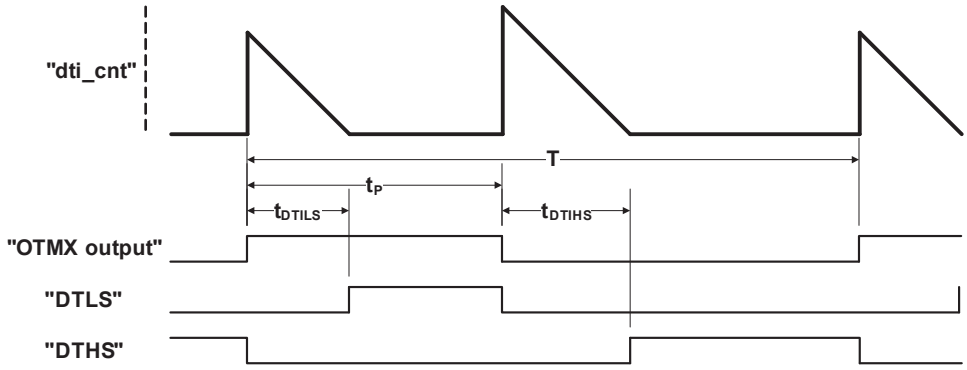
The DTI stage consists of four equal dead-time insertion generators; one for each of the first four compare channels. Figure 40-37 shows the block diagram of one DTI generator. The four channels have a common register which controls the dead time, which is independent of high side and low side setting.

Figure 40-37. Dead-Time Generator Block Diagram



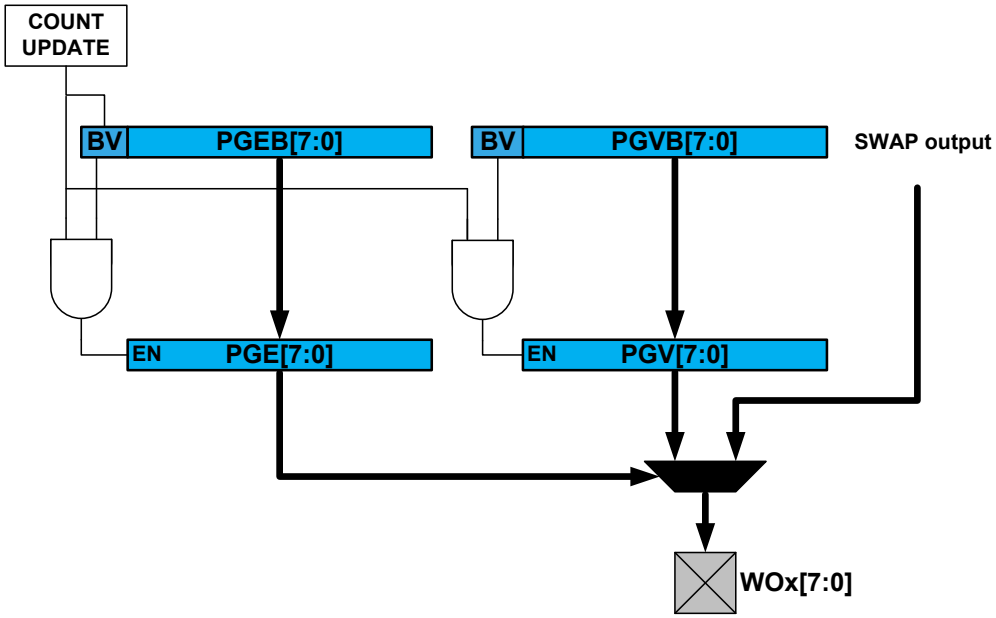
As shown in Figure 40-38, the 8-bit dead-time counter is decremented by one for each peripheral clock cycle until it reaches zero. A non-zero counter value will force both the low side and high side outputs into their OFF state. When the output matrix (OTMX) output changes, the dead-time counter is reloaded 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 40-38. 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. See also Figure 40-39.

Figure 40-39. 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 directly the PATT.PGE, PATT.PGV bits registers.

40.6.4 Host/Client Operation

Two TCC (TCC0 and TCC1) instances sharing the same GCLK_TCC 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 (TCC1). When the bit is set, the client TCC instance will synchronize the CC channels to the Host counter (TCC0).

40.6.5 DMA, Interrupts, and Events

Table 40-9. 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

.....continued					
Condition	**Interrupt request**	**Event output**	**Event input**	**DMA request**	**DMA request is cleared**
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			
Cycle	Yes	Yes			
Capture Overflow Error	Yes				
Debug Fault State	Yes				
Recoverable Faults	Yes				
Non-Recoverable Faults	Yes				
TCCx Event 0 input			Yes(4)		
TCCx Event 1 input			Yes(5)		

Notes:

1. DMA request set on Overflow, Underflow or Retrigger 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:
   - retrigger 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
5. On event input, either action can be executed:
   - retrigger counter
   - control counter direction
   - stop the counter
   - decrement the counter
   - perform period and pulse width capture
   - generate non-recoverable fault

40.6.5.1 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 Retrigger) is detected.
When an update condition (Overflow, Underflow or Retrigger) 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 once 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 buffer are intended to be used with RAMP2, RAMP2A and DSBOTH operation only.

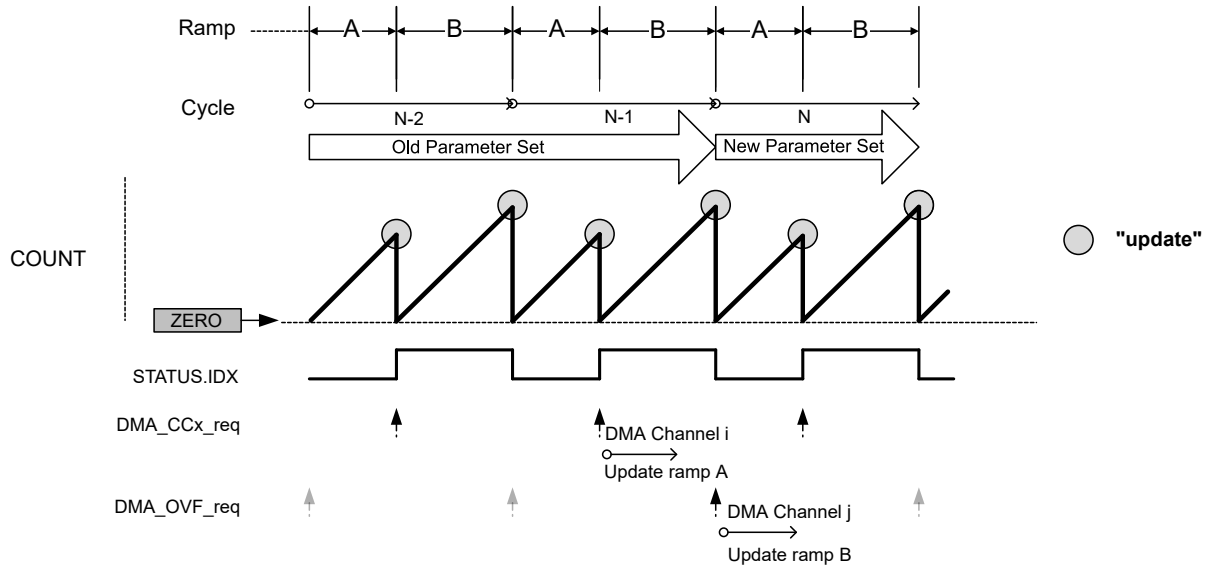
DMA Operation with Circular Buffer in RAMP2 and RAMP2A Mode

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 a 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 40-40. DMA Triggers in RAMP and RAMP2 Operation Mode and Circular Buffer Enabled



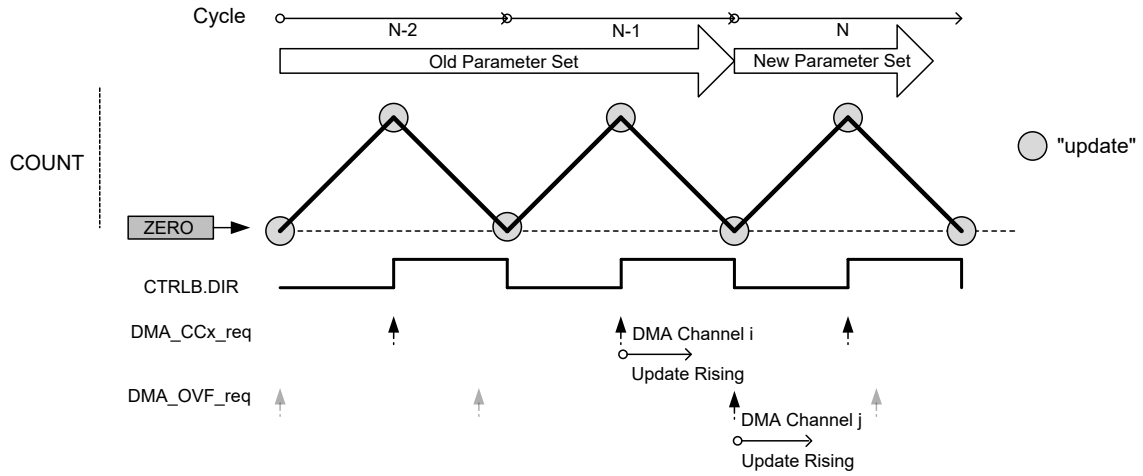
DMA Operation with Circular Buffer in DSBOTH Mode

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 up-counting phase with an effective DMA transfer on previous down-counting phase (DMA acknowledge).

When up-counting, all circular buffer values can be updated through a DMA channel triggered by MC trigger. When down-counting, all circular buffer values can be updated through a second DMA channel, triggered by the OVF DMA request.

Figure 40-41. DMA Triggers in DSBOTH Operation Mode and Circular Buffer Enabled



40.6.5.2 Interrupts

The TCC has the following interrupt sources:

- Overflow/Underflow (OVF)
- Retrigger (TRG)
- Count (CNT) - refer also to description of EVCTRL.CNTSEL.
- Capture Overflow Error (ERR)
- Debug Fault State (DFS)
- Update Fault State (UFS)
- Recoverable Faults (FAULTn)
- Non-recoverable Faults (FAULTx)
- Compare Match or Capture Channels (MCx)

These interrupts are asynchronous wake-up sources. See Sleep Mode Entry and Exit Table in PM/Sleep Mode Controller section for details.

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 TCC is reset. See 40.7.13. INTFLAG 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.

Note: Interrupts must be globally enabled for interrupt requests to be generated. Refer to Nested Vector Interrupt Controller for details.

40.6.5.3 Events

The TCC can generate the following output events:

- Overflow/Underflow (OVF)
- Trigger (TRG)
- Counter (CNT) For further details, refer to 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. Refer to EVSYS – Event System.

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 retrigger
- 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 retrigger
- 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 on 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.
- Capture overflow times (Max value)
- Non-recoverable fault

The counter Event Actions are available in the Event Control registers (EVCTRL.EVACT0 and EVCTRL.EVACT1). For further details, refer to EVCTRL.

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. Refer to EVSYS – Event System for details on how to configure the event system.

40.6.6 Sleep Mode Operation

The TCC can be configured to operate in any Sleep mode. To be able to run in standby the RUNSTDBY bit in the Control A register (CTRLA.RUNSTDBY) must be '1'. The MODULE can in any Sleep mode wake-up the device using interrupts or perform actions through the Event System.

40.6.7 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. Refer to the Debug Control (40.7.9. DBGCTRL) register for details.

40.6.8 Synchronization

Some registers (or bit fields within a register) require synchronization when read and/or written.

Synchronization is denoted by the "Read-Synchronized" (or "Read-Synchronized Bits”) and/or "Write-Synchronized" (or "Write-Synchronized Bits”) property in each individual register description.

For more details, refer to Register Synchronization.

40.7 Register Summary

Refer to the Registers Description section for more details on register properties and access permissions.

Important:
The peripheral register map is given as an example for CTRLA.RESOLUTION = 0x2 (DITH5) case.

CTRLA.RESOLUTION bit field selection affects the following registers' bit fields:

- COUNT
- PER
- CCx
- PERBUF
- CCBUFx

Refer to each register for more information.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRLA	31:24					CPTEN3	CPTEN2	CPTEN1	CPTEN0
		23:16	DMAOS							FCYCLE
		15:8	MSYNC	ALOCK	PRESCSYNC[1:0]		RUNSTDBY	PRESCALER[2:0]		
		7:0		RESOLUTION[1:0]					ENABLE	SWRST
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	31:24								
		23:16								
		15:8					CC3	CC2	CC1	CC0
		7:0	PER	WAVE	PATT	COUNT	STATUS	CTRLB	ENABLE	SWRST
0x0C	FCTRLA	31:24					FILTERVAL[3:0]			
		23:16	BLANKVAL[7:0]							
		15:8	BLANKPRESC	CAPTURE[2:0]			CHSEL[1:0]		HALT[1:0]	
		7:0	RESTART	BLANK[1:0]		QUAL	KEEP		SRC[1:0]	
0x10	FCTRLB	31:24					FILTERVAL[3:0]			
		23:16	BLANKVAL[7:0]							
		15:8	BLANKPRESC	CAPTURE[2:0]			CHSEL[1:0]		HALT[1:0]	
		7:0	RESTART	BLANK[1:0]		QUAL	KEEP		SRC[1:0]	
0x14	WEXCTRL	31:24	DTHS[7:0]							
		23:16	DTLS[7:0]							
		15:8					DTIEN3	DTIEN2	DTIEN1	DTIEN0
		7:0							OTMX[1:0]	
0x18	DRVCTRL	31:24	FILTERVAL1[3:0]				FILTERVAL0[3:0]			
		23:16	INVEN7	INVEN6	INVEN5	INVEN4	INVEN3	INVEN2	INVEN1	INVEN0
		15:8	NRV7	NRV6	NRV5	NRV4	NRV3	NRV2	NRV1	NRV0
		7:0	NRE7	NRE6	NRE5	NRE4	NRE3	NRE2	NRE1	NRE0
0x1C ... 0x1D	Reserved									
0x1E	DBGCTRL	7:0						FDDBD		DBGRUN
0x1F	Reserved									
0x20	EVCTRL	31:24					MCEO3	MCEO2	MCEO1	MCEO0
		23:16					MCEI3	MCEI2	MCEI1	MCEI0
		15:8	TCEI1	TCEI0	TCINV1	TCINV0		CNTEO	TRGEO	OVFEO
		7:0	CNTSEL[1:0]		EVACT1[2:0]			EVACT0[2:0]		

.....continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x24	INTENCLR	31:24								
		23:16					MC3	MC2	MC1	MC0
		15:8	FAULT1	FAULT0	FAULTB	FAULTA	DFS	UFS		
		7:0					ERR	CNT	TRG	OVF
0x28	INTENSET	31:24								
		23:16					MC3	MC2	MC1	MC0
		15:8	FAULT1	FAULT0	FAULTB	FAULTA	DFS	UFS		
		7:0					ERR	CNT	TRG	OVF
0x2C	INTFLAG	31:24								
		23:16					MC3	MC2	MC1	MC0
		15:8	FAULT1	FAULT0	FAULTB	FAULTA	DFS	UFS		
		7:0					ERR	CNT	TRG	OVF
0x30	STATUS	31:24					CMP3	CMP2	CMP1	CMP0
		23:16					CCBUFV3	CCBUFV2	CCBUFV1	CCBUFV0
		15:8	FAULT1	FAULT0	FAULTB	FAULTA	FAULT1IN	FAULT0IN	FAULTBIN	FAULTAIN
		7:0	PERBUFV		PATTBUFV	SLAVE	DFS	UFS	IDX	STOP
0x34	COUNT	31:24								
		23:16	COUNT[23:16]							
		15:8	COUNT[15:8]							
		7:0	COUNT[7:0]							
0x38	PATT	15:8	PGV7	PGV6	PGV5	PGV4	PGV3	PGV2	PGV1	PGV0
		7:0	PGE7	PGE6	PGE5	PGE4	PGE3	PGE2	PGE1	PGE0
0x3A ... 0x3B	Reserved									
0x3C	WAVE	31:24					SWAP3	SWAP2	SWAP1	SWAP0
		23:16					POL3	POL2	POL1	POL0
		15:8					CICCEN3	CICCEN2	CICCEN1	CICCEN0
		7:0	CIPEREN		RAMP[2:0]			WAVEGEN[2:0]		
0x40	PER	31:24								
		23:16	PER[17:10]							
		15:8	PER[9:2]							
		7:0	PER[1:0]		DITHER[5:0]					
0x44	CC0	31:24								
		23:16	CC[17:10]							
		15:8	CC[9:2]							
		7:0	CC[1:0]		DITHER[5:0]					
0x48	CC1	31:24								
		23:16	CC[17:10]							
		15:8	CC[9:2]							
		7:0	CC[1:0]		DITHER[5:0]					
0x4C	CC2	31:24								
		23:16	CC[17:10]							
		15:8	CC[9:2]							
		7:0	CC[1:0]		DITHER[5:0]					
0x50	CC3	31:24								
		23:16	CC[17:10]							
		15:8	CC[9:2]							
		7:0	CC[1:0]		DITHER[5:0]					
0x54 ... 0x63	Reserved									
0x64	PATTBUF	15:8	PGVB7	PGVB6	PGVB5	PGVB4	PGVB3	PGVB2	PGVB1	PGVB0
		7:0	PGEB7	PGEB6	PGEB5	PGEB4	PGEB3	PGEB2	PGEB1	PGEB0
0x66 ... 0x6B	Reserved									

.....continued

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x6C	PERBUF	31:24								
		23:16	PERBUF[17:10]							
		15:8	PERBUF[9:2]							
		7:0	PERBUF[1:0]		DITHERBUF[5:0]					
0x70	CCBUF0	31:24								
		23:16	CCBUF[17:10]							
		15:8	CCBUF[9:2]							
		7:0	CCBUF[1:0]		DITHERBUF[5:0]					
0x74	CCBUF1	31:24								
		23:16	CCBUF[17:10]							
		15:8	CCBUF[9:2]							
		7:0	CCBUF[1:0]		DITHERBUF[5:0]					
0x78	CCBUF2	31:24								
		23:16	CCBUF[17:10]							
		15:8	CCBUF[9:2]							
		7:0	CCBUF[1:0]		DITHERBUF[5:0]					
0x7C	CCBUF3	31:24								
		23:16	CCBUF[17:10]							
		15:8	CCBUF[9:2]							
		7:0	CCBUF[1:0]		DITHERBUF[5:0]					

40.7.1 Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected Bits, Write-Synchronized Bits

Bit	31	30	29	28	27	26	25	24
					CPTEN3	CPTEN2	CPTEN1	CPTEN0
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	23	22	21	20	19	18	17	16
	DMAOS							FCYCLE
Access	R/W							R/W
Reset	0							0

Bit	15	14	13	12	11	10	9	8
	MSYNC	ALOCK	PRESCSYNC[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 – 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.
Note: This bit is enable-protected. This bit is not synchronized.

Bit 23 – DMAOS DMA One-Shot Trigger Mode
This bit enables the DMA One-shot Trigger Mode.
Writing a '1' to this bit will generate a DMA trigger on TCC cycle following a TCC_CTRLBSET_CMD_DMAOS command.
Writing a '0' to this bit will generate DMA triggers on each TCC cycle.
Note: This bit is enable-protected. This bit is not synchronized.

Bit 16 – FCYCLE Full Cycle Enable
When this bit is set, TCC will wait for the end of the current cycle, to evaluate the stop condition.
Note: This bit is enable-protected. This bit is not synchronized.

Value	Description
0	The stop condition is evaluated immediately.
1	The stop condition is evaluated at the end of the cycle.

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.
Note: This bit is enable-protected. 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

Note: This bit is enable-protected. 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 retrigger events
1	CTRLB.LUPD is set to '1' on each overflow/underflow or retrigger event.

Bits 13:12 – PRESCSYNC[1:0] Prescaler and Counter Synchronization

These bits select if on retrigger 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 retrigger event.

Note: This bit field is enable-protected. This bit field is 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.

Note: This bit is enable-protected. This bit is not synchronized.

Value	Description
0	The TCC is halted in standby.
1	The TCC continues to run in standby.

Bits 10:8 – PRESCALER[2:0] Prescaler

These bits select the Counter prescaler factor.

Note: This bit field is enable-protected. This bit field is 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.

Note: This bit field is enable-protected. This bit field is not synchronized.

Table 40-10. 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

Notes:

1. This bit is write-synchronized: SYNCBUSY.ENABLE must be checked to ensure the CTRLA.ENABLE synchronization is complete.
2. 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 TCC (except DBGCTRL) to their initial state, and the TCC will be disabled.
Writing a '1' to CTRLA.SWRST will always take precedence; all other writes in the same write-operation will be discarded.

Notes:

1. When the CTRLA.SWRST is written, the user must poll SYNCBUSY.SWRST bit to know when the reset operation is complete.
2. During a SWRST, access to registers/bits without SWRST are disallowed until SYNCBUSY.SWRST is cleared by hardware.
3. This bit is not enable protected.

Value	Description
0	There is no Reset operation ongoing.
1	The Reset operation is ongoing.

40.7.2 Control B Clear

Name: CTRLBCLR
Offset: 0x04
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized, Read-Synchronized

This register allows the user to change this register without doing a read-modify-write operation. Changes in this register will also be reflected in the Control B Set register (CTRLBSET).
Note: This register is write-synchronized: SYNCBUSY.CTRLB must be checked to ensure the CTRLBCLR register synchronization is complete.

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
Writing a `0' to these bits has no effect.
Writing a '1' to any of these bits will clear the pending command.

Bits 4:3 – IDXCMD[1:0] Ramp Index Command
These bits can be used to force cycle A and cycle B changes in all RAMP2 operations. On timer/counter update condition, the command is executed, the IDX flag in STATUS register is updated and the IDXCMD command is cleared.
Writing zero to these bits has no effect.
Writing a '1' to any of these bits will clear 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 one-shot operation of the TCC. When one-shot operation is enabled, the TCC will stop 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 will disable 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 value from their buffered registers is enabled.
This bit has no effect when input capture operation is enabled.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will enable the registers updates on hardware UPDATE condition.

Value	Description
0	The CCBx, PERB, PGVB, PGOB, and SWAPBx buffer registers values *are* copied into the corresponding CCx, PER, PGV, PGO and SWAPx registers on hardware update condition.
1	The CCBx, PERB, PGVB, PGOB, and SWAPBx buffer registers values are *not* copied into the corresponding CCx, PER, PGV, PGO and SWAPx 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 clear the bit and make the counter count up.

Value	Description
0	The timer/counter is counting up (incrementing).
1	The timer/counter is counting down (decrementing).

40.7.3 Control B Set

Name: CTRLBSET
Offset: 0x05
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized, Read-Synchronized

This register allows the user to change this register without doing a read-modify-write operation. Changes in this register will also be reflected in the Control B Clear register (CTRLBCLR).
Note: This register is write-synchronized: SYNCBUSY.CTRLB must be checked to ensure the CTRLBCLR register synchronization is complete.

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 retriggering and stop commands of the TCC.
When a command has been executed, the CMD bit field will be read back as zero. The commands are executed on the next prescaled GCLK_TCCx clock cycle.
Writing zero to this bit field has no effect
Writing a value different from 0x0 to this bit field will issue a command for execution.

Important: This command requires synchronization before being executed.

A valid sequence is:
- Issue CMD command (CTRLBSET.CMD = command)
- Wait for CMD synchronization (SYNCBUSY.CTRLB = 0)
- Wait for CMD read back as zero (CTRLBSET.CMD = 0)

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 RAMP2 and RAMP2A operation. On timer/counter update condition, the command is executed, the IDX flag in STATUS register is updated and the IDXCMD command is cleared.
Writing a zero to these bits 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 value from their buffered registers is disabled. Disabling the update ensures that all buffer registers are valid before an hardware update is performed. After all the buffer registers are loaded correctly, the buffered registers can be unlocked.
This bit has no effect when input capture operation is enabled.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will disable the registers updates on hardware UPDATE condition.

Value	Description
0	The CCBx, PERB, PGVB, PGOB, and SWAPBx buffer registers values *are* copied into the corresponding CCx, PER, PGV, PGO and SWAPx registers on hardware update condition.
1	The CCBx, PERB, PGVB, PGOB, and SWAPBx buffer registers values are *not* copied into CCx, PER, PGV, PGO and SWAPx 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.
Note: When the TCC is counting down, the COUNT register must be initialized to the TOP value (PER or CC0 value depending on the mode).

Value	Description
0	The timer/counter is counting up (incrementing).
1	The timer/counter is counting down (decrementing).

40.7.4 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
					CC3	CC2	CC1	CC0
Access					R	R	R	R
Reset					0	0	0	0

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

Bits 8, 9, 10, 11 – CC Compare/Capture Channel x Synchronization Busy
This bit is cleared when the synchronization of Compare/Capture Channel x register between the clock domains is complete.
This bit is set when the synchronization of Compare/Capture Channel x register between clock domains is started.
CCx bit is available only for existing Compare/Capture Channels. For details on CC channels number, refer to each TCC feature list.

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 CTRLBSET or CTRLBCLR between the clock domains is complete.

This bit is set when the synchronization of CTRLBSET or CTRLBCLR 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.

40.7.5 Fault Control A

Name: FCTRLA
Offset: 0x0C
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 A Filter Value
These bits define the filter value applied on MCE0 event input line. The value must be set to zero when MCE0 event is used as synchronous event.

Bits 23:16 – BLANKVAL[7:0] Recoverable Fault A 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 (FCTRLA.BLANK).
When enabled, the fault input source is internally disabled for BLANKVAL* prescaled GCLK_TCC periods after the detection of the waveform edge.

Bit 15 – BLANKPRESC Recoverable Fault A 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_TCC.
1	Blank time is BLANKVAL* 64 * prescaled GCLK_TCC.

Bits 14:12 – CAPTURE[2:0] Recoverable Fault A Capture Action
These bits select the capture and Fault A interrupt/event conditions.

Table 40-11. Fault A Capture Action

Value	Name	Description
0x0	DISABLE	Capture on valid recoverable Fault A is disabled
0x1	CAPT	On rising edge of a valid recoverable Fault A, capture counter value on channel selected by CHSEL[1:0]. INTFLAG.FAULTA flag rises on each new captured value.
0x2	CAPTMIN	On rising edge of a valid recoverable Fault A, capture counter value on channel selected by CHSEL[1:0], if COUNT value is lower than the last stored capture value (CC). INTFLAG.FAULTA flag rises on each local minimum detection.

.....continued		
Value	**Name**	**Description**
0x3	CAPTMAX	On rising edge of a valid recoverable Fault A, capture counter value on channel selected by CHSEL[1:0], if COUNT value is higher than the last stored capture value (CC). INTFLAG.FAULTA flag rises on each local maximun detection.
0x4	LOCMIN	On rising edge of a valid recoverable Fault A, capture counter value on channel selected by CHSEL[1:0]. INTFLAG.FAULTA flag rises on each local minimum value detection.
0x5	LOCMAX	On rising edge of a valid recoverable Fault A, capture counter value on channel selected by CHSEL[1:0]. INTFLAG.FAULTA flag rises on each local maximun detection.
0x6	DERIV0	On rising edge of a valid recoverable Fault A, capture counter value on channel selected by CHSEL[1:0]. INTFLAG.FAULTA 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 A Capture Channel

These bits select the channel for capture operation triggered by recoverable Fault A.

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 A Halt Operation

These bits select the halt action for recoverable Fault A.

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 A Restart

Setting this bit enables restart action for Fault A.

Value	Description
0	Fault A restart action is disabled.
1	Fault A restart action is enabled.

Bits 6:5 – BLANK[1:0] Recoverable Fault A Blanking Operation

These bits, select the blanking start point for recoverable Fault A.

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 A Qualification

Setting this bit enables the recoverable Fault A input qualification.

Value	Description
0	The recoverable Fault A input is not disabled on CMPx value condition.
1	The recoverable Fault A input is disabled when output signal is at inactive level (CMPx == 0).

Bit 3 – KEEP Recoverable Fault A Keep

Setting this bit enables the Fault A keep action.

Value	Description
0	The Fault A state is released as soon as the recoverable Fault A is released.
1	The Fault A state is released at the end of TCC cycle.

Bits 1:0 – SRC[1:0] Recoverable Fault A Source

These bits select the TCC event input for recoverable Fault A.

Event system channel connected to MCE0 event input, must be configured to route the event asynchronously, when used as a recoverable Fault A input.

Value	Name	Description
0x0	DISABLE	Fault input disabled
0x1	ENABLE	MCE0 event input
0x2	INVERT	Inverted MCE0 event input
0x3	ALTFAULT	Alternate fault B state at the end of the previous period.

40.7.6 Fault Control B

Name: FCTRLB
Offset: 0x10
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 B Filter Value
These bits define the filter value applied on MCE1 event input line. The value must be set to zero when MCE1 event is used as synchronous event.

Bits 23:16 – BLANKVAL[7:0] Recoverable Fault B 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 (FCTRLB.BLANK).
When enabled, the fault input source is internally disabled for BLANKVAL* prescaled GCLK_TCC periods after the detection of the waveform edge.

Bit 15 – BLANKPRESC Recoverable Fault B 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_TCC.
1	Blank time is BLANKVAL* 64 * prescaled GCLK_TCC.

Bits 14:12 – CAPTURE[2:0] Recoverable Fault B Capture Action
These bits select the capture and Fault B interrupt/event conditions.

Table 40-12. Fault B Capture Action

Value	Name	Description
0x0	DISABLE	Capture on valid recoverable Fault B is disabled
0x1	CAPT	On rising edge of a valid recoverable Fault B, capture counter value on channel selected by CHSEL[1:0]. INTFLAG.FAULTB flag rises on each new captured value.
0x2	CAPTMIN	On rising edge of a valid recoverable Fault B, capture counter value on channel selected by CHSEL[1:0], if COUNT value is lower than the last stored capture value (CC). INTFLAG.FAULTB flag rises on each local minimum detection.

.....continued		
Value	**Name**	**Description**
0x3	CAPTMAX	On rising edge of a valid recoverable Fault B, capture counter value on channel selected by CHSEL[1:0], if COUNT value is higher than the last stored capture value (CC). INTFLAG.FAULTB flag rises on each local maximun detection.
0x4	LOCMIN	On rising edge of a valid recoverable Fault B, capture counter value on channel selected by CHSEL[1:0]. INTFLAG.FAULTB flag rises on each local minimum value detection.
0x5	LOCMAX	On rising edge of a valid recoverable Fault B, capture counter value on channel selected by CHSEL[1:0]. INTFLAG.FAULTB flag rises on each local maximun detection.
0x6	DERIV0	On rising edge of a valid recoverable Fault B, capture counter value on channel selected by CHSEL[1:0]. INTFLAG.FAULTB 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 B Capture Channel

These bits select the channel for capture operation triggered by recoverable Fault B.

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 B Halt Operation

These bits select the halt action for recoverable Fault B.

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 B Restart

Setting this bit enables restart action for Fault B.

Value	Description
0	Fault B restart action is disabled.
1	Fault B restart action is enabled.

Bits 6:5 – BLANK[1:0] Recoverable Fault B Blanking Operation

These bits, select the blanking start point for recoverable Fault B.

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 B Qualification

Setting this bit enables the recoverable Fault B input qualification.

Value	Description
0	The recoverable Fault B input is not disabled on CMPx value condition.
1	The recoverable Fault B input is disabled when output signal is at inactive level (CMPx == 0).

Bit 3 – KEEP Recoverable Fault B Keep

Setting this bit enables the Fault B keep action.

Value	Description
0	The Fault B state is released as soon as the recoverable Fault B is released.
1	The Fault B state is released at the end of TCC cycle.

Bits 1:0 – SRC[1:0] Recoverable Fault B Source

These bits select the TCC event input for recoverable Fault B.

Event system channel connected to MCE1 event input, must be configured to route the event asynchronously, when used as a recoverable Fault B input.

Value	Name	Description
0x0	DISABLE	Fault input disabled
0x1	ENABLE	MCE1 event input
0x2	INVERT	Inverted MCE1 event input
0x3	ALTFAULT	Alternate fault A state at the end of the previous period.

40.7.7 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
					DTIEN3	DTIEN2	DTIEN1	DTIEN0
Access					R/W	R/W	R/W	R/W
Reset					0	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_TCC 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_TCC clock cycles for the dead-time low side.

Bits 8, 9, 10, 11 – DTIEN Dead-time Insertion Generator x Enable
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 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 40.6.3.8. Waveform Extension.

40.7.8 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
	INVEN7	INVEN6	INVEN5	INVEN4	INVEN3	INVEN2	INVEN1	INVEN0
Access	R/W	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
	NRV7	NRV6	NRV5	NRV4	NRV3	NRV2	NRV1	NRV0
Access	R/W	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
	NRE7	NRE6	NRE5	NRE4	NRE3	NRE2	NRE1	NRE0
Access	R/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 31:28 – FILTERVAL1[3:0] Non-Recoverable Fault Input 1 Filter Value
These bits define the filter value applied on TCE1 event input line. When the TCE1 event input line is configured as an asynchronous 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 TCE0 event input line. When the TCE0 event input line is configured as an asynchronous event, this value must be 0x0.

Bits 16, 17, 18, 19, 20, 21, 22, 23 – INVEN Waveform Output x Inversion
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, 14, 15 – NRV NRVx Non-Recoverable State x Output Value
These bits define the value of the enabled override outputs, under non-recoverable fault condition.

Bits 0, 1, 2, 3, 4, 5, 6, 7 – NRE Non-Recoverable State x Output Enable
These bits enable the override of individual outputs by NRVx value, under non-recoverable fault condition.

Value	Description
0	Non-recoverable fault tri-state the output.
1	Non-recoverable faults set the output to NRVx level.

40.7.9 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 should not be changed by software while the TCC is enabled.
By default this bit is zero, and the on-chip debug (OCD) fault protection is disabled. When this bit is set, OCD fault protection is enabled and OCD break request from the OCD system will trigger 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 software Reset and should 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.

40.7.10 Event Control

Name: EVCTRL
Offset: 0x20
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
					MCEO3	MCEO2	MCEO1	MCEO0
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	23	22	21	20	19	18	17	16
					MCEI3	MCEI2	MCEI1	MCEI0
Access					R/W	R/W	R/W	R/W
Reset					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 – MCEO Match or Capture Channel x Event Output Enable
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 – MCEI Match or Capture Channel x Event Input Enable
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 the TCC and to enable recoverable Faults A and B.

Value	Description
0	Incoming events are disabled.
1	Incoming events are enabled.

Bits 14, 15 – TCEI Timer/Counter Event Input x Enable
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 – TCINV Timer/Counter Event x Invert Enable
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 begin or end of counter cycle depending of 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 depend of 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	START	An interrupt/event is generated when a new counter cycle starts.
0x1	END	An interrupt/event is generated when a counter cycle ends.
0x2	Reserved	Reserved
0x3	BOUNDARY	An interrupt/event is generated when a new counter cycle starts or a counter cycle ends.

Bits 5:3 – EVACT1[2:0] Timer/Counter Event Input 1 Action

These bits define the action the TCC will perform on TCE1 event input.

Value	Name	Description
0x0	OFF	Event action disabled.
0x1	RETRIGGER	Start, restart or retrigger TC on event
0x2	DIR	Direction control (ASYNC)
0x3	STOP	Stop TC on event
0x4	DEC	Decrement TC on event
0x5	Reserved	Reserved
0x6	PWP	Period captured into CC1 Pulse Width on CC0 (ASYNC)
0x7	FAULT	Non-recoverable Fault (ASYNC)

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 0.

Value	Name	Description
0x0	OFF	Event action disabled.
0x1	RETRIGGER	Start, restart or retrigger TC on event
0x2	COUNTEV	Count on event.
0x3	START	Start TC on event
0x4	INC	Increment TC on EVENT
0x5	COUNT	Count on active state of asynchronous event (ASYNC)
0x6	STAMP	Capture overflow times (Max value)
0x7	FAULT	Non-recoverable Fault (ASYNC)

40.7.11 Interrupt Enable Clear

Name: INTENCLR
Offset: 0x24
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 Set (INTENSET) register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					MC3	MC2	MC1	MC0
Access					R/W	R/W	R/W	R/W
Reset					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 – MC Match or Capture Channel x Interrupt Enable
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 Enable
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 Enable
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 Enable
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 Enable

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 Interrupt Enable

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 Enable

Writing a zero to this bit has no effect.
Writing a one 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 Enable

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 Enable

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 Enable

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 Enable

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 request.

Value	Description
0	The Overflow interrupt is disabled.

Value	Description
1	The Overflow interrupt is enabled.

40.7.12 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
					MC3	MC2	MC1	MC0
Access					R/W	R/W	R/W	R/W
Reset					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 – 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 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 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 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 zero to this bit has no effect.
Writing a one 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 Retrigger 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 request.

Value	Description
0	The Overflow interrupt is disabled.
1	The Overflow interrupt is enabled.

40.7.13 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
					MC3	MC2	MC1	MC0
Access					R/W	R/W	R/W	R/W
Reset					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 – MC Match or Capture Channel x Interrupt Flag
This flag is set after a match with the compare condition or once CCx register contain a valid capture value.
Writing a '0' to one of 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 15 – FAULT1 Non-Recoverable Fault 1 Interrupt Flag
This flag is set 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
This flag is set after a Non-Recoverable Fault 0 occurs.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Non-Recoverable Fault 0 interrupt flag.

Bit 13 – FAULTB Recoverable Fault B Interrupt Flag
This flag is set 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 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 after an 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
This flag is set when the RAMP index changes and the Lock Update bit is set (CTRLBSET.LUPD).
Writing a zero to this bit has no effect.
Writing a one 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 when the corresponding Match or Capture Channel x interrupt flag is one. In which case there is nowhere 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 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 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 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.

40.7.14 Status

Name: STATUS
Offset: 0x30
Reset: 0x00000001
Property: Read-Synchronized, Write-Synchronized

Note: This register is read- and write-synchronized: SYNCBUSY.STATUS must be checked to ensure the STATUS register synchronization is complete.

Important: 8-bit or 16-bit writes to this register are forbidden.

Bit	31	30	29	28	27	26	25	24
					CMP3	CMP2	CMP1	CMP0
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	23	22	21	20	19	18	17	16
					CCBUFV3	CCBUFV2	CCBUFV1	CCBUFV0
Access					R/W	R/W	R/W	R/W
Reset					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	SLAVE	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 – CMP Channel x Compare Value
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 – CCBUFV Channel x Compare or Capture Buffer Valid
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 on "update" (when CCBUF is copied into its CC register).

Bits 14, 15 – FAULT Non-recoverable Fault x State
This bit is set by hardware as soon as 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.
Once this bit is clear, the timer/counter will restart from the last COUNT value. To restart the timer/counter from BOTTOM, the timer/counter restart command must be executed before clearing the corresponding STATEx bit. For further details on timer/counter commands, refer to available commands description (40.7.3. CTRLBSET.CMD).

Bit 13 – FAULTB Recoverable Fault B State
This bit is set by hardware as soon as recoverable Fault B condition occurs.
This bit can be clear by hardware when Fault B action is resumed, or by writing a '1' to this bit when the corresponding FAULTBIN bit is low. If software halt command is enabled (FAULTB.HALT=SW), clearing this bit will release the timer/counter.

Bit 12 – FAULTA Recoverable Fault A State
This bit is set by hardware as soon as recoverable Fault A condition occurs.
This bit can be clear by hardware when Fault A action is resumed, or by writing a '1' to this bit when the corresponding FAULTAIN bit is low. If software halt command is enabled (FAULTA.HALT=SW), clearing this bit will release 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 UPDATE condition when CTRLB.LUPD is set, or by writing a '1' to this bit when CTRLB.LUPD is set.

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 UPDATE condition when CTRLB.LUPD is set, or by writing a '1' to this bit.

Bit 4 – SLAVE Client
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 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 depend on DRVCTRL.NRE and DRVCTRL.NRV registers.

Bit 1 – IDX Ramp Index
In RAMP2 operation, the bit is cleared during the cycle A and set during the cycle B. In RAMP1 operation, the bit always reads zero. For details on ramp operations, refer to 40.6.3.4. Ramp Operations.

Bit 0 – STOP Stop
This bit is set when the TCC is disabled either on a STOP command or on an UPDATE condition when One-Shot operation mode is enabled (CTRLBSET.ONESHOT=1).
This bit is clear on the next incoming counter increment or decrement.

Value	Description
0	Counter is running.
1	Counter is stopped.

40.7.15 Counter Value

Name: COUNT
Offset: 0x34
Reset: 0x00000000
Property: PAC Write-Protection, Write-Synchronized

Note: Prior to any read access, this register must be synchronized by the user by writing the TCC Command value to the Control B Set register (CTRLBSET.CMD=READSYNC).

Note: 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
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.
Note: When the TCC is configured as 16-bit timer/counter, the excess bits are read zero.

Note: 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 (depicted)
0x1 - DITH4	23:4
0x2 - DITH5	23:5
0x3 - DITH6	23:6

40.7.16 Pattern

Name: PATT
Offset: 0x38
Reset: 0x0000
Property: Write-Synchronized

Note: This register is write-synchronized: SYNCBUSY.PATT must be checked to ensure the PATT register synchronization is complete.

Bit	15	14	13	12	11	10	9	8
	PGV7	PGV6	PGV5	PGV4	PGV3	PGV2	PGV1	PGV0
Access	R/W	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
	PGE7	PGE6	PGE5	PGE4	PGE3	PGE2	PGE1	PGE0
Access	R/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 8, 9, 10, 11, 12, 13, 14, 15 – PGV Pattern Generation Output Value
This register holds the values of pattern for each waveform output.

Bits 0, 1, 2, 3, 4, 5, 6, 7 – PGE Pattern Generation Output Enable
This register holds the enable status of pattern generation for each waveform output. A bit written to '1' will override the corresponding SWAP output with the corresponding PGVn value.

40.7.17 Waveform

Name: WAVE
Offset: 0x3C
Reset: 0x00000000
Property: Enable-Protected Bits, Write-Synchronized Bits

Note: This register is write-synchronized: SYNCBUSY.WAVE must be checked to ensure the WAVE register bits synchronization is complete.

Bit	31	30	29	28	27	26	25	24
					SWAP3	SWAP2	SWAP1	SWAP0
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	23	22	21	20	19	18	17	16
					POL3	POL2	POL1	POL0
Access					R/W	R/W	R/W	R/W
Reset					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[2:0]				WAVEGEN[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 24, 25, 26, 27 – SWAP Swap DTI Output Pair x
Setting these bits enables output swap of DTI outputs [x] and [x+WO_NUM/2]. Note the DTIxEN settings will not affect the swap operation.
Note: This bit is not enable-protected. This bit is write-synchronized.

Bits 16, 17, 18, 19 – POL Channel Polarity x
Setting these bits enables the output polarity in single-slope and dual-slope PWM operations.
Note: This bit field is not enable-protected. This bit field is write-synchronized.

Value	Name	Description
0	SINGLESLOPEPOL0	Compare output is initialized to ~DIR and set to DIR when TCC counter matches CCx value (single-slope PWM waveform generation).
1	SINGLESLOPEPOL1	Compare output is initialized to DIR and set to ~DIR when TCC counter matches CCx value (single-slope PWM waveform generation).
0	DUALSLOPEPOL0	Compare output is set to ~DIR when TCC counter matches CCx value (dual-slope PWM waveform generation).
1	DUALSLOPEPOL1	Compare output is set to DIR when TCC counter matches CCx value (dual-slope PWM waveform generation).
0	DUALCOMPAREPOL0	Compare output is initialized to ~DIR, set to DIR when TCC counter matches CCx value and set to ~DIR when TCC counter matches CC[x+WO_NUM/2] value (dual compare PWM waveform generation).
1	DUALCOMPAREPOL1	Compare output is initialized to DIR, set to ~DIR when TCC counter matches CCx value and set to DIR when TCC counter matches CC[x+WO_NUM/2] value (dual compare PWM waveform generation).

Bits 8, 9, 10, 11 – CICCEN Circular CC Enable x

Setting this bits enables the compare circular buffer option on channel. When the bit is set, CCx register value is copied-back into the CCx register on UPDATE condition.

Note: This bit is not enable-protected. This bit is write-synchronized.

Bit 7 – CIPEREN Circular Period Enable

Setting this bits enable the period circular buffer option. When the bit is set, the PER register value is copied-back into the PERB register on UPDATE condition.

Note: This bit is not enable-protected. This bit is write-synchronized.

Bits 6:4 – RAMP[2:0] Ramp Operation

These bits select Ramp operation (RAMP).

Note: This bit field is enable-protected. This bit field is not write-synchronized.

Value	Name	Description
0x0	RAMP1	RAMP1 operation
0x1	RAMP2A	Alternative RAMP2 operation
0x2	RAMP2	RAMP2 operation
0x3	RAMP2C	Critical RAMP2 operation
0x4	RAMP2CS	Critical Swapped 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 should be used.

Note: This bit field is enable-protected. This bit field is not write-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	DPWM	Dual Compare PWM	PER	TOP/ZERO	Set/Clear	Clear	TOP	Zero
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	–

40.7.18 Period Value

Name: PER
Offset: 0x40
Reset: 0xFFFFFFFF
Property: Write-Synchronized

Note: This register is write-synchronized: SYNCBUSY.PER must be checked to ensure the PER register synchronization is complete.

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.
Note: When the TCC is configured as 16-bit timer/counter, the excess bits are read zero.

Note: 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 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)

40.7.19 Compare/Capture Channel x

Name: CC
Offset: 0x44 + n*0x04 [n=0..3]
Reset: 0x00000000
Property: Write-Synchronized

The CCx register represents the 16-, 24- bit value, CCx. The register has two functions, depending of the mode of operation.

For capture operation, this register represents the second buffer level and access point for the CPU and DMA.

For compare operation, this register is continuously compared to the counter value. Normally, the output form the comparator is then used for generating waveforms.

CCx register is updated with the buffer value from their corresponding CCBUFx register when an UPDATE condition occurs.

In addition, in match frequency operation, the CC0 register controls the counter period.

Note: This register is write-synchronized: SYNCBUSY.CCx must be checked to ensure the CCx register synchronization is complete.

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.
Note: When the TCC is configured as 16-bit timer/counter, the excess bits are read zero.

Note: This bit field occupies the m 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)

40.7.20 Pattern Buffer

Name: PATTBUF
Offset: 0x64
Reset: 0x0000
Property: Write-Synchronized

Note: This register is write-synchronized: SYNCBUSY.PATT must be checked to ensure the PATT register synchronization is complete.

Bit	15	14	13	12	11	10	9	8
	PGVB7	PGVB6	PGVB5	PGVB4	PGVB3	PGVB2	PGVB1	PGVB0
Access	R/W	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
	PGEB7	PGEB6	PGEB5	PGEB4	PGEB3	PGEB2	PGEB1	PGEB0
Access	R/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 8, 9, 10, 11, 12, 13, 14, 15 – PGVB Pattern Generation Output Value Buffer
This register is the buffer for the PGV register. If double buffering is used, valid content in this register is copied to the PGV register on an UPDATE condition or CTRLBSET.CMD = UPDATE command.

Bits 0, 1, 2, 3, 4, 5, 6, 7 – PGEB Pattern Generation Output Enable Buffer
This register is the buffer of the PGE register. If double buffering is used, valid content in this register is copied into the PGE register at an UPDATE condition or CTRLBSET.CMD = UPDATE command.

40.7.21 Period Buffer Value

Name: PERBUF
Offset: 0x6C
Reset: 0xFFFFFFFF
Property: Write-Synchronized

Note: This register is write-synchronized: SYNCBUSY.PER must be checked to ensure the PER register synchronization is complete.

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 PER register on UPDATE condition or CTRLBSET.CMD = UPDATE command when CTRLBSET.LUPD is 1.
Note: When the TCC is configured as 16-bit timer/counter, the excess bits are read zero.

Note: 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 – 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 or CTRLBSET.CMD = UPDATE command when CTRLBSET.LUPD is 1.
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)

40.7.22 Channel x Compare/Capture Buffer Value

Name: CCBUF
Offset: 0x70 + n*0x04 [n=0..3]
Reset: 0x00000000
Property: Write-Synchronized

CCBUFx is copied into CCx at TCC update time or CTRLBSET.CMD = UPDATE command when CTRLBSET.LUPD is 1.

Note: This register is write-synchronized: SYNCBUSY.CCx must be checked to ensure the CCx register synchronization is complete.

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
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.
Note: When the TCC is configured as 16-bit timer/counter, the excess bits are read zero.

Note: 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 – DITHERBUF[5:0] Dithering Buffer Cycle Number
These bits represent the CCx.DITHER bits buffer. When the double buffering is enable, DITHERBUF bits value is copied to the CCx.DITHER bits on an UPDATE condition or CTRLBSET.CMD = UPDATE command when CTRLBSET.LUPD is 1.

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. True Random Number Generator (TRNG)

41.1 Overview

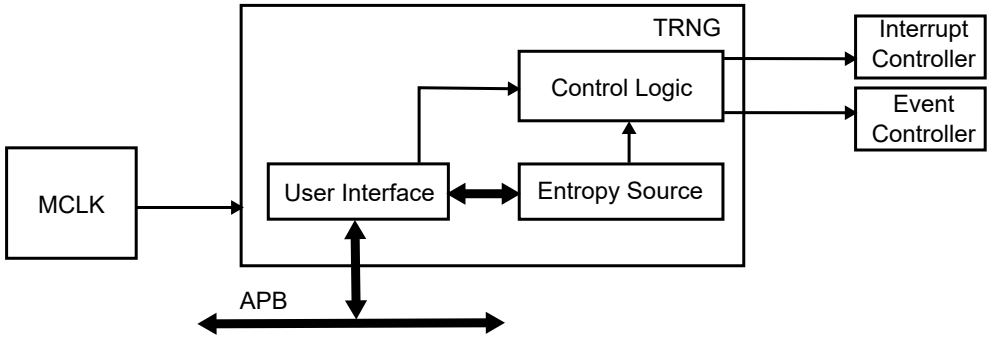
The True Random Number Generator (TRNG) generates unpredictable random numbers that are not generated by an algorithm.

41.2 Features

- Provides a 32-bit random number every 84 clock cycles

41.3 Block Diagram

Figure 41-1. TRNG Block Diagram.



41.4 Peripheral Dependencies

Table 41-1. TRNG Configuration Summary

Peripheral name	Base address	NVIC IRQ Index	MCLK AHB/APB Clocks	PAC Peripheral Identifier Index (PAC.WRCTRL)	Events (EVSYS)		Power Domain (PM.STDBYCFG)
					Users (EVSYS.USER)	Generators (EVSYS.CHANNEL)	
TRNG	0x42004400	68: DATARDY	CLK_TRNG_APB	81	—	86 : DATARDY	PDSW

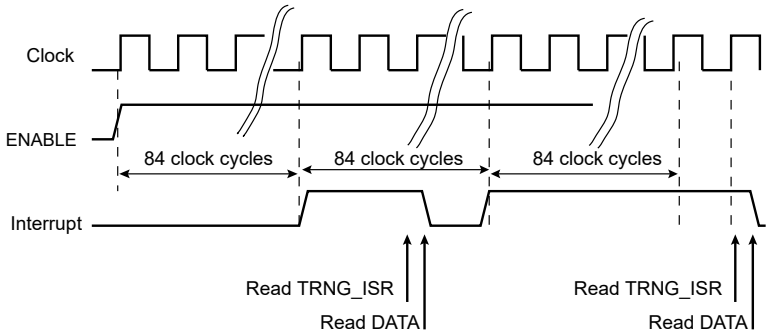
41.5 Functional Description

41.5.1 Principle of Operation

When the TRNG is enabled, the peripheral starts providing new 32-bit random numbers every 84 CLK_TRNG_APB clock cycles.

The TRNG can be configured to generate an interrupt or event when a new random number is available.

Figure 41-2. TRNG Data Generation Sequence



41.5.2 Basic Operation

41.5.2.1 Initialization

To operate TRNG, follow these steps:

1. Configure the clock source for CLK_TRNG_APB in the Main Clock Controller (MCLK) and enable the clock by writing a ‘1’ to the TRNG bit in the APB Mask register of the MCLK.
   – Optional: Enable the output event by writing a ‘1’ to the EVCTRL.DATARDYEO bit.
   – Optional: Enable the TRNG to Run in Standby Sleep mode by writing a ‘1’ to CTRLA.RUNSTDBY.
2. Enable the TRNG operation by writing a ‘1’ to CTRLA.ENABLE.
   Note: A delay between TRNG Enable (CTRLA.ENABLE = 1) and the first random number read is required. Refer to the TRNG Electrical Specifications.

The following register is enable-protected, meaning that it can only be written when the TRNG is disabled (CTRLA.ENABLE = 0): Event Control register (EVCTRL)

Enable-protection is denoted by the Enable-Protected property in the register description.

41.5.2.2 Enabling, Disabling and Resetting

The TRNG is enabled by writing '1' to the Enable bit in the Control A register (CTRLA.ENABLE). The TRNG is disabled by writing a zero to CTRLA.ENABLE.

41.5.3 Interrupts

The TRNG has the following interrupt source:

- Data Ready (DATARDY): Indicates that a new random number is available in the DATA register and ready to be read.
  This interrupt is a synchronous wake-up source. See Sleep Mode Controller for details.

The interrupt source has an interrupt flag associated with it. The interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.DATARDY) is set to ‘1’ when the interrupt condition occurs. The interrupt can be enabled by writing a '1' to the corresponding bit in the Interrupt Enable Set register (INTENSET.DATARDY), and disabled by writing a '1' to the corresponding bit in the Interrupt Enable Clear (INTENCLR.DATARDY) 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, or the interrupt is disabled. See 11.2. Nested Vector Interrupt Controller for details on how to clear interrupt flags.

Note that interrupts must be globally enabled for interrupt requests to be generated.

41.5.4 Events

The TRNG can generate the following output event:

- Data Ready (DATARDY): Generated when a new random number is available in the DATA register.

Writing '1' to the Data Ready Event Output bit in the Event Control Register (EVCTRL.DATARDYEO) enables the DTARDY event. Writing a '0' to this bit disables the corresponding output event. Refer to EVSYS – Event System for details on configuring the Event System.

41.5.5 Sleep Mode Operation

The Run in Standby bit in Control A register (CTRLA.RUNSTDBY) controls the behavior of the TRNG during standby sleep mode:

When this bit is '0', the TRNG is disabled during sleep, but maintains its current configuration.

When this bit is '1', the TRNG continues to operate during sleep and any enabled TRNG interrupt source can wake up the CPU.

41.5.6 Debug Operation

When the CPU is halted in debug mode the TRNG continues normal operation. If the TRNG 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.

41.6 Register Summary

Refer to the Registers Description section for more details on register properties and access permissions.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRLA	7:0		RUNSTDBY					ENABLE	
0x01 ... 0x03	Reserved									
0x04	EVCTRL	7:0								DATARDYEO
0x05 ... 0x07	Reserved									
0x08	INTENCLR	7:0								DATARDY
0x09	INTENSET	7:0								DATARDY
0x0A	INTFLAG	7:0								DATARDY
0x0B ... 0x1F	Reserved									
0x20	DATA	31:24	DATA[31:24]							
		23:16	DATA[23:16]							
		15:8	DATA[15:8]							
		7:0	DATA[7:0]							

41.6.1 Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00
Property: PAC Write-Protection

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

Bit 6 – RUNSTDBY Run in Standby

This bit controls how the ADC behaves during Standby Sleep mode:

Value	Description
0	The TRNG is halted during Standby Sleep mode.
1	The TRNG is not stopped in Standby Sleep mode.

Bit 1 – ENABLE Enable

Note: A delay between TRNG Enable (CTRLA.ENABLE = 1) and the first random number read is required. Refer to the TRNG Electrical Specifications.

Value	Description
0	The TRNG is disabled.
1	The TRNG is enabled.

41.6.2 Event Control

Name: EVCTRL
Offset: 0x04
Reset: 0x00
Property: PAC Write-Protection, Enable-Protected

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

Bit 0 – DATARDYEO Data Ready Event Output
This bit indicates whether the Data Ready event output is enabled and whether an output event will be generated when a new random value is ready.

Value	Description
0	Data Ready event output is disabled and an event will not be generated.
1	Data Ready event output is enabled and an event will be generated.

41.6.3 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 will also be reflected in the Interrupt Enable Set (INTENSET) register.

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

Bit 0 – DATARDY Data Ready Interrupt Enable
Writing a '1' to this bit will clear the Data Ready Interrupt Enable bit, which disables the corresponding interrupt request.

Value	Description
0	The DATARDY interrupt is disabled.
1	The DATARDY interrupt is enabled.

41.6.4 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 will also be reflected in the Interrupt Enable Clear (INTENCLR) register.

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

Bit 0 – DATARDY Data Ready Interrupt Enable
Writing a '1' to this bit will set the Data Ready Interrupt Enable bit, which enables the corresponding interrupt request.

Value	Description
0	The DATARDY interrupt is disabled.
1	The DATARDY interrupt is enabled.

41.6.5 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x0A
Reset: 0x00
Property: -

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

Bit 0 – DATARDY Data Ready
This flag is set when a new random value is generated, and an interrupt will be generated if INTENSET.DATARDY=1.
This flag is cleared by writing a '1' to the flag or by reading the DATA register.
Writing a '0' to this bit has no effect.

41.6.6 Output Data

Name: DATA
Offset: 0x20
Reset: -
Property: -

Bit	31	30	29	28	27	26	25	24
	DATA[31:24]							
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
	DATA[23:16]							
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
	DATA[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	x	x	x	x	x	x	x	x

Bit	7	6	5	4	3	2	1	0
	DATA[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	x	x	x	x	x	x	x	x

Bits 31:0 – DATA[31:0] Output Data
These bits hold the 32-bit randomly generated output data.

42. Configurable Custom Logic (CCL)

42.1 Overview

The Configurable Custom Logic (CCL) is a programmable logic peripheral which 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 LookUp Table (LUT) consists of three inputs, a truth table, an optional synchronizer/filter, and an optional edge detector. Each LUT can generate an output as a user programmable logic expression with three inputs. Inputs can be individually masked.

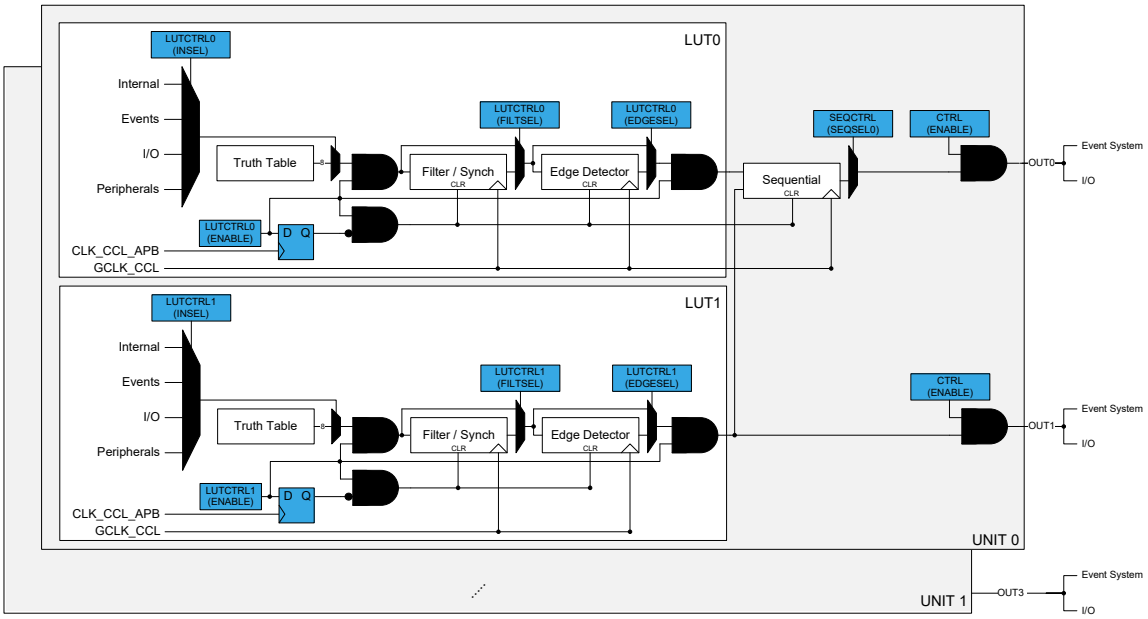
The output can be combinatorially generated from the inputs, and can be filtered to remove spikes. Optional sequential logic can be used. The inputs of the sequential module are individually controlled by two independent, adjacent LUT (LUT2n/LUT2n+1) outputs, enabling complex waveform generation.

42.2 Features

- Glue logic for general purpose PCB design
- Up to 4 programmable LookUp 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

42.3 Block Diagram

Figure 42-1. Configurable Custom Logic



42.4 Signal Description

Pin Name	Type	Description
OUT[3:0]	Digital output	Output from lookup table
IN[11:0]	Digital input	Input to lookup table

Refer to the Pinout for details on the pin mapping for this peripheral. One signal can be mapped on several pins.

42.5 Peripheral Dependencies

Table 42-1. CCL Configuration Summary

Peripheral name	Base address	MCLK AHB/APB Clocks	GCLK Peripheral Channel Index (GCLK.PCHCTRL)	PAC Peripheral Identifier Index (PAC.WRCTRL)	Events (EVSYS)		Power Domain (PM.STDBYCFG)
					Users (EVSYS.USER)	Generators (EVSYS.CHANNEL)	
CCL	0x42004800	CLK_CCL_APB	32: GCLK_CCL	82	48-51 : LUT0-3	87-90 : LUT0-3	PDSW

42.6 Functional Description

42.6.1 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.

42.6.2 Operation

42.6.2.1 Initialization

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. Refer to GCLK - Generic Clock Controller for details.

This generic clock is asynchronous to the user interface clock (CLK_CCL_APB).

The following bits are enable-protected, meaning that they can only be written when the corresponding even LUT is disabled (LUTCTRLn.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 (LUTCTRLn.ENABLE=0):

- LUT Control n (LUTCTRLn) register, except the ENABLE bit

Enable-protected bits in the LUTCTRLn registers can be written at the same time as LUTCTRLn.ENABLE is written to '1', but not at the same time as LUTCTRLn.ENABLE is written to '0'.

Enable-protection is denoted by the Enable-Protected property in the register description.

42.6.2.2 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 (LUTCTRLn.ENABLE). Each LUT is disabled by writing a '0' to LUTCTRLn.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 will be reset to their initial state, and the CCL will be disabled. Refer to 42.7.1. CTRL for details.

42.6.2.3 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 42-2. One or more inputs can be masked. The truth table for the expression is defined by TRUTH bits in LUT Control x register (LUTCTRLn.TRUTH).

Figure 42-2. Truth Table Output Value Selection

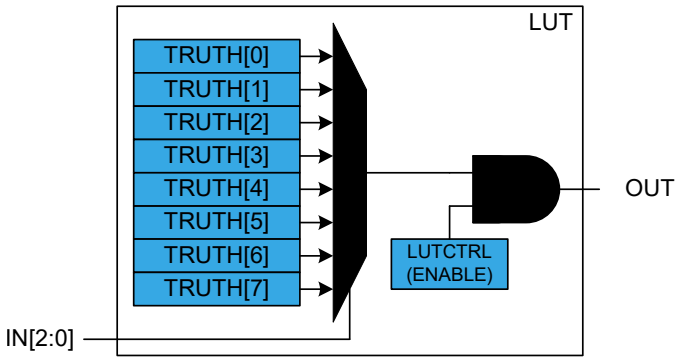


Table 42-2. Truth Table of LUT

IN[2]	IN[1]	IN[0]	OUT
0	0	0	TRUTH[0]
0	0	1	TRUTH[1]

.....continued			
IN[2]	IN[1]	IN[0]	OUT
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]

42.6.2.4 Truth Table Inputs Selection

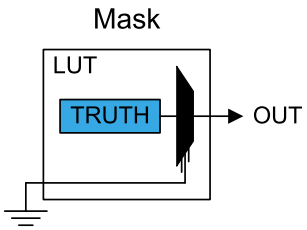
Input Overview
The inputs can be individually:

- Masked
- Driven by peripherals:
  - Analog comparator output (AC)
  - Timer/Counters waveform outputs (TC)
  - Timer/Counters for Control Applications waveform outputs (TCC)
  - 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 x of LUT n is configured by writing the Input x Source Selection bit in the LUT n Control register (LUTCTRLn.INSELx).

Masked Inputs (MASK)
When a LUT input is masked (LUTCTRLn.INSELx = MASK), the corresponding TRUTH input (IN) is internally tied to zero, as shown in this figure:

Figure 42-3. Masked Input Selection



Internal Feedback Inputs (FEEDBACK)
When selected (LUTCTRLn.INSELx = 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 input source for the LUT, see figure below for an example for LUT0 and LUT1. The sequential selection for each LUT follows the formula:

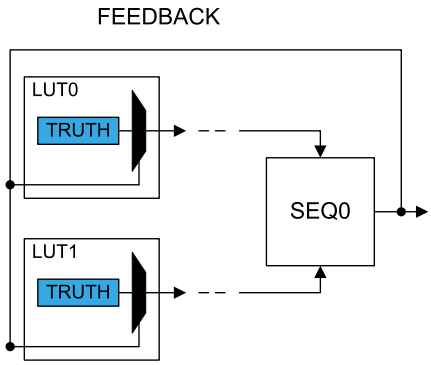
$\mathrm{IN}[2n] = \mathrm{SEQ}[n]$

$\mathrm{IN}[2n+1] = \mathrm{SEQ}[n]$

With *n* representing the sequencer number.

For details, refer to 42.6.2.7. Sequential Logic.

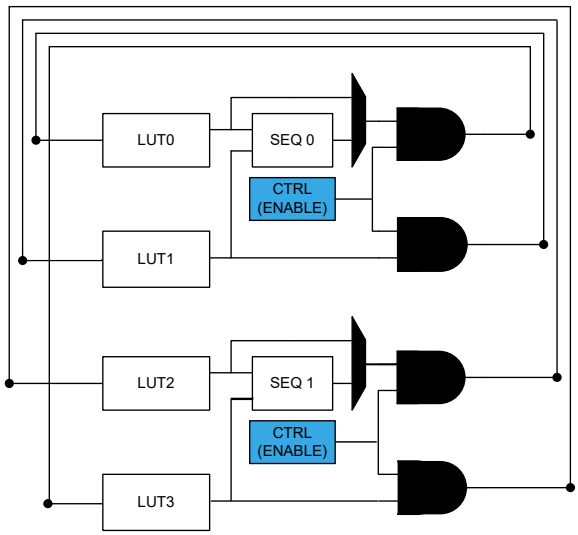
Figure 42-4. Feedback Input Selection



Linked LUT (LINK)

When selected (LUTCTRLn.INSELx = LINK), the subsequent LUT output is used as the LUT input, for example, LUT1 is the input for LUT0), as shown in the figure below:

Figure 42-5. Linked LUT Input Selection



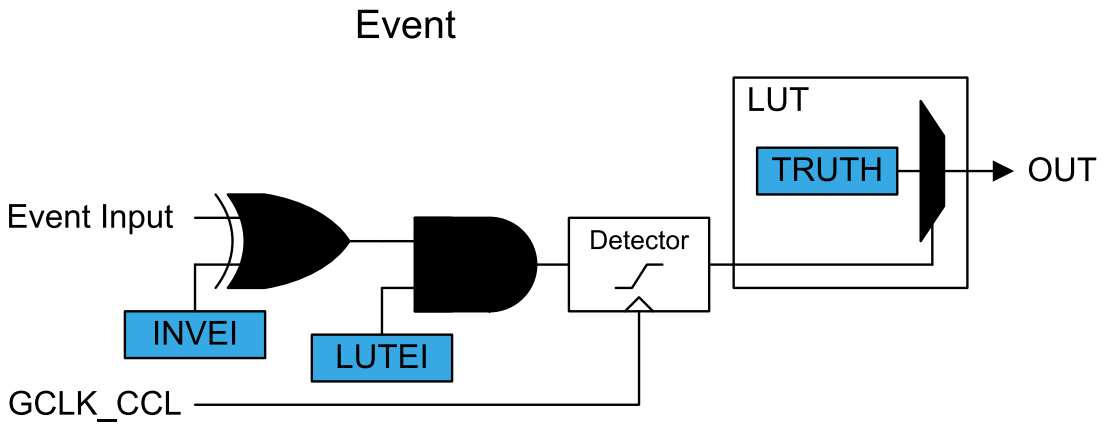
Internal Events Inputs Selection (EVENT and ASYNCEVENT)

Asynchronous events from the Event System can be used as input selection, as shown in the below image. For each LUT, one event input line is available and can be selected on each LUT input. Before enabling the event selection by writing LUTCTRLn.INSELx = 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. The following steps ensure proper operation:

1. Enable the GCLK_CCL clock.
2. Configure the Event System to route the event asynchronously.
3. Select the event input type (LUTCTRLn.INSELx).
4. If a strobe must be generated on the event input falling edge, write a '1' to the Inverted Event Input Enable bit in LUT Control register (LUTCTRLn.INVEI) .
5. Enable the event input by writing the Event Input Enable bit in LUT Control register (LUTCTRLn.LUTEI) to '1'.

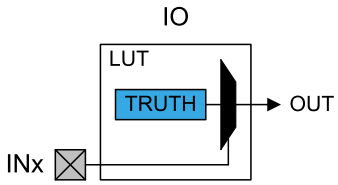
Figure 42-6. Event Input Selection



I/O Pin Inputs (IO)

When the IO pin is selected as LUT input (LUTCTRLn.INSELx = IO), the corresponding LUT input will be connected to the pin, as shown in the figure below.

Figure 42-7. I/O Pin Input Selection



Analog Comparator Inputs (AC)

The AC outputs can be used as input source for the LUT (LUTCTRLn.INSELx = AC).

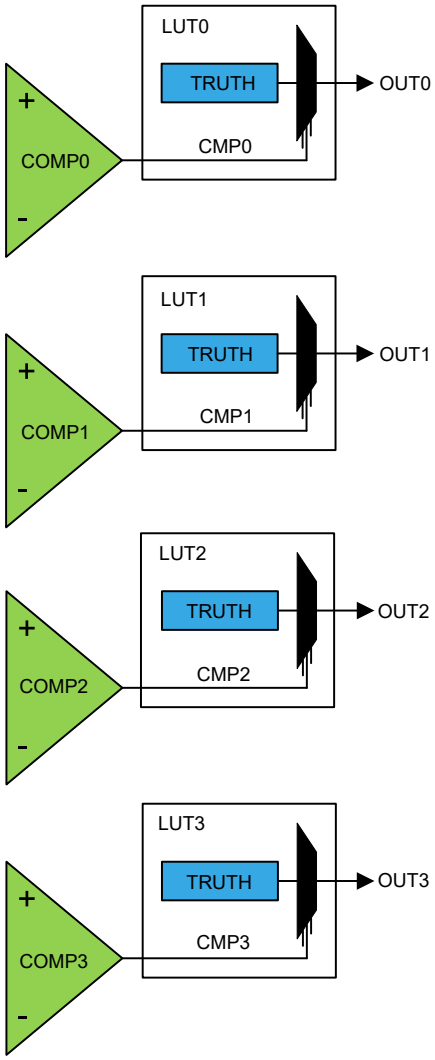
The analog comparator outputs are distributed following the formula:

$\mathrm{IN}[n] = AC[n \,\%\, \mathrm{ComparatorOutput\_Number}]$

With *n* representing the LUT number.

Before selecting the comparator output, the AC must be configured first.

Figure 42-8. AC Input Selection



Timer/Counter Inputs (TC)

The TC waveform output WO[0] can be used as input source for the LUT (LUTCTRLn.INSELx = TC). Only consecutive instances of the TC, i.e. TCn and the subsequent TC(n+1), are available as default and alternative TC selections (e.g., TC0 and TC1 are sources for LUT0, TC1 and TC2 are sources for LUT1, etc). See the figure below for an example for LUT0. More general, the Timer/Counter selection for each LUT follows the formula:

$\mathrm{IN}[n] = DefaultTC[n \,\%\, \mathrm{TC\_Instance\_Number}]$

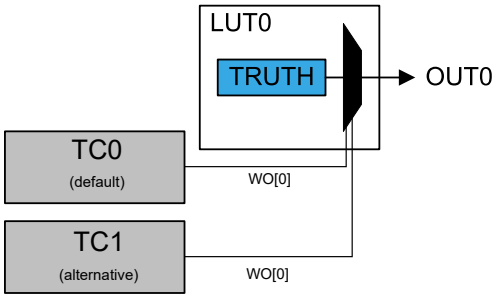
$\mathrm{IN}[n] = AlternativeTC[(n+1) \,\%\, \mathrm{TC\_Instance\_Number}]$

With *n* representing the LUT number.

Note that for not implemented TC_Instance_Number, the corresponding input is tied to ground.

Before selecting the waveform outputs, the TC must be configured first.

Figure 42-9. TC Input Selection



Timer/Counter for Control Application Inputs (TCC)
The TCC waveform outputs can be used as 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 shown in the figure below.

The TCC selection for each LUT follows the formula:

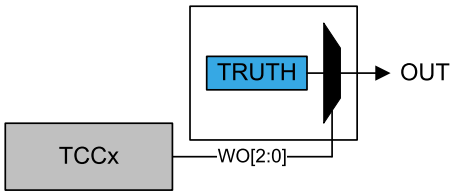
$\mathrm{IN}[n] = TCC[n \,\%\, TCC\_\mathrm{Instance\_Number}]$

With *n* representing the LUT number.

Note: TCC2 only outputs 2 WO signals, so TCC2.WO[0] is connected to both LUT2.IN[0] and LUT2.IN[2], and TCC2.WO[1] is connected to LUT2.IN[1].

Before selecting the waveform outputs, the TCC must be configured first.

Figure 42-10. TCC Input Selection



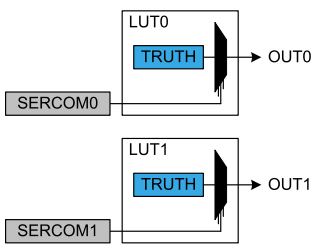
Serial Communication Output Transmit Inputs (SERCOM)
The serial engine transmitter output from Serial Communication Interface (SERCOM TX, TXD for USART, MOSI for SPI) can be used as input source for the LUT. The figure below shows an example for LUT0 and LUT1. The SERCOM selection for each LUT follows the formula:

$\mathrm{IN}[n] = SERCOM[n \,\%\, \mathrm{SERCOM\_Instance\_Number}]$

With *n* representing the LUT number.

Before selecting the SERCOM as input source, the SERCOM must be configured first: the SERCOM TX signal must be output on SERCOMn/PAD[0], which serves as input pad to the CCL.

Figure 42-11. SERCOM Input Selection



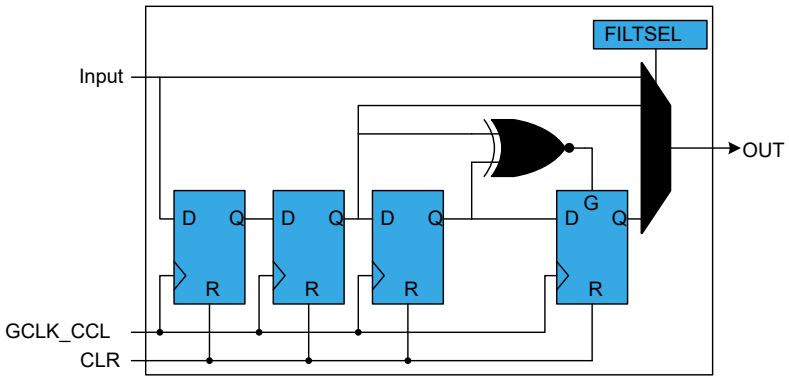
42.6.2.5 Filter

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 (LUTCTRLn.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 42-12. Filter



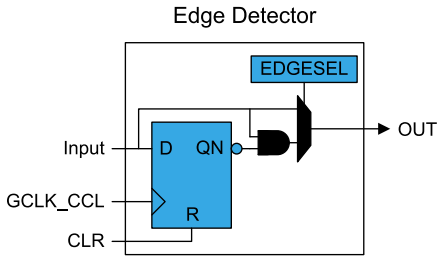
42.6.2.6 Edge Detector

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 should be inverted.

The edge detector is enabled by writing '1' to the Edge Selection bit in LUT Control register (LUTCTRLn.EDGESEL). In order to avoid unpredictable behavior, either the filter or synchronizer must be enabled.

Edge detection is disabled by writing a '0' to LUTCTRLn.EDGESEL. After disabling a LUT, the corresponding internal Edge Detector logic is cleared one APB clock cycle later.

Figure 42-13. Edge Detector



42.6.2.7 Sequential Logic

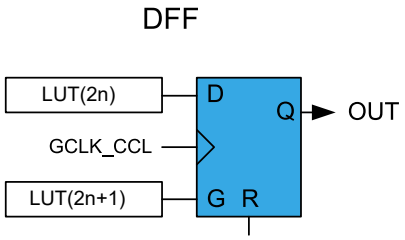
Each LUT pair can be connected to the internal sequential logic which can be configured to work as D flip flop, JK flip flop, gated D-latch or RS-latch by writing the Sequential Selection bits on the corresponding Sequential Control x register (SEQCTRLx.SEQSEL). Before using sequential logic, the GCLK_CCL 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 (LUT(2n)), and the G-input is driven by the odd LUT output (LUT(2n+1)), as shown in Figure 42-14.

Figure 42-14. D Flip Flop



When the even LUT is disabled (LUTCTRL(2n).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 Table 42-3.

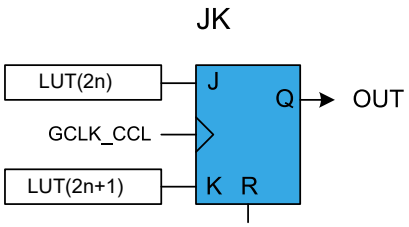
Table 42-3. DFF Characteristics

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 (LUT(2n)), and the K-input is driven by the odd LUT output (LUT(2n+1)), as shown in Figure 42-15.

Figure 42-15. JK Flip Flop



When the even LUT is disabled (LUTCTRL(2n).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 Table 42-4.

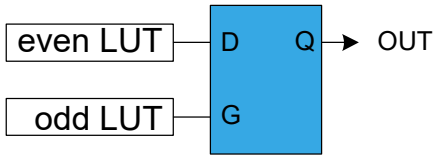
Table 42-4. JK Characteristics

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 (LUT(2n)), and the G-input is driven by the odd LUT output (LUT(2n+1)), as shown in Figure 42-14.

Figure 42-16. D-Latch



When the even LUT is disabled (LUTCTRL(2n).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 Table 42-5.

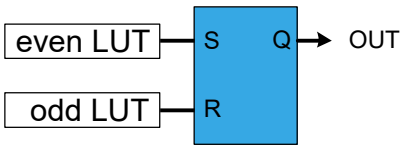
Table 42-5. D-Latch Characteristics

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 (LUT(2n)), and the R-input is driven by the odd LUT output (LUT(2n+1)), as shown in Figure 42-17.

Figure 42-17. RS-Latch



When the even LUT is disabled LUTCTRL(2n).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 Table 42-6.

Table 42-6. RS-Latch Characteristics

S	R	OUT
0	0	Hold state (no change)
0	1	Clear
1	0	Set
1	1	Forbidden state

42.6.3 Events

The CCL can generate the following output events:

- LUTn where n=0-3: Lookup 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

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.

For further information, refer to the 32. Event System (EVSYS).

42.6.4 Sleep Mode Operation

When using the GCLK_CCL internal clocking, writing the Run In Standby bit in the Control register (CTRL.RUNSTDBY) to '1' will allow GCLK_CCL to be enabled in Standby Sleep mode.

If CTRL.RUNSTDBY=0, the GCLK_CCL will be disabled in Standby Sleep mode. If the Filter, Edge Detector or Sequential logic are enabled, the LUT output will be forced to zero in STANDBY mode. In all other cases, the TRUTH table decoder will continue operation and the LUT output will be refreshed accordingly.

For further information, refer to the 22. Power Manager (PM).

42.6.5 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.

42.7 Register Summary

Refer to the Registers Description section for more details on register properties and access permissions.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRL	7:0		RUNSTDBY					ENABLE	SWRST
0x01 ... 0x03	Reserved									
0x04	SEQCTRL0	7:0					SEQSEL[3:0]			
0x05	SEQCTRL1	7:0					SEQSEL[3:0]			
0x06 ... 0x07	Reserved									
0x08	LUTCTRL0	31:24	TRUTH[7:0]							
		23:16		LUTEO	LUTEI	INVEI	INSEL2[3:0]			
		15:8	INSEL1[3:0]				INSEL0[3:0]			
		7:0	EDGESEL		FILTSEL[1:0]				ENABLE	
0x0C	LUTCTRL1	31:24	TRUTH[7:0]							
		23:16		LUTEO	LUTEI	INVEI	INSEL2[3:0]			
		15:8	INSEL1[3:0]				INSEL0[3:0]			
		7:0	EDGESEL		FILTSEL[1:0]				ENABLE	
0x10	LUTCTRL2	31:24	TRUTH[7:0]							
		23:16		LUTEO	LUTEI	INVEI	INSEL2[3:0]			
		15:8	INSEL1[3:0]				INSEL0[3:0]			
		7:0	EDGESEL		FILTSEL[1:0]				ENABLE	
0x14	LUTCTRL3	31:24	TRUTH[7:0]							
		23:16		LUTEO	LUTEI	INVEI	INSEL2[3:0]			
		15:8	INSEL1[3:0]				INSEL0[3:0]			
		7:0	EDGESEL		FILTSEL[1:0]				ENABLE	

42.7.1 Control

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

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 mode. The setting is ignored for configurations where the generic clock is not required. For details refer to 42.6.4. Sleep Mode Operation.

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.

42.7.2 Sequential Control x

Name: SEQCTRL
Offset: 0x04 + n*0x01 [n=0..1]
Reset: 0x00
Property: PAC Write-Protection, Enable-protected

Note: SEQCTRL0 (SEQCTRL1) register is Enable-protected when CCL.LUTCTRL0.ENABLE = 1 (CCL.LUTCTRL2.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

42.7.3 LUT Control n

Name: LUTCTRL
Offset: 0x08 + n*0x04 [n=0..3]
Reset: 0x00000000
Property: PAC Write-Protection, Enable-protected

Note: 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 [x=0..2]
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
0x5	AC	AC input source: CMP[n] (LUTn) [n=0..3]

Value	Name	Description
0x6	TC	TC input source: TC0 (LUT0) / TC1 (LUT1) / TC2 (LUT2) / TC0 (LUT3)
0x7	ALTTC	Alternative TC input source: TC1 (LUT0) / TC2 (LUT1) / TC0 (LUT2) / TC1 (LUT3)
0x8	TCC	TCC input source: TCCn (LUTn) [n=0..3]
0x9	SERCOM	SERCOM input source: SERCOMn (LUTn) [n=0..3]
0xA	Reserved	Reserved
0xB	ASYNCEVENT	Asynchronous event input source
0xC - 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

This bit is not Enable-Protected

Value	Description
0	The LUT is disabled.
1	The LUT is enabled.

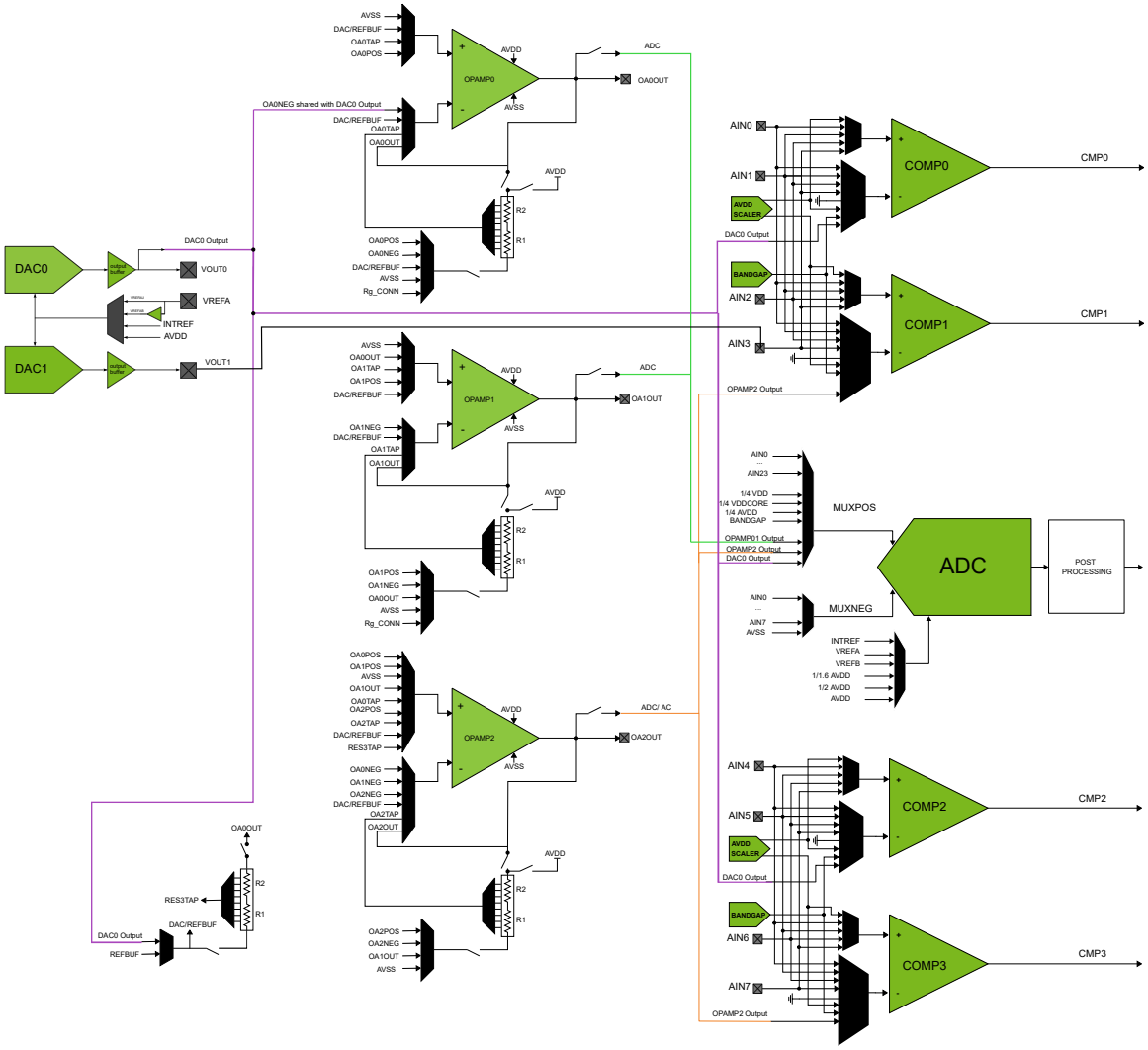
43. Analog Peripherals Considerations

This chapter provides a global view of the analog system, which is composed of the following analog peripherals: AC, ADC, DAC, OPAMP. The analog peripherals can be connected to each other as illustrated in the following block diagram.

⚠ CAUTION

- Only one analog output function can be used on the same pin.
- When DAC0 or DAC1 channel is enabled (DACCTRLx.ENABLE = 1), respective VOUT0 (PA02) or VOUT1 (PA07) pins cannot be used for other purposes (excluding analog inputs).
- Some OPAMP Outputs (OAxOUT) can be connected to a specific Analog Comparator or ADC Inputs (AINx) if they share the same pad, for example, OA0OUT can be connected to the Analog Comparator AIN3 or the ADC AIN5 input (the PA07 pin).
- The OA1OUT pin is only available on 64-pin and 100-pin packages, but can still be used on lower pin count packages as an internal input signal

Figure 43-1. Analog Signal Components Interconnections



43.1 Reference Voltages

Some analog peripherals require a reference voltage for proper operation.

Apart from external voltages (that is, AVDD or V_{REFx}), the device has a DETREF module that provides two different internal voltage references:

- BANDGAP: A stable voltage reference, fixed at 1.1V.
- INTREF: A variable voltage reference, configured by the Voltage References System Control register in the Supply Controller (SUPC.VREF).

The respective reference voltage source must be selected within each dedicated analog peripheral register:

- ADC: Reference Control register (ADC.REFCTRL)

Note: AC has a fixed reference voltage to BANDGAP value.

43.2 Analog On Demand Feature

The Analog On Demand feature allows the ADC and the AC analog peripherals to automatically enable the OPAMPx only when it is needed, thereby allowing a reduction in power consumption. It also allows the ADC analog block to be powered-off when a conversion is completed.
Note: The Analog On Demand is independent from the On Demand Clock request feature, which is used by peripherals to automatically request a source clock which was previously stopped.

OPAMP case
The Analog On Demand feature of the OPAMPx is activated by writing a '1' to the OPAMP.OPAMPCTRLx.ONDEMAND bit.

In that case, the OPAMPx is automatically enabled when the ADC or the AC requests it (as an input) and is automatically disabled when no more requests are coming from these peripherals.

⚠ CAUTION The Analog On Demand feature is not fully supported on cascaded OPAMPs. If several OPAMPs are cascaded together, only the OPAMPx that is connected to the ADC or AC can be enabled/disabled automatically. Upstream OPAMPs will not benefit from this feature.

In Standby Sleep mode, the Analog On Demand feature is still supported if OPAMP.OPAMPCTRLx.RUNSTDBY=1.

If OPAMP.OPAMPCTRLx.RUNSTDBY=0, the OPAMPx will be disabled entering this Sleep mode.

ADC case
For the ADC peripheral, Analog On Demand feature is enabled by writing the ADC.CTRLA.ONDEMAND bit to '1'.

When this feature is activated, the analog block is powered-off when the conversion is complete.

In Sleep mode, when an ADC start request is detected, the analog block is powered-on again and the ADC starts a new conversion after the start-up time delay.
Note: If the OPAMPx is set to accept Analog On Demand requests but the ADC is not, the ADC will send continuous requests to the OPAMPx keeping it enabled until the ADC is switching on another input.

AC case
For the AC peripheral,there is no explicit ONDEMAND bit.

Analog On Demand requests are issued either when the AC is used in Single-Shot mode, or when comparisons are triggered by events from the Event System.

44. Analog-to-Digital Converter (ADC)

44.1 Overview

The Analog-to-Digital Converter (ADC) converts analog signals to digital values. The ADC has up to 12-bit resolution, and is capable of a sampling rate of up to 1MSPS. The input selection is flexible, and both differential and single-ended measurements can be performed. In addition, several internal signal inputs are available. The ADC can provide both signed and unsigned results.

ADC measurements can be started by either application software or an incoming event from another peripheral in the device. ADC measurements can be started with predictable timing, and without software intervention.

Both internal and external reference voltages can be used.

The bandgap voltage, as well as the scaled I/O and core voltages, can also be measured by the ADC.

The ADC has a compare function for accurate monitoring of user-defined thresholds, with minimum software intervention required.

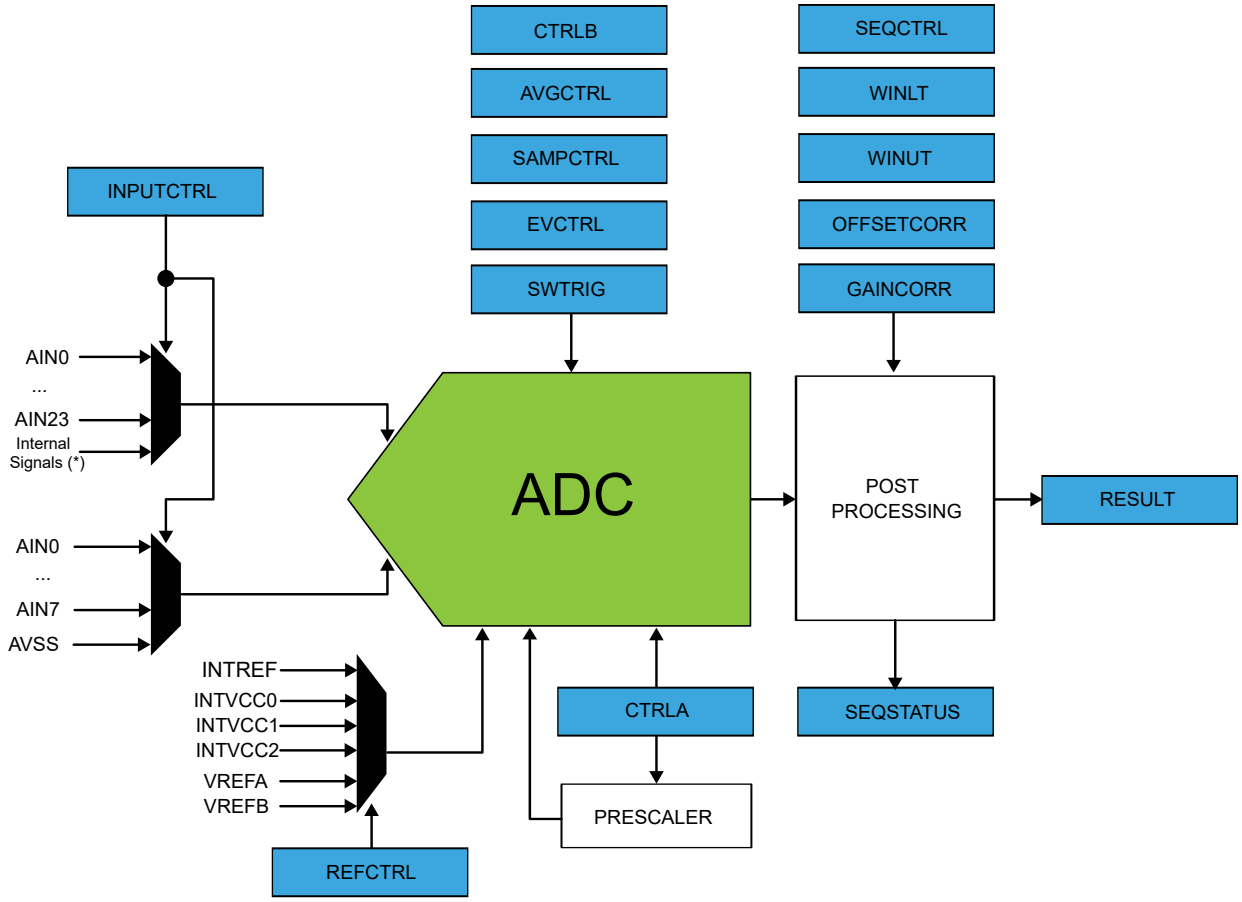
The ADC can be configured for 8-bit, 10-bit or 12-bit results. ADC conversion results are provided left- or right-adjusted, which eases calculation when the result is represented as a signed value. It is possible to use DMA to move ADC results directly to memory or peripherals when conversions are done.

44.2 Features

- 8-bit, 10-bit or 12-bit resolution
- Up to 1,000,000 samples per second (1MSPS)
- Differential and single-ended inputs
  - Up to 24 analog inputs
    24 positive and 8 negative, including internal and external
- Internal inputs:
  - Bandgap voltage
  - Scaled core supply
  - Scaled I/O supply
  - DAC0
  - OPAMP
- Single, continuous, and sequencing options
- Windowing monitor with selectable channel
- Built-in internal reference and external reference options
- Event-triggered conversion for accurate timing (one event input)
- Optional DMA transfer of conversion settings or result
- Hardware gain and offset compensation
- Averaging and oversampling with decimation to support up to 16-bit result
- Selectable sampling time
- Flexible Power or Throughput rate management

44.3 Block Diagram

Figure 44-1. ADC Block Diagram



Note: Internal Signals are described in the ADC INPUTCTRL register.

44.4 Signal Description

Signal	Description	Type
VREFA/B	Analog input	External reference voltage
AIN[23..0]	Analog input	Analog input channels

Notes:

- One signal can be mapped on several pins.
- VREFA is shared between the ADC and DAC peripherals.

44.5 Peripheral Dependencies

Table 44-1. ADC Configuration Summary

Peripheral name	Base address	NVIC IRQ Index	MCLK AHB/APB Clocks	GCLK Peripheral Channel Index (GCLK.PCHCTRL)	PAC Peripheral Identifier Index (PAC.WRCTRL)	DMA Trigger Source Index (DMAC.CHCTRLB)	Events (EVSYS)		Power Domain (PM.STDBYCFG)
							Users (EVSYS.USER)	Generators (EVSYS.CHANNEL)	
ADC	0x42003800	62: OVERRUN, WINMON 63: RESRDY	CLK_ADC_APB	28: GCLK_ADC	78	41: RESRDY	38: START 39 : FLUSH	74: RESRDY 75: WINMON	PDSW

I/O-pins (AINx), as well as the VREFA/VREFB reference voltage pin are analog inputs to the ADC. Any internal reference source, such as a bandgap voltage reference, or DAC must be configured and enabled prior to its use with the ADC.

The analog signals of AC, ADC, DAC and OPAMP can be interconnected. The AC and ADC peripheral can request the OPAMP using an analog ONDEMAND functionality.

For more information, refer to *Analog Connections of Peripherals*.

44.6 Functional Description

44.6.1 Principle of Operation

By default, the ADC provides results with 12-bit resolution. 8-bit or 10-bit results can be selected in order to reduce the conversion time, see 44.6.2.8. Conversion Timing and Sampling Rate.

The ADC has an oversampling with decimation option that can extend the resolution to 16 bits. The input values can be either internal or external (connected I/O pins). The user can also configure whether the conversion should be single-ended or differential.

44.6.2 Basic Operation

44.6.2.1 Initialization

The BIASREFBUF and BIASCOMP calibration values from the production test must be loaded from the NVM Software Calibration Area into the ADC Calibration register (CALIB) by software to achieve specified accuracy.

The following registers are enable-protected, meaning that they can only be written when the ADC is disabled (CTRLA.ENABLE=0):

- Control B register (CTRLB)
- Reference Control register (REFCTRL)
- Event Control register (EVCTRL)
- Calibration register (CALIB)

Enable-protection is denoted by the "Enable-Protected" property in the register description.

44.6.2.2 Enabling, Disabling and Resetting

The ADC is enabled by writing a '1' to the Enable bit in the Control A register (CTRLA.ENABLE). The ADC is disabled by writing CTRLA.ENABLE=0.

The ADC is reset by writing a '1' to the Software Reset bit in the Control A register (CTRLA.SWRST). All registers in the ADC, except DBGCTRL, will be reset to their initial state, and the ADC will be disabled. Refer to 44.7.1. CTRLA for details.

44.6.2.3 Operation

In the most basic configuration, the ADC samples values from the configured internal or external sources (INPUTCTRL register). The rate of the conversion depends on the combination of the GCLK_ADC frequency and the clock prescaler.

To convert analog values to digital values, the ADC needs to be initialized first, as described in the Initialization section. Data conversion can be started either manually by setting the Start bit in the Software Trigger register (SWTRIG.START=1), or automatically by configuring an automatic trigger to initiate the conversions. A free-running mode can be used to continuously convert an input channel. When using free-running mode the first conversion must be started, while subsequent conversions will start automatically at the end of previous conversions.

The result of the conversion is stored in the Result register (RESULT) overwriting the result from the previous conversion.

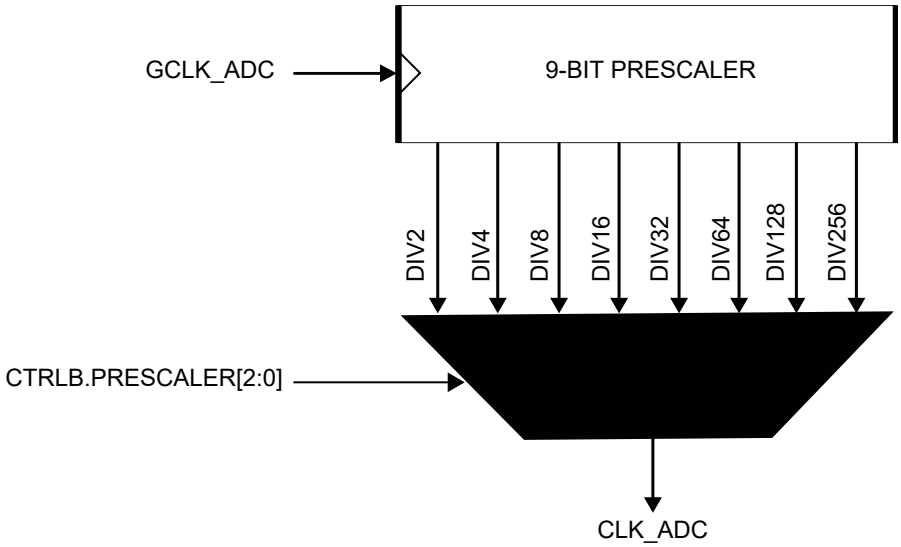
To avoid data loss if more than one channel is enabled, the conversion result must be read as soon as it is available (INTFLAG.RESRDY). Failing to do so will result in an overrun error condition, indicated by the OVERRUN bit in the Interrupt Flag Status and Clear register (INTFLAG.OVERRUN).

To enable one of the available interrupts sources, the corresponding bit in the Interrupt Enable Set register (INTENSET) must be written to '1'.

44.6.2.4 Prescaler Selection

The ADC is clocked by GCLK_ADC. There is also a prescaler in the ADC to enable conversion at lower clock rates. Refer to CTRLB for details on prescaler settings. Refer to 44.6.2.8. Conversion Timing and Sampling Rate for details on timing and sampling rate.

Figure 44-2. ADC Prescaler



Note: The minimum prescaling factor is DIV2.

44.6.2.5 Reference Configuration

The ADC has various sources for its reference voltage V_{REF}. The Reference Voltage Selection bit field in the Reference Control register (REFCTRL.REFSEL) determines which reference is selected. By default, the internal voltage reference INTREF is selected. Based on customer application requirements, the external or internal reference can be selected. Refer to REFCTRL.REFSEL for further details on available selections.

44.6.2.6 ADC Resolution

The ADC supports 8-bit, 10-bit or 12-bit resolution. Resolution can be changed by writing the Resolution bit group in the Control C register (CTRLC.RESSEL). The RESSEL bit group has a specific configuration mode used to support Accumulation or Oversampling and Decimation features.

By default, the ADC resolution is set to 12 bits. The resolution affects the propagation delay, see also 44.6.2.8. Conversion Timing and Sampling Rate.

44.6.2.7 Differential and Single-Ended Conversions

The ADC has two conversion options: differential and single-ended:

If the positive input is always positive, the single-ended conversion should be used in order to have full 12-bit resolution in the conversion.

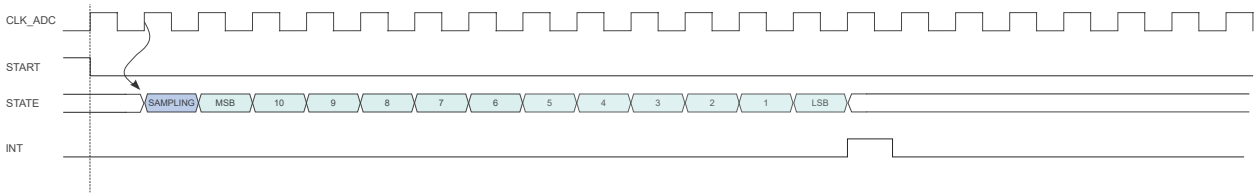
If the positive input may go below the negative input, the differential mode should be used in order to get correct results.

The differential mode is enabled by setting DIFFMODE bit in the Control C register (CTRLC.DIFFMODE). Both conversion types could be run in single mode or in free-running mode. When the free-running mode is selected, an ADC input will continuously sample the input and performs a new conversion. The INTFLAG.RESRDY bit will be set at the end of each conversion.

44.6.2.8 Conversion Timing and Sampling Rate

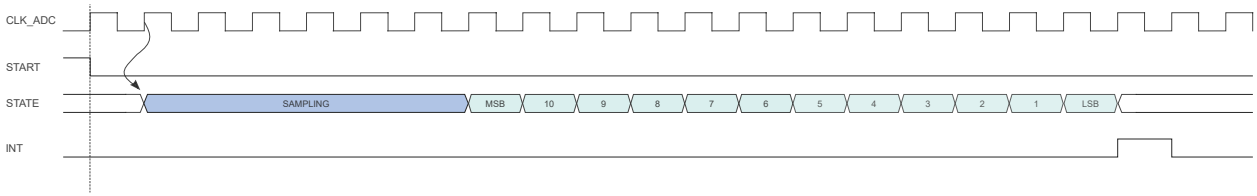
The following figure shows the ADC timing for one single conversion. A conversion starts after the software or event start are synchronized with the GCLK_ADC clock. The input channel is sampled in the first half CLK_ADC period.

Figure 44-3. ADC Timing for One Conversion in 12-bit Resolution



The sampling time can be increased by using the Sampling Time Length bit group in the Sampling Time Control register (SAMPCTRL.SAMPLEN). As example, the next figure is showing the timing conversion with sampling time increased to six CLK_ADC cycles.

Figure 44-4. ADC Timing for One Conversion with Increased Sampling Time, 12-bit



The ADC provides also offset compensation, see the following figure. The offset compensation is enabled by the Offset Compensation bit in the Sampling Control register (SAMPCTRL.OFFCOMP).

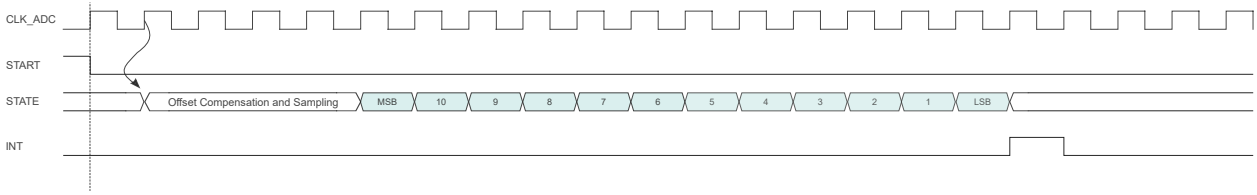
Note: ADC sampling time is fixed to 4 ADC Clock cycles when offset compensation (OFFCOMP=1) is used.

In free running mode, the sampling rate R_S is calculated by

$R_S = f_{CLK\_ADC} / (n_{SAMPLING} + n_{OFFCOMP} + n_{DATA})$

Here, $n_{SAMPLING}$ is the sampling duration in CLK_ADC cycles, $n_{OFFCOMP}$ is the offset compensation duration in clock cycles, and n_{DATA} is the bit resolution. f_{CLK_ADC} is the ADC clock frequency from the internal prescaler: $f_{CLK\_ADC} = f_{GCLK\_ADC} / 2^{(1 + CTRLB.PRESCALER)}$

Figure 44-5. ADC Timing for One Conversion with Offset Compensation, 12-bit



The impact of resolution on the sampling rate is seen in the next two figures, where free-running sampling in 12-bit and 8-bit resolution are compared.

Figure 44-6. ADC Timing for Free Running in 12-bit Resolution

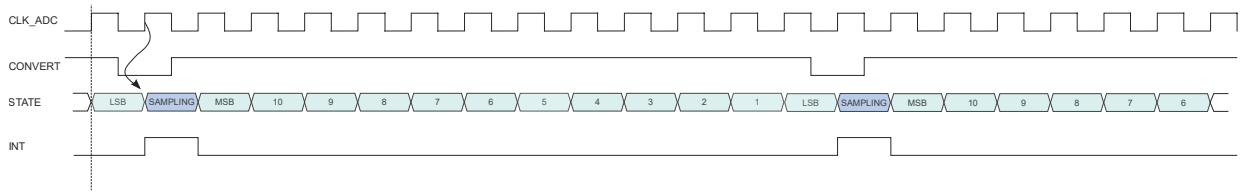
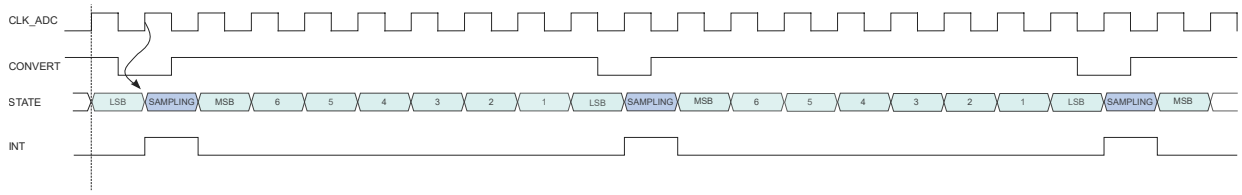


Figure 44-7. ADC Timing for Free Running in 8-bit Resolution



The propagation delay of an ADC measurement is given by:

$$\text{PropagationDelay} = \frac{1 + \text{Resolution}}{f_{\text{ADC}}}$$

> **Example.** In order to obtain 1MSPS in 12-bit resolution with a sampling time length of four CLK_ADC cycles, f_{CLK_ADC} must be 1MSPS * (4 + 12) = 16MHz. As the minimal division factor of the prescaler is 2, GCLK_ADC must be 32MHz.

44.6.2.9 Accumulation

The result from multiple consecutive conversions can be accumulated. The number of samples to be accumulated is specified by the Sample Number field in the Average Control register (AVGCTRL.SAMPLENUM). When accumulating more than 16 samples, the result will be too large to match the 16-bit RESULT register size. To avoid overflow, the result is right shifted automatically to fit within the available register size. The number of automatic right shifts is specified in the table below.

Note: To perform the accumulation of two or more samples, the Conversion Result Resolution field in the Control C register (CTRLC.RESSEL) must be set depending on the final result precision. For resolutions strictly higher than 12 bits, RESSEL must be set to 16 bits.

Table 44-2. Accumulation

Number of Accumulated Samples	AVGCTRL. SAMPLENUM	Number of Automatic Right Shifts	Final Result Precision	Automatic Division Factor
1	0x0	0	12 bits	0
2	0x1	0	13 bits	0
4	0x2	0	14 bits	0
8	0x3	0	15 bits	0
16	0x4	0	16 bits	0
32	0x5	1	16 bits	2
64	0x6	2	16 bits	4
128	0x7	3	16 bits	8
256	0x8	4	16 bits	16
512	0x9	5	16 bits	32

.....continued				
Number of Accumulated Samples	AVGCTRL. SAMPLENUM	Number of Automatic Right Shifts	Final Result Precision	Automatic Division Factor
1024	0xA	6	16 bits	64
Reserved	0xB –0xF		12 bits	0

44.6.2.10 Averaging

Averaging is a feature that increases the sample accuracy at the cost of a reduced sampling rate. This feature is suitable when operating in noisy conditions.

Averaging is done by accumulating 'm' samples, as described in 44.6.2.9. Accumulation, and dividing the result by 'm'. The averaged result is available in the RESULT register. The number of samples to be accumulated is specified by writing to AVGCTRL.SAMPLENUM as shown in Table 44-3.

The division is obtained by a combination of the automatic right shift described above, and an additional right shift that must be specified by writing to the Adjusting Result/Division Coefficient field in AVGCTRL (AVGCTRL.ADJRES), as described in Table 44-3.

Averaging AVGCTRL.SAMPLENUM samples will reduce the un-averaged sampling rate by a factor $\frac{1}{\text{AVGCTRL.SAMPLENUM}}$.

When the averaged result is available, the INTFLAG.RESRDY bit will be set.

Table 44-3. Averaging

Number of Accumulated Samples	AVGCTRL. SAMPLENUM	Intermediate Result Precision	Number of Automatic Right Shifts	Division Factor	AVGCTRL.ADJRES	Total Number of Right Shifts	Final Result Precision	Automatic Division Factor
1	0x0	12 bits	0	1	0x0		12 bits	0
2	0x1	13	0	2	0x1	1	12 bits	0
4	0x2	14	0	4	0x2	2	12 bits	0
8	0x3	15	0	8	0x3	3	12 bits	0
16	0x4	16	0	16	0x4	4	12 bits	0
32	0x5	17	1	16	0x4	5	12 bits	2
64	0x6	18	2	16	0x4	6	12 bits	4
128	0x7	19	3	16	0x4	7	12 bits	8
256	0x8	20	4	16	0x4	8	12 bits	16
512	0x9	21	5	16	0x4	9	12 bits	32
1024	0xA	22	6	16	0x4	10	12 bits	64
Reserved	0xB –0xF				0x0		12 bits	0

44.6.2.11 Oversampling and Decimation

By using oversampling and decimation, the ADC resolution can be increased from 12 bits up to 16 bits, for the cost of reduced effective sampling rate.

Note: To perform oversampling and decimation, the Conversion Result Resolution field in the Control C register (CTRLC.RESSEL) must be set to 16-bit mode.

To increase the resolution by 'n' bits, 4^n samples must be accumulated. The result must then be right shifted by 'n' bits. This right shift is a combination of the automatic right shift and the value written to AVGCTRL.ADJRES. To

obtain the correct resolution, the ADJRES must be configured as described in the table below. This method will result in ‘n’ bit extra LSB resolution.

Table 44-4. Configuration Required for Oversampling and Decimation

Result Resolution	Number of Samples to Average	AVGCTRL.SAMPLENUM[3:0]	Number of Automatic Right Shifts	AVGCTRL.ADJRES[2:0]
13 bits	$4^1 = 4$	0x2	0	0x1
14 bits	$4^2 = 16$	0x4	0	0x2
15 bits	$4^3 = 64$	0x6	2	0x1
16 bits	$4^4 = 256$	0x8	4	0x0

44.6.2.12 Automatic Sequences

The ADC has the ability to automatically sequence a series of conversions. This means that each time the ADC receives a start-of-conversion request, it can perform multiple conversions automatically. All of the 32 positive inputs can be included in a sequence by writing to corresponding bits in the Sequence Control register (SEQCTRL). The order of the conversion in a sequence is the lower positive MUX selection to upper positive MUX (AIN0, AIN1, AIN2 ...). In differential mode, the negative inputs selected by MUXNEG field, will be used for the entire sequence.

When a sequence starts, the Sequence Busy status bit in Sequence Status register (SEQSTATUS.SEQBUSY) will be set. When the sequence is complete, the Sequence Busy status bit will be cleared.

Each time a conversion is completed, the Sequence State bit in Sequence Status register (SEQSTATUS.SEQSTATE) will store the input number from which the conversion is done. The result will be stored in the RESULT register, and the Result Ready Interrupt Flag (INTFLAG.RESRDY) is set.

If additional inputs must be scanned, the ADC will automatically start a new conversion on the next input present in the sequence list.

Note that if SEQCTRL register has no bits set to '1', the conversion is done with the selected MUXPOS input.

44.6.2.13 Window Monitor

The window monitor feature allows the conversion result in the RESULT register to be compared to predefined threshold values. The window mode is selected by setting the Window Monitor Mode bits in the Control C register (CTRLC.WINMODE). Threshold values must be written in the Window Monitor Lower Threshold register (WINLT) and Window Monitor Upper Threshold register (WINUT).

If differential input is selected, the WINLT and WINUT are evaluated as signed values. Otherwise they are evaluated as unsigned values. The significant WINLT and WINUT bits are given by the precision selected in the Conversion Result Resolution bit group in the Control C register (CTRLC.RESSEL). This means that for example in 8-bit mode, only the eight lower bits will be considered. In addition, in differential mode, the eighth bit will be considered as the sign bit, even if the ninth bit is zero.

The INTFLAG.WINMON interrupt flag will be set if the conversion result matches the window monitor condition.

44.6.2.14 Offset and Gain Correction

Inherent gain and offset errors affect the absolute accuracy of the ADC.

The offset error is defined as the deviation of the actual ADC transfer function from an ideal straight line at zero input voltage. The offset error cancellation is handled by the Offset Correction register (OFFSETCORR). The offset correction value is subtracted from the converted data before writing the Result register (RESULT).

The gain error is defined as the deviation of the last output step’s midpoint from the ideal straight line, after compensating for offset error. The gain error cancellation is handled by the Gain Correction register (GAINCORR).

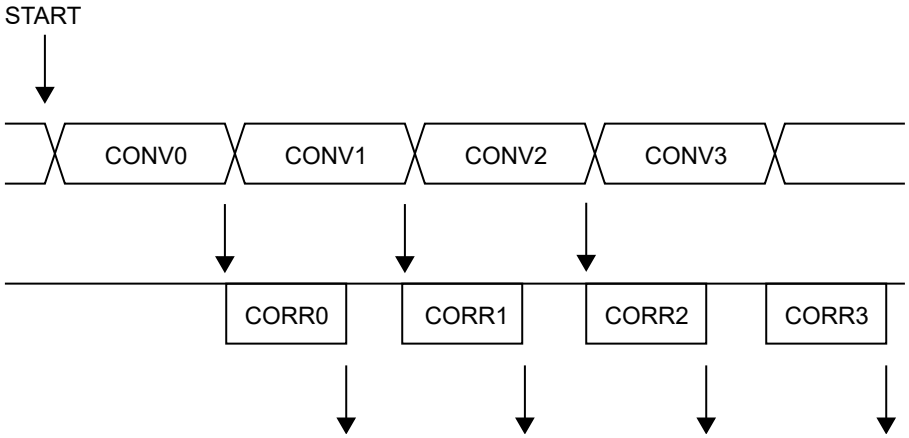
To correct these two errors, the Digital Correction Logic Enabled bit in the Control C register (CTRLC.CORREN) must be set.

Offset and gain error compensation results are both calculated according to:

$$\text{Result} = (\text{Conversion value} + - \text{OFFSETCORR}) \cdot \text{GAINCORR}$$

The correction will introduce a latency of 13 CLK_ADC clock cycles. In free running mode this latency is introduced on the first conversion only, since its duration is always less than the propagation delay. In single conversion mode this latency is introduced for each conversion.

Figure 44-8. ADC Timing Correction Enabled



44.6.2.15 Reference Buffer Compensation Offset

A hardware compensation using a reference buffer can be used.

When the REFCTRL.REFCOMP bit is set, the offset of the reference buffer is sensed during the ADC sampling phase. This offset will be then canceled during the conversion phase. This feature allows to decrease the overall gain error of the ADC.

There is also a digital gain correction (refer to Offset or Gain Correction) but contrary to that digital gain correction, the hardware compensation won't introduce any latency.

44.6.3 Additional Features

44.6.3.1 Double Buffering

The following registers are double buffered:

- Input Control (INPUTCTRL)
- Control C (CTRLC)
- Average Control (AVGCTRL)
- Sampling Time Control (SAMPCTRL)
- Window Monitor Lower Threshold (WINLT)
- Window Monitor Upper Threshold (WINUT)
- Gain Correction (GAINCORR)
- Offset Correction (OFFSETCORR)

When one of these registers is written, the data is stored in the corresponding buffer as long as the current conversion is not impacted, and the corresponding busy status will be set in the Synchronization Busy register (SYNCBUSY). When a new RESULT is available, data stored in the buffer registers will be transfered to the ADC and a new conversion can start.

44.6.4 DMA Operation

The ADC generates the following DMA request:

- Result Conversion Ready (RESRDY): the request is set when a conversion result is available and cleared when the RESULT register is read. When the averaging operation is enabled, the DMA request is set when the averaging is completed and result is available.

44.6.5 Interrupts

The ADC has the following interrupt sources:

- Result Conversion Ready: RESRDY
- Window Monitor: WINMON
- Overrun: OVERRUN

These interrupts, except the OVERRUN interrupt, are asynchronous wake-up sources. See Sleep Mode Controller for details.

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 ADC is reset. See INTFLAG register 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. Refer to Nested Vector Interrupt Controller for details. 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. Refer to Nested Vector Interrupt Controller for details.

44.6.6 Events

The ADC can generate the following output events:

- Result Ready (RESRDY): Generated when the conversion is complete and the result is available. Refer to 44.7.4. EVCTRL for details.
- Window Monitor (WINMON): Generated when the window monitor condition match. Refer to 44.7.10. CTRLC for details.

Setting an Event Output bit in the Event Control Register (EVCTRL.xxEO=1) enables the corresponding output event. Clearing this bit disables the corresponding output event. Refer to the Event System chapter for details on configuring the event system.

The ADC can take the following actions on an input event:

- Start conversion (START): Start a conversion. Refer to 44.7.17. SWTRIG for details.
- Conversion flush (FLUSH): Flush the conversion. Refer to 44.7.17. SWTRIG for details.

Setting an Event Input bit in the Event Control register (EVCTRL.xxEI=1) enables the corresponding action on input event. Clearing this bit disables the corresponding action on input event.

By default, the ADC will detect a rising edge on the incoming event. If the ADC action must be performed on the falling edge of the incoming event, the event line must be inverted first. This is done by setting the corresponding Event Invert Enable bit in Event Control register (EVCTRL.xINV=1).

Note: If several events are connected to the ADC, the enabled action will be taken on any of the incoming events. If FLUSH and START events are available at the same time, the FLUSH event has priority.

44.6.7 Sleep Mode Operation

The ADC will continue to operate in any Sleep mode where the selected source clock is running. The ADC's interrupts, except the OVERRUN interrupt, can be used to wake up the device from Sleep modes. Events connected to the event system can trigger other operations in the system without exiting Sleep modes.

The ONDEMAND and RUNSTDBY bits in the Control A register (CTRLA) control the behavior of the ADC during standby sleep mode, in cases where the ADC is enabled (CTRLA.ENABLE = 1). For further details on available options, refer to Table 44-5.

Note: When CTRLA.ONDEMAND=1, the analog block is powered-off when the conversion is complete. When a start request is detected, the system returns from sleep and starts a new conversion after the start-up time delay.

Table 44-5. ADC Sleep Behavior

CTRLA.RUNSTDBY	CTRLA.ONDEMAND	CTRLA.ENABLE	Description
x	x	0	Disabled

.....continued			
CTRLA.RUNSTDBY	CTRLA.ONDEMAND	CTRLA.ENABLE	Description
0	0	1	Run in all sleep modes except STANDBY.
0	1	1	Run in all sleep modes on request, except STANDBY.
1	0	1	Run in all sleep modes.
1	1	1	Run in all sleep modes on request.

44.6.8 Debug Operation

When the CPU is halted in debug mode the ADC will halt normal operation. The ADC can be forced to continue operation during debugging. Refer to DBGCTRL register for details.

44.6.9 Synchronization

Some registers (or bit fields within a register) require synchronization when read and/or written.

Synchronization is denoted by the "Read-Synchronized" (or "Read-Synchronized Bits”) and/or "Write-Synchronized" (or "Write-Synchronized Bits”) property in each individual register description.

For more details, refer to Register Synchronization.

44.7 Register Summary

Refer to the Registers Description section for more details on register properties and access permissions.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRLA	7:0	ONDEMAND	RUNSTDBY					ENABLE	SWRST
0x01	CTRLB	7:0						PRESCALER[2:0]		
0x02	REFCTRL	7:0	REFCOMP				REFSEL[3:0]			
0x03	EVCTRL	7:0			WINMONEO	RESRDYEO	STARTINV	FLUSHINV	STARTEI	FLUSHEI
0x04	INTENCLR	7:0						WINMON	OVERRUN	RESRDY
0x05	INTENSET	7:0						WINMON	OVERRUN	RESRDY
0x06	INTFLAG	7:0						WINMON	OVERRUN	RESRDY
0x07	SEQSTATUS	7:0	SEQBUSY			SEQSTATE[4:0]				
0x08	INPUTCTRL	15:8				MUXNEG[4:0]				
		7:0				MUXPOS[4:0]				
0x0A	CTRLC	15:8						WINMODE[2:0]		
		7:0			RESSEL[1:0]		CORREN	FREERUN	LEFTADJ	DIFFMODE
0x0C	AVGCTRL	7:0			ADJRES[2:0]		SAMPLENUM[3:0]			
0x0D	SAMPCTRL	7:0	OFFCOMP		SAMPLEN[5:0]					
0x0E	WINLT	15:8	WINLT[15:8]							
		7:0	WINLT[7:0]							
0x10	WINUT	15:8	WINUT[15:8]							
		7:0	WINUT[7:0]							
0x12	GAINCORR	15:8					GAINCORR[11:8]			
		7:0	GAINCORR[7:0]							
0x14	OFFSETCORR	15:8					OFFSETCORR[11:8]			
		7:0	OFFSETCORR[7:0]							
0x16 ... 0x17	Reserved									
0x18	SWTRIG	7:0							START	FLUSH
0x19 ... 0x1B	Reserved									
0x1C	DBGCTRL	7:0								DBGRUN
0x1D ... 0x1F	Reserved									
0x20	SYNCBUSY	15:8						SWTRIG	OFFSETCOR R	GAINCORR
		7:0	WINUT	WINLT	SAMPCTRL	AVGCTRL	CTRLC	INPUTCTRL	ENABLE	SWRST
0x22 ... 0x23	Reserved									
0x24	RESULT	15:8	RESULT[15:8]							
		7:0	RESULT[7:0]							
0x26 ... 0x27	Reserved									
0x28	SEQCTRL	31:24	SEQEN[31:24]							
		23:16	SEQEN[23:16]							
		15:8	SEQEN[15:8]							
		7:0	SEQEN[7:0]							
0x2C	CALIB	15:8						BIASREFBUF[2:0]		
		7:0						BIASCOMP[2:0]		

44.7.1 Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized Bits

Bit	7	6	5	4	3	2	1	0
	ONDEMAND	RUNSTDBY					ENABLE	SWRST
Access	R/W	R/W					R/W	R/W
Reset	0	0					0	0

Bit 7 – ONDEMAND On Demand Control
The On Demand operation mode allows the ADC to be enabled or disabled, depending on other peripheral requests. In On Demand operation mode, i.e., if the ONDEMAND bit has been previously set, the ADC will only be running when requested by a peripheral. If there is no peripheral requesting the ADC will be in a disable state.
If On Demand is disabled the ADC will always be running when enabled.
In standby sleep mode, the On Demand operation is still active if the CTRLA.RUNSTDBY bit is '1'. If CTRLA.RUNSTDBY is '0', the ADC is disabled.
This bit is not synchronized.

Value	Description
0	The ADC is always on , if enabled.
1	The ADC is enabled, when a peripheral is requesting the ADC conversion. The ADC is disabled if no peripheral is requesting it.

Bit 6 – RUNSTDBY Run in Standby
This bit controls how the ADC behaves during standby sleep mode.
This bit is not synchronized.

Value	Description
0	The ADC is halted during standby sleep mode.
1	The ADC is not stopped in standby sleep mode. If CTRLA.ONDEMAND=1, the ADC will be running when a peripheral is requesting it. If CTRLA.ONDEMAND=0, the ADC will always be running in standby sleep mode.

Bit 1 – ENABLE Enable
Note: This bit is write-synchronized: SYNCBUSY.ENABLE must be checked to ensure the CTRLA.ENABLE synchronization is complete.

Value	Description
0	The ADC is disabled.
1	The ADC 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 ADC, except DBGCTRL, to their initial state, and the ADC will be disabled.
Writing a '1' to CTRL.SWRST will always take precedence, meaning that all other writes in the same write-operation will be discarded.
Notes:

1. When the CTRLA.SWRST is written, the user must poll SYNCBUSY.SWRST bit to know when the reset operation is complete.
2. 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.

44.7.2 Control B

Name: CTRLB
Offset: 0x01
Reset: 0x00
Property: PAC Write-Protection, Enable-Protected

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

Bits 2:0 – PRESCALER[2:0] Prescaler Configuration
This field defines the ADC clock relative to the peripheral clock.

Value	Name	Description
0x0	DIV2	Peripheral clock divided by 2
0x1	DIV4	Peripheral clock divided by 4
0x2	DIV8	Peripheral clock divided by 8
0x3	DIV16	Peripheral clock divided by 16
0x4	DIV32	Peripheral clock divided by 32
0x5	DIV64	Peripheral clock divided by 64
0x6	DIV128	Peripheral clock divided by 128
0x7	DIV256	Peripheral clock divided by 256

44.7.3 Reference Control

Name: REFCTRL
Offset: 0x02
Reset: 0x00
Property: PAC Write-Protection, Enable-Protected

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

Bit 7 – REFCOMP Reference Buffer Offset Compensation Enable
The gain error can be reduced by enabling the reference buffer offset compensation. This will increase the start-up time of the reference.

Value	Description
0	Reference buffer offset compensation is disabled.
1	Reference buffer offset compensation is enabled.

Bits 3:0 – REFSEL[3:0] Reference Selection
These bits select the reference for the ADC.

Value	Name	Description
0x0	INTREF	Internal variable reference voltage, refer to the SUPC.VREF register for voltage reference value
0x1	INTVCC0	1/1.6 AVDD
0x2	INTVCC1	1/2 AVDD (only for AVDD > 2.0V)
0x3	VREFA	External reference (shared with DAC)
0x4	VREFB	External reference
0x5	INTVCC2	AVDD
0x6 – 0xF	-	Reserved

44.7.4 Event Control

Name: EVCTRL
Offset: 0x03
Reset: 0x00
Property: PAC Write-Protection, Enable-Protected

Bit	7	6	5	4	3	2	1	0
			WINMONEO	RESRDYEO	STARTINV	FLUSHINV	STARTEI	FLUSHEI
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit 5 – WINMONEO Window Monitor Event Out
This bit indicates whether the Window Monitor event output is enabled or not and an output event will be generated when the window monitor detects something.

Value	Description
0	Window Monitor event output is disabled and an event will not be generated.
1	Window Monitor event output is enabled and an event will be generated.

Bit 4 – RESRDYEO Result Ready Event Out
This bit indicates whether the Result Ready event output is enabled or not and an output event will be generated when the conversion result is available.

Value	Description
0	Result Ready event output is disabled and an event will not be generated.
1	Result Ready event output is enabled and an event will be generated.

Bit 3 – STARTINV Start Conversion Event Invert Enable

Value	Description
0	Start event input source is not inverted.
1	Start event input source is inverted.

Bit 2 – FLUSHINV Flush Event Invert Enable

Value	Description
0	Flush event input source is not inverted.
1	Flush event input source is inverted.

Bit 1 – STARTEI Start Conversion Event Input Enable

Value	Description
0	A new conversion will not be triggered on any incoming event.
1	A new conversion will be triggered on any incoming event.

Bit 0 – FLUSHEI Flush Event Input Enable

Value	Description
0	A flush and new conversion will not be triggered on any incoming event.
1	A flush and new conversion will be triggered on any incoming event.

44.7.5 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 (INTENSET) register.

Bit	7	6	5	4	3	2	1	0
						WINMON	OVERRUN	RESRDY
Access						R/W	R/W	R/W
Reset						0	0	0

Bit 2 – WINMON Window Monitor Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Window Monitor Interrupt Enable bit, which disables the corresponding interrupt request.

Value	Description
0	The window monitor interrupt is disabled.
1	The window monitor interrupt is enabled, and an interrupt request will be generated when the Window Monitor interrupt flag is set.

Bit 1 – OVERRUN Overrun Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Overrun Interrupt Enable bit, which disables the corresponding interrupt request.

Value	Description
0	The Overrun interrupt is disabled.
1	The Overrun interrupt is enabled, and an interrupt request will be generated when the Overrun interrupt flag is set.

Bit 0 – RESRDY Result Ready Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Result Ready Interrupt Enable bit, which disables the corresponding interrupt request.

Value	Description
0	The Result Ready interrupt is disabled.
1	The Result Ready interrupt is enabled, and an interrupt request will be generated when the Result Ready interrupt flag is set.

44.7.6 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 (INTENCLR) register.

Bit	7	6	5	4	3	2	1	0
						WINMON	OVERRUN	RESRDY
Access						R/W	R/W	R/W
Reset						0	0	0

Bit 2 – WINMON Window Monitor Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Window Monitor Interrupt bit, which enables the Window Monitor interrupt.

Value	Description
0	The Window Monitor interrupt is disabled.
1	The Window Monitor interrupt is enabled.

Bit 1 – OVERRUN Overrun Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Overrun Interrupt bit, which enables the Overrun interrupt.

Value	Description
0	The Overrun interrupt is disabled.
1	The Overrun interrupt is enabled.

Bit 0 – RESRDY Result Ready Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Result Ready Interrupt bit, which enables the Result Ready interrupt.

Value	Description
0	The Result Ready interrupt is disabled.
1	The Result Ready interrupt is enabled.

44.7.7 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x06
Reset: 0x00
Property: –

Bit	7	6	5	4	3	2	1	0
						WINMON	OVERRUN	RESRDY
Access						R/W	R/W	R/W
Reset						0	0	0

Bit 2 – WINMON Window Monitor
This flag is cleared by writing a '1' to the flag or by reading the RESULT register.
This flag is set on the next GCLK_ADC cycle after a match with the window monitor condition, and an interrupt request will be generated if INTENSET.WINMON is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Window Monitor interrupt flag.

Bit 1 – OVERRUN Overrun
This flag is cleared by writing a '1' to the flag.
This flag is set if RESULT is written before the previous value has been read by CPU, and an interrupt request will be generated if INTENSET.OVERRUN=1.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Overrun interrupt flag.

Bit 0 – RESRDY Result Ready
This flag is cleared by writing a '1' to the flag or by reading the RESULT register.
This flag is set when the conversion result is available, and an interrupt will be generated if INTENSET.RESRDY=1.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Result Ready interrupt flag.

44.7.8 Sequence Status

Name: SEQSTATUS
Offset: 0x07
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
	SEQBUSY			SEQSTATE[4:0]				
Access	R			R	R	R	R	R
Reset	0			0	0	0	0	0

Bit 7 – SEQBUSY Sequence busy
This bit is set when the sequence start.
This bit is clear when the last conversion in a sequence is done.

Bits 4:0 – SEQSTATE[4:0] Sequence State
These bit fields are the pointer of sequence. This value identifies the last conversion done in the sequence.

44.7.9 Input Control

Name: INPUTCTRL
Offset: 0x08
Reset: 0x0000
Property: PAC Write-Protection, Write-Synchronized

Note: This register is write-synchronized: SYNCBUSY.INPUTCTRL must be checked to ensure the INPUTCTRL register synchronization is complete.

Bit	15	14	13	12	11	10	9	8
				MUXNEG[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
				MUXPOS[4:0]				
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bits 12:8 – MUXNEG[4:0] Negative MUX Input Selection
These bits define the MUX selection for the negative ADC input.

Value	Name	Description
0x00	AIN0	ADC AIN0 pin
0x01	AIN1	ADC AIN1 pin
0x02	AIN2	ADC AIN2 pin
0x03	AIN3	ADC AIN3 pin
0x04	AIN4	ADC AIN4 pin
0x05	AIN5	ADC AIN5 pin
0x06	AIN6	ADC AIN6 pin
0x07	AIN7	ADC AIN7 pin
0x08 – 0x17	-	Reserved
0x18	AVSS	Ground
0x19-0x1F	-	Reserved

Bits 4:0 – MUXPOS[4:0] Positive MUX Input Selection
These bits define the MUX selection for the positive ADC input. If the internal bandgap voltage input channel is selected, then the Sampling Time Length bit group in the Sampling Control register must be written with a corresponding value.

Value	Name	Description
0x00	AIN0	ADC AIN0 pin
0x01	AIN1	ADC AIN1 pin
0x02	AIN2	ADC AIN2 pin
0x03	AIN3	ADC AIN3 pin
0x04	AIN4	ADC AIN4 pin
0x05	AIN5	ADC AIN5 pin
0x06	AIN6	ADC AIN6 pin
0x07	AIN7	ADC AIN7 pin
0x08	AIN8	ADC AIN8 pin
0x09	AIN9	ADC AIN9 pin
0x0A	AIN10	ADC AIN10 pin
0x0B	AIN11	ADC AIN11 pin
0x0C	AIN12	ADC AIN12 pin
0x0D	AIN13	ADC AIN13 pin
0x0E	AIN14	ADC AIN14 pin

Value	Name	Description
0x0F	AIN15	ADC AIN15 pin
0x10	AIN16	ADC AIN16 pin
0x11	AIN17	ADC AIN17 pin
0x12	AIN18	ADC AIN18 pin
0x13	AIN19	ADC AIN19 pin
0x14	AIN20	ADC AIN20 pin
0x15	AIN21	ADC AIN21 pin
0x16	AIN22	ADC AIN22 pin
0x17	AIN23	ADC AIN23 pin
0x18	-	Reserved
0x19	BANDGAP	Bandgap Voltage
0x1A	SCALEDVDDCORE	1/4 Scaled VDDCORE Supply
0x1B	SCALEDAVDD	1/4 Scaled AVDD Supply
0x1C	DAC0	DAC0 Output
0x1D	SCALEDVDD	1/4 Scaled VDD Supply
0x1E	OPAMP01	OPAMP0 or OPAMP1 Output
0x1F	OPAMP2	OPAMP2 Output

44.7.10 Control C

Name: CTRLC
Offset: 0x0A
Reset: 0x0000
Property: PAC Write-Protection, Write-Synchronized

Note: This register is write-synchronized: SYNCBUSY.CTRLC must be checked to ensure the CTRLC register synchronization is complete.

Bit	15	14	13	12	11	10	9	8
						WINMODE[2:0]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	7	6	5	4	3	2	1	0
			RESSEL[1:0]		CORREN	FREERUN	LEFTADJ	DIFFMODE
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bits 10:8 – WINMODE[2:0] Window Monitor Mode
These bits enable and define the window monitor mode.

Value	Name	Description
0x0	DISABLE	No window mode (default)
0x1	MODE1	RESULT > WINLT
0x2	MODE2	RESULT < WINUT
0x3	MODE3	WINLT < RESULT < WINUT
0x4	MODE4	WINUT < RESULT < WINLT
0x5 – 0x7	-	Reserved

Bits 5:4 – RESSEL[1:0] Conversion Result Resolution
These bits define whether the ADC completes the conversion 12-, 10- or 8-bit result resolution.

Value	Name	Description
0x0	12BIT	12-bit result
0x1	16BIT	Accumulation or Oversampling and Decimation modes
0x2	10BIT	10-bit result
0x3	8BIT	8-bit result

Bit 3 – CORREN Digital Correction Logic Enabled

Value	Description
0	Disable the digital result correction.
1	Enable the digital result correction. The ADC conversion result in the RESULT register is then corrected for gain and offset based on the values in the GAINCORR and OFFSETCORR registers. Conversion time will be increased by 13 cycles according to the value in the Offset Correction Value bit group in the Offset Correction register.

Bit 2 – FREERUN Free Running Mode

Value	Description
0	The ADC run in single conversion mode.
1	The ADC is in free running mode and a new conversion will be initiated when a previous conversion completes.

Bit 1 – LEFTADJ Left-Adjusted Result

Value	Description
0	The ADC conversion result is right-adjusted in the RESULT register.

Value	Description
1	The ADC conversion result is left-adjusted in the RESULT register. The high byte of the 12-bit result will be present in the upper part of the result register. Writing this bit to zero (default) will right-adjust the value in the RESULT register.

Bit 0 – DIFFMODE Differential Mode

Value	Description
0	The ADC is running in singled-ended mode.
1	The ADC is running in differential mode. In this mode, the voltage difference between the MUXPOS and MUXNEG inputs will be converted by the ADC.

44.7.11 Average Control

Name: AVGCTRL
Offset: 0x0C
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized

Note: This register is write-synchronized: SYNCBUSY.AVGCTRL must be checked to ensure the AVGCTRL register synchronization is complete.

Bit	7	6	5	4	3	2	1	0
		ADJRES[2:0]			SAMPLENUM[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

Bits 6:4 – ADJRES[2:0] Adjusting Result / Division Coefficient
These bits define the division coefficient in 2n steps.

Bits 3:0 – SAMPLENUM[3:0] Number of Samples to be Collected
These bits define how many samples are added together. The result will be available in the Result register (RESULT). Note: if the result width increases, CTRLC.RESSEL must be changed.

Value	Description
0x0	1 sample
0x1	2 samples
0x2	4 samples
0x3	8 samples
0x4	16 samples
0x5	32 samples
0x6	64 samples
0x7	128 samples
0x8	256 samples
0x9	512 samples
0xA	1024 samples
0xB – 0xF	Reserved

44.7.12 Sampling Time Control

Name: SAMPCTRL
Offset: 0x0D
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized

Note: This register is write-synchronized: SYNCBUSY.SAMPCTRL must be checked to ensure the SAMPCTRL register synchronization is complete.

Bit	7	6	5	4	3	2	1	0
	OFFCOMP		SAMPLEN[5:0]					
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 – OFFCOMP Comparator Offset Compensation Enable
Setting this bit enables the offset compensation for each sampling period to ensure low offset and immunity to temperature or voltage drift. ADC sampling time is fixed to 4 ADC Clock cycles when OFFCOMP = 1.

Bits 5:0 – SAMPLEN[5:0] Sampling Time Length
These bits control the ADC sample time (T_{SAMP}) in number of ADC Clock Period (CLK_ADC aka TAD), depending on the prescaler value, thus controlling the ADC input impedance. Sampling time is set according to the equation:
$T_{SAMP} = (\mathrm{SAMPLEN}+1) \cdot (TAD)$

SAMPLEN is only available when OFFCOMP = 0.
Note: Refer to ADC Electrical Characteristics for T_{SAMP} computation.

44.7.13 Window Monitor Lower Threshold

Name: WINLT
Offset: 0x0E
Reset: 0x0000
Property: PAC Write-Protection, Write-Synchronized

Note: This register is write-synchronized: SYNCBUSY.WINLT must be checked to ensure the WINLT register synchronization is complete.

Bit	15	14	13	12	11	10	9	8
	WINLT[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
	WINLT[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 – WINLT[15:0] Window Lower Threshold
If the window monitor is enabled, these bits define the lower threshold value.

44.7.14 Window Monitor Upper Threshold

Name: WINUT
Offset: 0x10
Reset: 0x0000
Property: PAV Write-Protection, Write-Synchronized

Note: This register is write-synchronized: SYNCBUSY.WINUT must be checked to ensure the WINUT register synchronization is complete.

Bit	15	14	13	12	11	10	9	8
	WINUT[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
	WINUT[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 – WINUT[15:0] Window Upper Threshold
If the window monitor is enabled, these bits define the upper threshold value.

44.7.15 Gain Correction

Name: GAINCORR
Offset: 0x12
Reset: 0x0000
Property: PAC Write-Protection, Write-Synchronized

Note: This register is write-synchronized: SYNCBUSY.GAINCORR must be checked to ensure the GAINCORR register synchronization is complete.

Bit	15	14	13	12	11	10	9	8
					GAINCORR[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
	GAINCORR[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 – GAINCORR[11:0] Gain Correction Value
If CTRLC.CORREN=1, these bits define how the ADC conversion result is compensated for gain error before being written to the result register. The gain correction is a fractional value, a 1-bit integer plus an 11-bit fraction, and therefore ½ <= GAINCORR < 2. GAINCORR values range from 0.10000000000 to 1.11111111111.

44.7.16 Offset Correction

Name: OFFSETCORR
Offset: 0x14
Reset: 0x0000
Property: PAC Write-Protection, Write-Synchronized

Note: This register is write-synchronized: SYNCBUSY.OFFSETCORR must be checked to ensure the OFFSETCORR register synchronization is complete.

Bit	15	14	13	12	11	10	9	8
					OFFSETCORR[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
	OFFSETCORR[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 – OFFSETCORR[11:0] Offset Correction Value
If CTRLC.CORREN=1, these bits define how the ADC conversion result is compensated for offset error before being written to the Result register. This OFFSETCORR value is in two's complement format.

44.7.17 Software Trigger

Name: SWTRIG
Offset: 0x18
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized

Note: This register is write-synchronized: SYNCBUSY.SWTRIG must be checked to ensure the SWTRIG register synchronization is complete.

Bit	7	6	5	4	3	2	1	0
							START	FLUSH
Access							W	W
Reset							0	0

Bit 1 – START ADC Start Conversion
Writing a '1' to this bit will start a conversion or sequence. The bit is cleared by hardware when the conversion has started. Writing a '1' to this bit when it is already set has no effect.
Writing a '0' to this bit will have no effect.

Bit 0 – FLUSH ADC Conversion Flush
Writing a '1' to this bit will flush the ADC pipeline. A flush will restart the ADC clock on the next peripheral clock edge, and all conversions in progress will be aborted and lost. This bit is cleared until the ADC has been flushed.
After the flush, the ADC will resume where it left off; i.e., if a conversion was pending, the ADC will start a new conversion.
Writing this bit to '0' will have no effect.

44.7.18 Debug Control

Name: DBGCTRL
Offset: 0x1C
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.
This bit should be written only while a conversion is not ongoing.

Value	Description
0	The ADC is halted when the CPU is halted by an external debugger.
1	The ADC continues normal operation when the CPU is halted by an external debugger.

44.7.19 Synchronization Busy

Name: SYNCBUSY
Offset: 0x20
Reset: 0x0000
Property: -

Bit	15	14	13	12	11	10	9	8
						SWTRIG	OFFSETCORR	GAINCORR
Access						R	R	R
Reset						0	0	0

Bit	7	6	5	4	3	2	1	0
	WINUT	WINLT	SAMPCTRL	AVGCTRL	CTRLC	INPUTCTRL	ENABLE	SWRST
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 10 – SWTRIG Software Trigger Synchronization Busy
This bit is cleared when the synchronization of SWTRIG register between the clock domains is complete.
This bit is set when the synchronization of SWTRIG register between clock domains is started.

Bit 9 – OFFSETCORR Offset Correction Synchronization Busy
This bit is cleared when the synchronization of OFFSETCORR register between the clock domains is complete.
This bit is set when the synchronization of OFFSETCORR register between clock domains is started.

Bit 8 – GAINCORR Gain Correction Synchronization Busy
This bit is cleared when the synchronization of GAINCORR register between the clock domains is complete.
This bit is set when the synchronization of GAINCORR register between clock domains is started.

Bit 7 – WINUT Window Monitor Lower Threshold Synchronization Busy
This bit is cleared when the synchronization of WINUT register between the clock domains is complete.
This bit is set when the synchronization of WINUT register between clock domains is started.

Bit 6 – WINLT Window Monitor Upper Threshold Synchronization Busy
This bit is cleared when the synchronization of WINLT register between the clock domains is complete.
This bit is set when the synchronization of WINLT register between clock domains is started.

Bit 5 – SAMPCTRL Sampling Time Control Synchronization Busy
This bit is cleared when the synchronization of SAMPCTRL register between the clock domains is complete.
This bit is set when the synchronization of SAMPCTRL register between clock domains is started.

Bit 4 – AVGCTRL Average Control Synchronization Busy
This bit is cleared when the synchronization of AVGCTRL register between the clock domains is complete.
This bit is set when the synchronization of AVGCTRL register between clock domains is started.

Bit 3 – CTRLC Control C Synchronization Busy
This bit is cleared when the synchronization of CTRLC register between the clock domains is complete.
This bit is set when the synchronization of CTRLC register between clock domains is started.

Bit 2 – INPUTCTRL Input Control Synchronization Busy
This bit is cleared when the synchronization of INPUTCTRL register between the clock domains is complete.
This bit is set when the synchronization of INPUTCTRL register between clock domains is started.

Bit 1 – ENABLE ENABLE Synchronization Busy
This bit is cleared when the synchronization of ENABLE register between the clock domains is complete.
This bit is set when the synchronization of ENABLE register between clock domains is started.

Bit 0 – SWRST SWRST Synchronization Busy

This bit is cleared when the synchronization of SWRST register between the clock domains is complete.
This bit is set when the synchronization of SWRST register between clock domains is started

44.7.20 Result

Name: RESULT
Offset: 0x24
Reset: 0x0000
Property: -

Bit	15	14	13	12	11	10	9	8
	RESULT[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
	RESULT[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – RESULT[15:0] Result Conversion Value

These bits will hold up to a 16-bit ADC conversion result, depending on the configuration.
In single conversion mode without averaging, the ADC conversion will produce a 12-bit result, which can be left- or right-shifted, depending on the setting of CTRLC.LEFTADJ.
If the result is left-adjusted (CTRLC.LEFTADJ), the high byte of the result will be in bit position [15:8], while the remaining 4 bits of the result will be placed in bit locations [7:4]. This can be used only if an 8-bit result is needed; i.e., one can read only the high byte of the entire 16-bit register.
If the result is not left-adjusted (CTRLC.LEFTADJ) and no oversampling is used, the result will be available in bit locations [11:0], and the result is then 12 bits long. If oversampling is used, the result will be located in bit locations [15:0], depending on the settings of the Average Control register.

44.7.21 Sequence Control

Name: SEQCTRL
Offset: 0x28
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
	SEQEN[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
	SEQEN[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
	SEQEN[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
	SEQEN[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 – SEQEN[31:0] Enable Positive Input in the Sequence
For details on available positive mux selection, refer to 44.7.9. INPUTCTRLMUXNEG.
The sequence start from the lowest input, and go to the next enabled input automatically when the conversion is done. If no bits are set the sequence is disabled.

Value	Description
0	Disable the positive input mux bit selection from the sequence.
1	Enable the positive input mux bit selection to the sequence.

44.7.22 Calibration

Name: CALIB
Offset: 0x2C
Reset: 0x0000
Property: PAC Write-Protection, Enable-Protected

Bit	15	14	13	12	11	10	9	8
						BIASREFBUF[2:0]		
Access						R/W	R/W	R/W
Reset						0	0	0

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

Bits 10:8 – BIASREFBUF[2:0] Bias Reference Buffer Scaling
This value from production test must be loaded from the NVM Software Calibration Row into the CALIB register by software to allow the conversion and achieve the specified accuracy.
The value must be copied only, and must not be changed.

Bits 2:0 – BIASCOMP[2:0] Bias Comparator Scaling
This value from production test must be loaded from the NVM Software Calibration Row into the CALIB register by software to allow the conversion and achieve the specified accuracy.
The value must be copied only, and must not be changed

45. Analog Comparators (AC)

45.1 Overview

The Analog Comparator (AC) supports multiple individual comparators. Each comparator (COMP) compares the voltage levels on two inputs, and provides a digital output based on this comparison. Each comparator may be configured to generate interrupt requests and/or peripheral events upon several different combinations of input change.

Hysteresis and propagation delay can be adjusted to achieve the optimal operation for each application.

The input selection includes 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 or two pairs of comparators . These are called Comparator 0 (COMP0) and Comparator 1 (COMP1) for the first pair and Comparator 2 (COMP2) and Comparator 3 (COMP3) for the second pair. They have identical behaviors, but separate control registers. Each pair can be set in window mode to compare a signal to a voltage range instead of a single voltage level.

45.2 Features

- Four individual comparators
- Selectable propagation delay versus current consumption
- Selectable hysteresis: 4-level On, or Off
- Analog comparator outputs available on pins
  - Asynchronous or synchronous
- Flexible input selection:
  - Four pins selectable for positive or negative inputs
  - Ground (for zero crossing)
  - Bandgap reference voltage
  - 64-level programmable AVDD scaler per comparator
  - DAC0
  - OPAMP2
- 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

45.3 Block Diagram

Figure 45-1. Analog Comparator Block Diagram (First Pair)

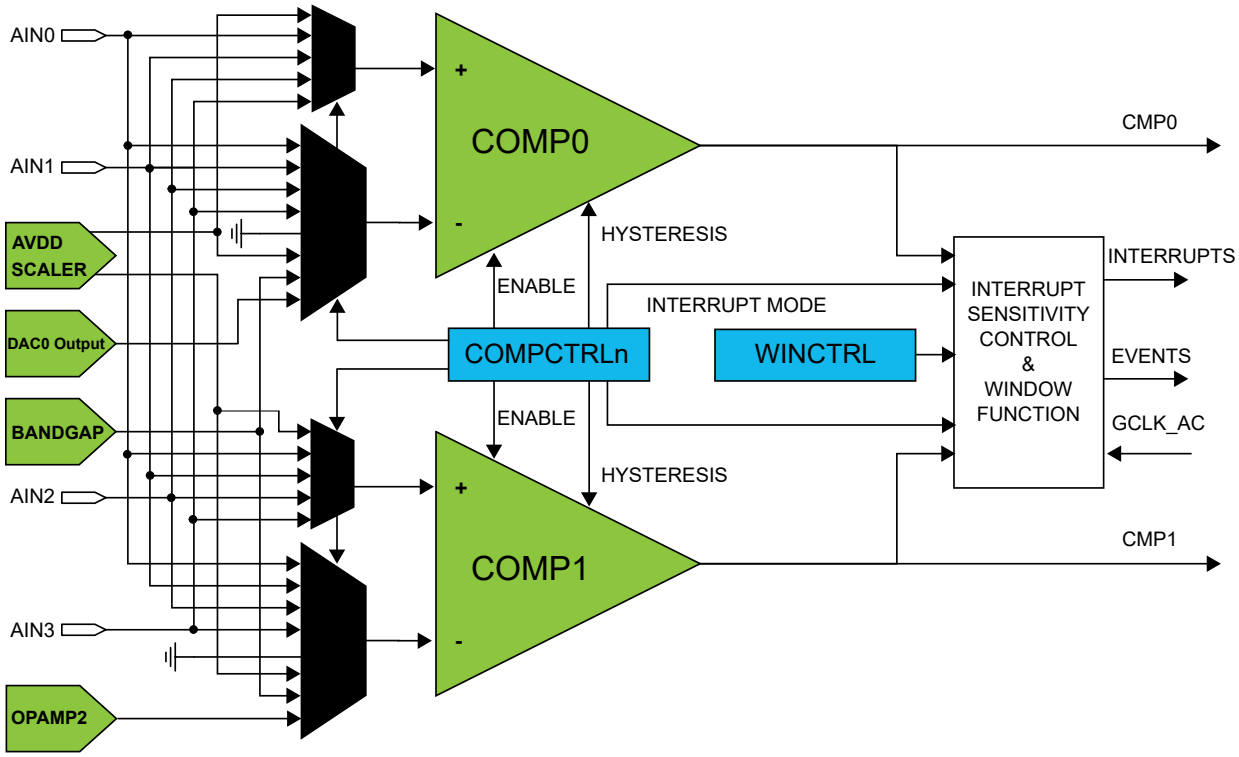
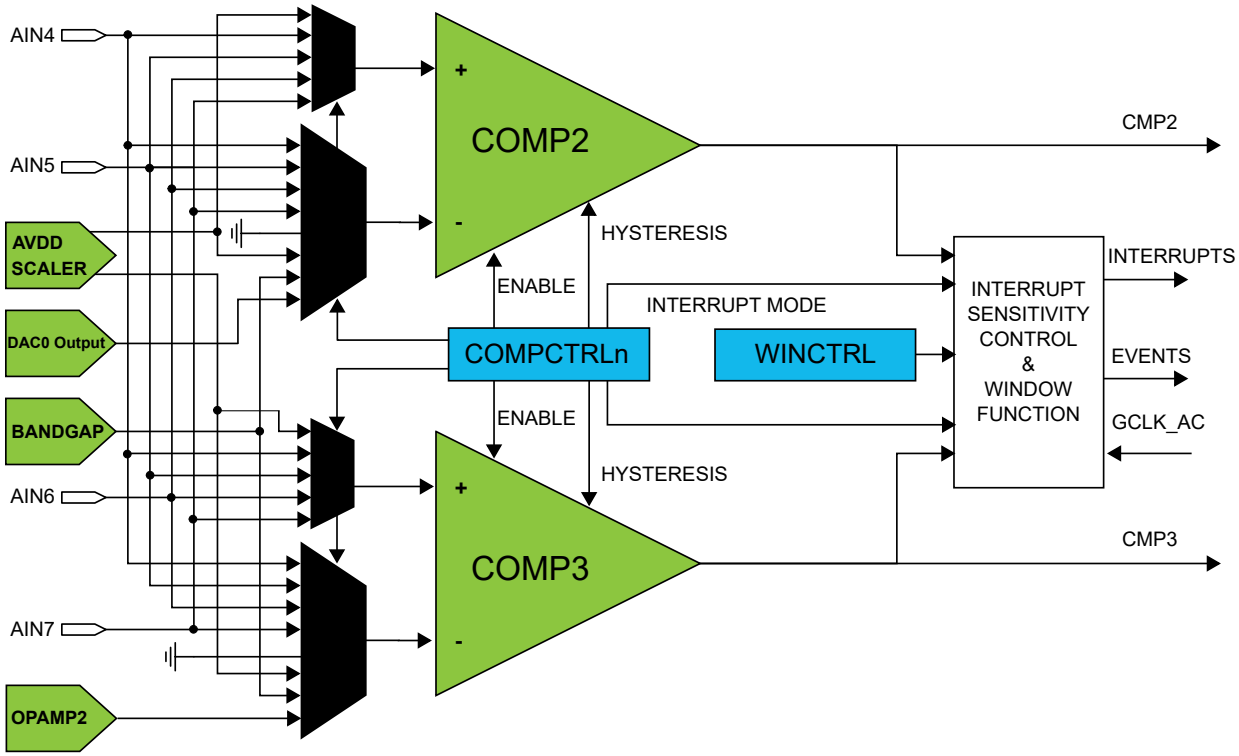


Figure 45-2. Analog Comparator Block Diagram (Second Pair)



45.4 Signal Description

Signal	Description	Type
AIN[7..0]	Analog input	Comparator inputs
CMP[3..0]	Digital output	Comparator outputs

Refer to the Pinout for details on the pin mapping for this peripheral. One signal can be mapped on several pins.

45.5 Peripheral Dependencies

Table 45-1. AC Configuration Summary

Peripheral name	Base address	NVIC IRQ Index	MCLK AHB/APB Clocks	GCLK Peripheral Channel Index (GCLK.PCHCTRL)	PAC Peripheral Identifier Index (PAC.WRCTRL)	Events (EVSYS)		Power Domain (PM.STDBYCFG)
						Users (EVSYS.USER)	Generators (EVSYS.CHANNEL)	
AC	0x40003400	64: COMP0-3, WIN0-1	CLK_AC_APB	29: GCLK_AC	13	40-43: COMP0-3	76-79: COMP0-3 80-81: WIN0-1	PDAO

Each comparator has up to four I/O pins that can be used as analog inputs. Each 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, OPAMP2, or DAC must be configured and enabled prior to its use as a comparator input.

The analog signals of AC, ADC, DAC and OPAMP can be interconnected. The AC and ADC peripheral can request the OPAMP using an analog ONDEMAND functionality.

See Analog Connections of Peripherals for details.

45.6 Functional Description

45.6.1 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 a 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).

45.6.2 Basic Operation

45.6.2.1 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.

45.6.2.2 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 writing a '0' to CTRLA.ENABLE.

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 will be reset to their initial state, and the AC will be disabled. Refer to CTRLA for details.

45.6.2.3 Comparator Configuration

Each individual comparator must be configured by its respective Comparator Control register (COMPCTRLn) before that comparator is enabled. These settings cannot be changed while the comparator is enabled.

- Select the desired measurement mode with COMPCTRLn.SINGLE. See 45.6.2.4. Starting a Comparison for more details.
- Select the desired hysteresis with COMPCTRLn.HYSTEN and COMPCTRLn.HYST. See 45.6.6. Input Hysteresis for more details.
- Select the comparator speed versus power with COMPCTRLn.SPEED. See 45.6.7. Propagation Delay vs. Power Consumption for more details.
- Select the interrupt source with COMPCTRLn.INTSEL.
- Select the positive and negative input sources with the COMPCTRLn.MUXPOS and COMPCTRLn.MUXNEG bits. See 45.6.3. Selecting Comparator Inputs for more details.
- Select the filtering option with COMPCTRLn.FLEN.
- Select standby operation with Run in Standby bit (COMPCTRLn.RUNSTDBY).

The individual comparators are enabled by writing a '1' to the Enable bit in the Comparator x Control registers (COMPCTRLn.ENABLE). The individual comparators are disabled by writing a '0' to COMPCTRLn.ENABLE. Writing a '0' to CTRLA.ENABLE will also disable all the comparators, but will not clear their COMPCTRLn.ENABLE bits.

45.6.2.4 Starting a Comparison

Each comparator channel can be in one of two different measurement modes, determined by the Single bit in the Comparator x Control register (COMPCTRLn.SINGLE):

- 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. 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 (COMPCTRLn.INTSEL). Events are generated using the comparator output state, regardless of whether the interrupt is enabled or not.

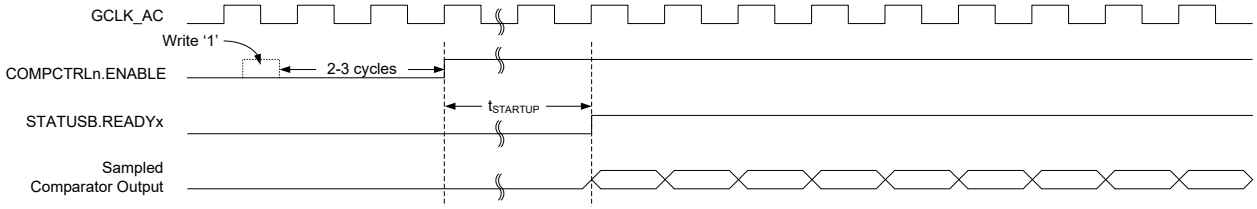
45.6.2.4.1 Continuous Measurement

Continuous measurement is selected by writing COMPCTRLn.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 COMPCTRLn.ENABLE bit is written to zero. The start-up time applies only to the first comparison.

In 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 45-3. Continuous Measurement Example



For low-power operation, comparisons can be performed during sleep modes without a clock. The comparator is enabled continuously, and changes of the comparator state are detected asynchronously. When a toggle occurs, the Power Manager 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.

45.6.2.4.2 Single-Shot

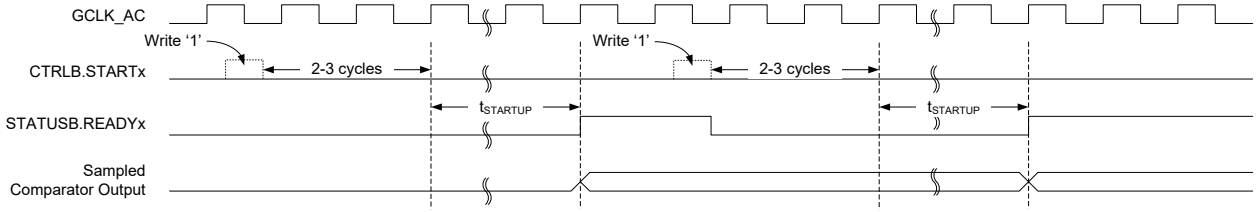
Single-shot operation is selected by writing COMPCTRLn.SINGLE to '1'. During single-shot operation, the comparator is normally idle. The user starts a single comparison by writing '1' to the respective Start Comparison bit in the write-only Control B register (CTRLB.STARTx). The comparator is enabled, 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 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 45-4. Single-Shot Example



For low-power operation, event-triggered measurements can be performed during sleep modes. When the event occurs, the Power Manager will start GCLK_AC. The comparator is enabled, and after the startup 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.

45.6.3 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 several 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 (COMPCTRLn.MUXPOS)
- The negative input is selected by the Negative Input MUX Select bit group in the Comparator Control register (COMPCTRLn.MUXNEG)

In the case of using an external I/O pin, the selected pin must be configured for analog use in the PORT Controller 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.

45.6.4 Window Operation

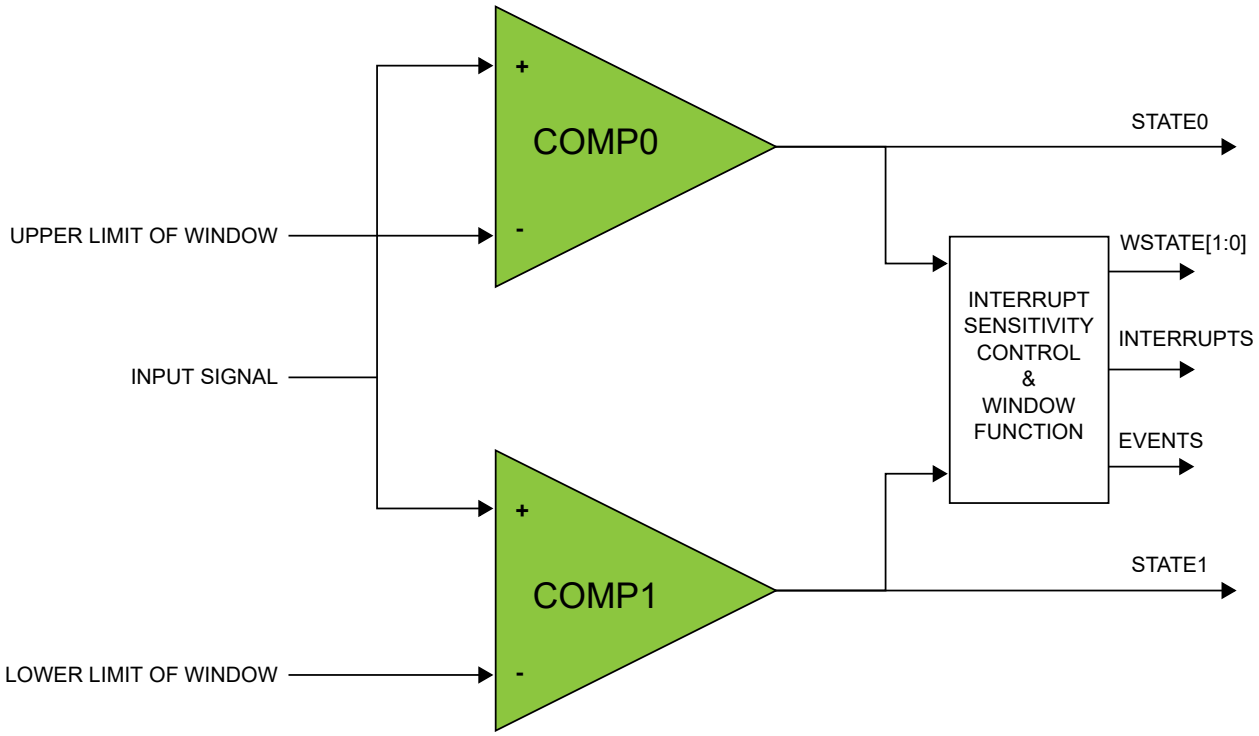
Each comparator pair can be configured to work together in window mode. In this mode, a voltage range is defined, and the comparators give information about whether an input signal is within this range or not. Window mode is enabled by the Window Enable x bit in the Window Control register (WINCTRL.WENx). Both comparators in a pair must have the same measurement mode setting in their respective Comparator Control Registers (COMPCTRLn.SINGLE).

To physically configure the pair of 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 Comparators in Window Mode, 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.

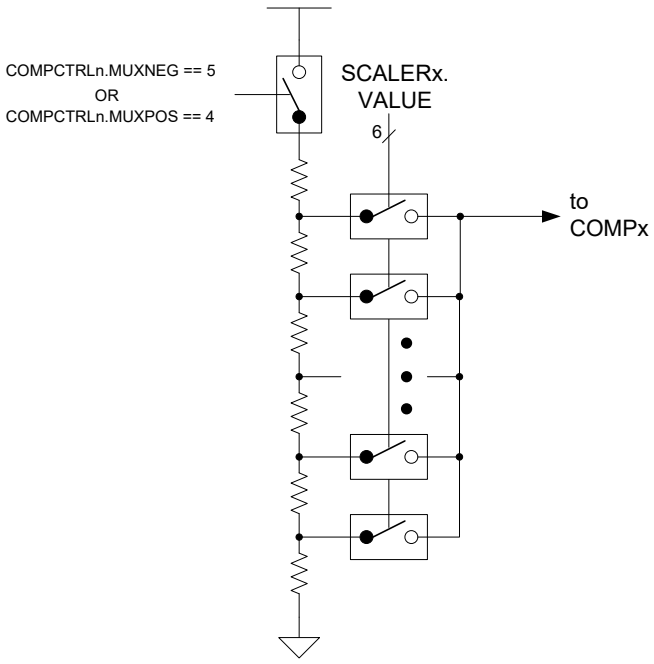
Figure 45-5. Comparators in Window Mode



45.6.5 AVDD Scaler

The AVDD scaler generates a reference voltage that is a fraction of the device's supply voltage, with 64 levels. One independent voltage channel is dedicated for each comparator. 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 (COMPCTRLn.MUXNEG, or COMPCTRLn.MUXPOS) is set to 0x5 for Negative Input or 0x04 for Positive Input and the comparator is enabled. The voltage of each channel is selected by the Value bit field in the SCALERx registers (SCALERx.VALUE).

Figure 45-6. AVDD Scaler



45.6.6 Input Hysteresis

Application software can selectively enable/disable hysteresis for the comparison. Applying hysteresis will help prevent constant toggling of the output, which can be caused by noise when the input signals are close to each other.

Hysteresis is enabled for each comparator individually by the Hysteresis Enable bit in the Comparator x Control register (COMPCTRLn.HYSTEN). Furthermore, when enabled, the level of hysteresis is programmable through the Hysteresis Level bits also in the Comparator x Control register (COMPCTRLn.HYST). Hysteresis is available only in continuous mode (COMPCTRLn.SINGLE=0).

45.6.7 Propagation Delay vs. Power Consumption

It is possible to trade off comparison speed for power efficiency to get the shortest possible propagation delay or the lowest power consumption. The speed setting is configured for each comparator individually by the Speed bit group in the Comparator n Control register (COMPCTRLn.SPEED). The Speed bits select the amount of bias current provided to the comparator, and as such will also affect the start-up time.

45.6.8 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 (COMPCTRLn.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 the following figure. For single-shot mode, the comparison completes after the Nth filter sample, as shown in *Single-Shot Filtering*.

Figure 45-7. Continuous Mode Filtering

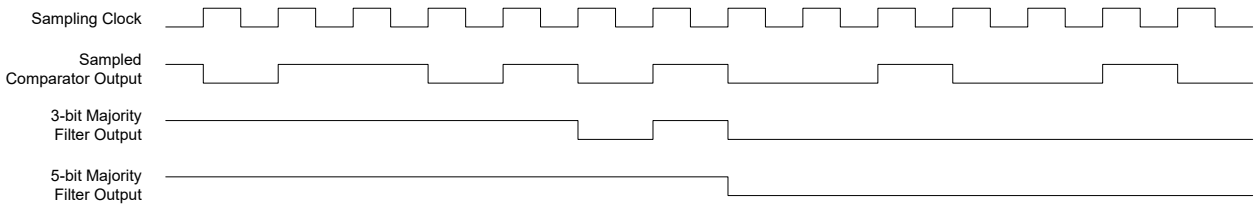
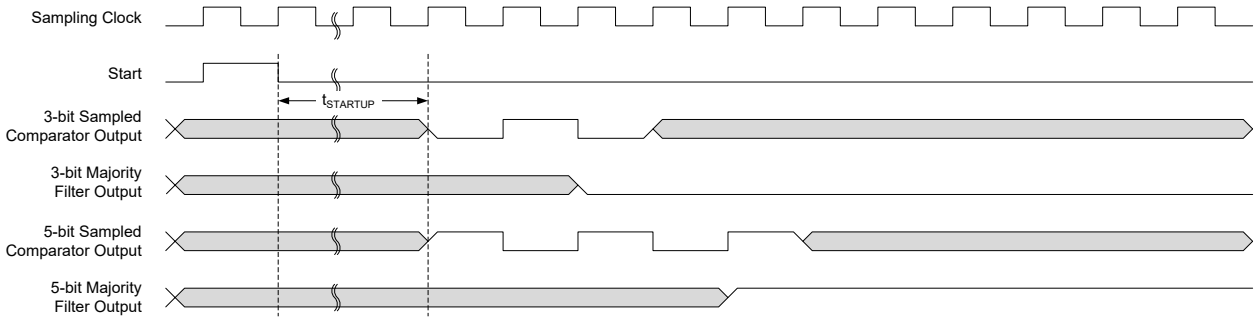


Figure 45-8. Single-Shot Filtering



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.

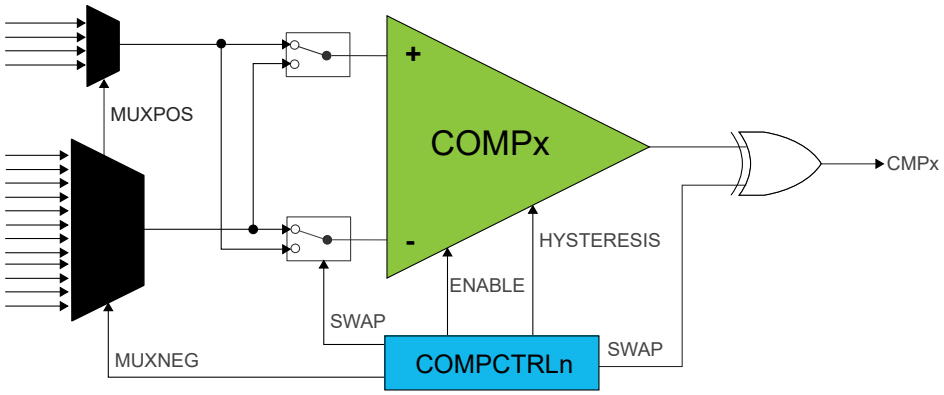
45.6.9 Comparator Output

The output of each comparator can be routed to an I/O pin by setting the Output bit group in the Comparator Control n register (COMPCTRLn.OUT). This allows the comparator to be used by external circuitry. Either the raw, non-synchronized output of the comparator or the CLK_AC-synchronized version, including filtering, can be used as the I/O signal source. The output appears on the corresponding CMP[x] pin.

45.6.10 Offset Compensation

The Swap bit in the Comparator Control registers (COMPCTRLn.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 the following figure. This allows the user to measure or compensate for the comparator input offset voltage. As part of the input selection, COMPCTRLn.SWAP can be changed only while the comparator is disabled.

Figure 45-9. Input Swapping for Offset Compensation



45.6.11 Interrupts

The AC has the following interrupt sources:

- Comparator (COMP0, COMP1, COMP2, COMP3): Indicates a change in comparator status.
- Window (WIN0, WIN1): 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 (COMPCTRLn.INTSEL). Window interrupts are generated based on the conditions selected by the Window Interrupt Selection bit group in the Window Control register (WINCTRL.WINTSELx[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 INFLAG register 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.

Note that interrupts must be globally enabled for interrupt requests to be generated.

45.6.12 Events

The AC can generate the following output events:

- Comparator (COMP0, COMP1, COMP2, COMP3): Generated as a copy of the comparator status
- Window (WIN0, WIN1): 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. Refer to the Event System chapter for details on configuring the event system.

The AC can take the following action on an input event (COMP0-3):

- Start comparison

Writing a one to an Event Input bit into the Event Control register (EVCTRL.COMPEIx) enables the corresponding action on input event. Writing a zero to this bit disables the corresponding action on input event. Note that if several events are connected to the AC, the enabled action will be taken on any of the incoming events. Refer to the Event System chapter for details on configuring 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.

45.6.13 Sleep Mode Operation

The Run in Standby bits in the Comparator x Control registers (COMPCTRLn.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 the following table.

Table 45-2. Sleep Mode Operation

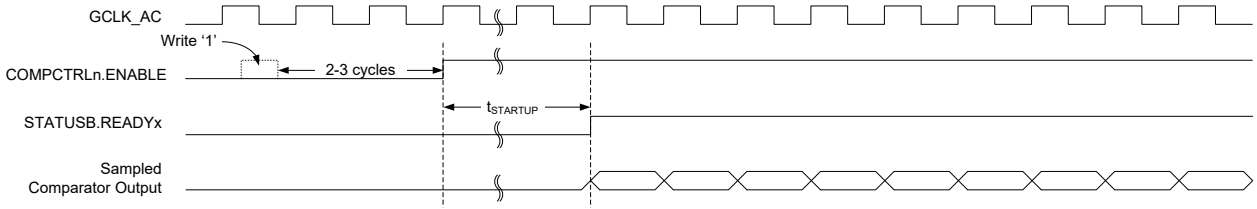
COMPCTRLn.MODE	RUNSTDBY=0	RUNSTDBY=1
0 (Continuous)	COMPx disabled	GCLK_AC stopped, COMPx enabled
1 (Single-shot)	COMPx disabled	GCLK_AC stopped, COMPx enabled only when triggered by an input event

45.6.13.1 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.

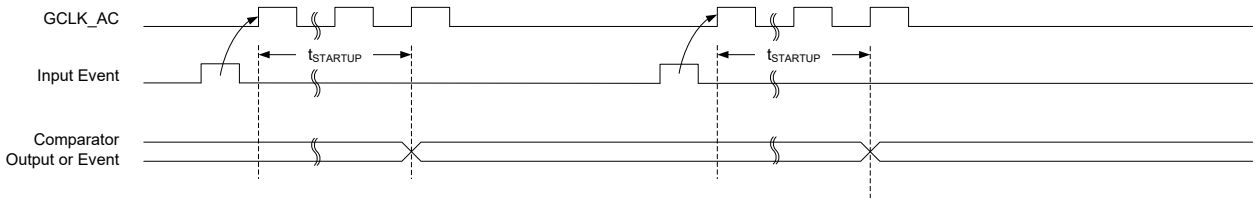
Figure 45-10. Continuous Mode SleepWalking



45.6.13.2 Single-Shot Measurement during Sleep

For low-power operation, event-triggered measurements can be performed during sleep modes. When the event occurs, the Power Manager 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 Figure 45-11. 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 45-11. Single-Shot SleepWalking



45.6.14 Debug Operation

When the CPU is halted in debug mode, the AC will halt normal operation after any on-going comparison is completed. The AC can be forced to continue normal operation during debugging. Refer to DBGCTRL for details. 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.

45.6.15 Synchronization

Some registers (or bit fields within a register) require synchronization when read and/or written.

Synchronization is denoted by the "Read-Synchronized" (or "Read-Synchronized Bits") and/or "Write-Synchronized" (or "Write-Synchronized Bits") property in each individual register description.

For more details, refer to Register Synchronization.

45.7 Register Summary

Refer to the Registers Description section for more details on register properties and access permissions.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRLA	7:0							ENABLE	SWRST
0x01	CTRLB	7:0					START3	START2	START1	START0
0x02	EVCTRL	15:8	INVEI3	INVEI2	INVEI1	INVEI0	COMPEI3	COMPEI2	COMPEI1	COMPEI0
		7:0			WINEO1	WINEO0	COMPEO3	COMPEO2	COMPEO1	COMPEO0
0x04	INTENCLR	7:0			WIN1	WIN0	COMP3	COMP2	COMP1	COMP0
0x05	INTENSET	7:0			WIN1	WIN0	COMP3	COMP2	COMP1	COMP0
0x06	INTFLAG	7:0			WIN1	WIN0	COMP3	COMP2	COMP1	COMP0
0x07	STATUSA	7:0	WSTATE1[1:0]		WSTATE0[1:0]		STATE3	STATE2	STATE1	STATE0
0x08	STATUSB	7:0					READY3	READY2	READY1	READY0
0x09	DBGCTRL	7:0								DBGRUN
0x0A	WINCTRL	7:0		WINTSEL1[1:0]		WEN1		WINTSEL0[1:0]		WEN0
0x0B	Reserved									
0x0C	SCALER0	7:0			VALUE[5:0]					
0x0D	SCALER1	7:0			VALUE[5:0]					
0x0E	SCALER2	7:0			VALUE[5:0]					
0x0F	SCALER3	7:0			VALUE[5:0]					
0x10	COMPCTRL0	31:24			OUT[1:0]			FLEN[2:0]		
		23:16			HYST[1:0]		HYSTEN		SPEED[1:0]	
		15:8	SWAP		MUXPOS[2:0]			MUXNEG[2:0]		
		7:0		RUNSTDBY		INTSEL[1:0]		SINGLE	ENABLE	
0x14	COMPCTRL1	31:24			OUT[1:0]			FLEN[2:0]		
		23:16			HYST[1:0]		HYSTEN		SPEED[1:0]	
		15:8	SWAP		MUXPOS[2:0]			MUXNEG[2:0]		
		7:0		RUNSTDBY		INTSEL[1:0]		SINGLE	ENABLE	
0x18	COMPCTRL2	31:24			OUT[1:0]			FLEN[2:0]		
		23:16			HYST[1:0]		HYSTEN		SPEED[1:0]	
		15:8	SWAP		MUXPOS[2:0]			MUXNEG[2:0]		
		7:0		RUNSTDBY		INTSEL[1:0]		SINGLE	ENABLE	
0x1C	COMPCTRL3	31:24			OUT[1:0]			FLEN[2:0]		
		23:16			HYST[1:0]		HYSTEN		SPEED[1:0]	
		15:8	SWAP		MUXPOS[2:0]			MUXNEG[2:0]		
		7:0		RUNSTDBY		INTSEL[1:0]		SINGLE	ENABLE	
0x20	SYNCBUSY	31:24								
		23:16								
		15:8								
		7:0		COMPCTRL3	COMPCTRL2	COMPCTRL1	COMPCTRL0	WINCTRL	ENABLE	SWRST

45.7.1 Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized Bits

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

Bit 1 – ENABLE Enable

Note: This bit is write-synchronized: SYNCBUSY.ENABLE must be checked to ensure the CTRLA.ENABLE synchronization is complete.

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 will be disabled.
Writing a '1' to CTRLA.SWRST will always take precedence, meaning that all other writes in the same write-operation will be discarded.

Notes:

1. When the CTRLA.SWRST is written, the user must poll SYNCBUSY.SWRST bit to know when the reset operation is complete.
2. 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.

45.7.2 Control B

Name: CTRLB
Offset: 0x01
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
					START3	START2	START1	START0
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bits 0, 1, 2, 3 – START 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' (COMPCTRLn.SINGLE and COMPCTRLn.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 zero.

45.7.3 Event Control

Name: EVCTRL
Offset: 0x02
Reset: 0x0000
Property: PAC Write-Protection, Enable-Protected

Bit	15	14	13	12	11	10	9	8
	INVEI3	INVEI2	INVEI1	INVEI0	COMPEI3	COMPEI2	COMPEI1	COMPEI0
Access	R/W	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
			WINEO1	WINEO0	COMPEO3	COMPEO2	COMPEO1	COMPEO0
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bits 12, 13, 14, 15 – INVEI 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, 10, 11 – COMPEI 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.

Bits 4, 5 – WINEO Window x Event Output Enable

These bits indicate whether the window x function can generate a peripheral event or not.

Value	Description
0	Window x Event is disabled.
1	Window x Event is enabled.

Bits 0, 1, 2, 3 – COMPEO 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.

45.7.4 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
			WIN1	WIN0	COMP3	COMP2	COMP1	COMP0
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bits 4, 5 – WIN Window x Interrupt Enable
Reading this bit returns the state of the Window x interrupt enable.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit disables the Window x interrupt.

Value	Description
0	The Window x interrupt is disabled.
1	The Window x interrupt is enabled.

Bits 0, 1, 2, 3 – COMP 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.

45.7.5 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
			WIN1	WIN0	COMP3	COMP2	COMP1	COMP0
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bits 4, 5 – WIN Window x Interrupt Enable
Reading this bit returns the state of the Window x interrupt enable.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit enables the Window x interrupt.

Value	Description
0	The Window x interrupt is disabled.
1	The Window x interrupt is enabled.

Bits 0, 1, 2, 3 – COMP 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 will set 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.

45.7.6 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x06
Reset: 0x00
Property: –

Bit	7	6	5	4	3	2	1	0
			WIN1	WIN0	COMP3	COMP2	COMP1	COMP0
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bits 4, 5 – WIN Window x
This flag is set according to the Window x Interrupt Selection bit group in the 45.7.10. WINCTRL register (WINCTRL.WINTSELx) and will generate an interrupt if INTENSET.WINx is also one.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Window x interrupt flag.

Bits 0, 1, 2, 3 – COMP Comparator x
Reading this bit returns the status of the Comparator x interrupt flag. If comparator x is not implemented, COMPx always reads as zero.
This flag is set according to the Interrupt Selection bit group in the Comparator x Control register (COMPCTRLn.INTSEL) and will generate an interrupt if INTENSET.COMPx is also one.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Comparator x interrupt flag.

45.7.7 Status A

Name: STATUSA
Offset: 0x07
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
	WSTATE1[1:0]		WSTATE0[1:0]		STATE3	STATE2	STATE1	STATE0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 7:6 – WSTATE1[1:0] Window 1 Current State
These bits show the current state of the signal if the window 1 mode is enabled.

Value	Name	Description
0x0	ABOVE	Signal is above window
0x1	INSIDE	Signal is inside window
0x2	BELOW	Signal is below window
0x3		Reserved

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.

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, 2, 3 – STATE Comparator x Current State
This bit shows the current state of the output signal from COMPx. STATEx is valid only when STATUSB.READYx is one.

45.7.8 Status B

Name: STATUSB
Offset: 0x08
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
					READY3	READY2	READY1	READY0
Access					R	R	R	R
Reset					0	0	0	0

Bits 0, 1, 2, 3 – READY Comparator x Ready
This bit is cleared when the comparator x output is not ready.
This bit is set when the comparator x output is ready.
If comparator x is not implemented, READYx always reads as zero.

45.7.9 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 will complete.
1	The AC continues normal operation when the CPU is halted by an external debugger.

45.7.10 Window Control

Name: WINCTRL
Offset: 0x0A
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized

Note: This register is write-synchronized: SYNCBUSY.WINCTRL must be checked to ensure the WINCTRL register synchronization is complete.

Bit	7	6	5	4	3	2	1	0
		WINTSEL1[1:0]		WEN1		WINTSEL0[1:0]		WEN0
Access		R/W	R/W	R/W		R/W	R/W	R/W
Reset		0	0	0		0	0	0

Bits 6:5 – WINTSEL1[1:0] Window 1 Interrupt Selection
These bits configure the interrupt mode for the comparator window 1 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 4 – WEN1 Window 1 Mode Enable

Value	Description
0	Window mode is disabled for comparators 2 and 3.
1	Window mode is enabled for comparators 2 and 3.

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 for comparators 0 and 1.
1	Window mode is enabled for comparators 0 and 1.

45.7.11 Scaler n

Name: SCALER
Offset: 0x0C + n*0x01 [n=0..3]
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
			VALUE[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 – VALUE[5:0] Scaler Value
These bits define the scaling factor for channel n of the AVDD voltage scaler. The output voltage, V_{SCALE}, is:

$$V_{\mathrm{SCALE}} = \frac{AV_{\mathrm{DD}} \cdot (\mathrm{VALUE}+1)}{64}$$

45.7.12 Comparator Control n

Name: COMPCTRL
Offset: 0x10 + n*0x04 [n=0..3]
Reset: 0x00000000
Property: PAC Write-Protection, Write-Synchronized

Note: This register is write-synchronized: SYNCBUSY.COMPCTRLn must be checked to ensure the COMPCTRLn register synchronization is complete.

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
			HYST[1:0]		HYSTEN		SPEED[1:0]	
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
	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 zero.
Note: For internal use of the comparison results by the CCL, this bit 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 zero.
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

Bits 21:20 – HYST[1:0] Hysteresis Level
These bits indicate the hysteresis level of comparator n when hysteresis is enabled (COMPCTRLn.HYSTEN=1). Hysteresis is available only for continuous mode (COMPCTRLn.SINGLE=0). COMPCTRLn.HYST can be written only while COMPCTRLn.ENABLE is zero.
These bits are not synchronized.

Value	Name	Description
0x0	HYST50	50mV
0x1	HYST70	70mV
0x2	HYST90	90mV
0x3	HYST110	110mV

Bit 19 – HYSTEN Hysteresis Enable

This bit indicates the hysteresis mode of comparator n. Hysteresis is available only for continuous mode (COMPCTRLn.SINGLE=0). COMPCTRLn.HYST can be written only while COMPCTRLn.ENABLE is zero.
This bit is not synchronized.

Value	Description
0	Hysteresis is disabled.
1	Hysteresis is enabled.

Bits 17:16 – SPEED[1:0] Speed Selection

This bit indicates the speed/propagation delay mode of comparator n. COMPCTRLn.SPEED can be written only while COMPCTRLn.ENABLE is zero.
These bits are not synchronized.

Value	Name	Description
0x0	LOW	Low speed
0x1	MEDLOW	Medium low speed
0x2	MEDHIGH	Medium high speed
0x3	HIGH	High speed

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 zero.
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 will be connected to the positive input of comparator n. COMPCTRLn.MUXPOS can be written only while COMPCTRLn.ENABLE is zero.
These bits are not synchronized.

Value	Name (COMP0/1)	Name (COMP2/3)	Description
0x0	AIN0	AIN4	AC Input
0x1	AIN1	AIN5	AC Input
0x2	AIN2	AIN6	AC Input
0x3	AIN3 (DAC1 VOUT1 if DAC is enabled)	AIN7	AC Input (DAC1 Voltage Output if DAC is enabled)
0x4	V_{SCALE}		AVDD Scaler
0x5-0x7	Reserved		Reserved

Bits 10:8 – MUXNEG[2:0] Negative Input Mux Selection

These bits select which input will be connected to the negative input of comparator n. COMPCTRLn.MUXNEG can only be written while COMPCTRLn.ENABLE is zero.
These bits are not synchronized.

Table 45-3. MUXNEG for COMPCTRL0 and COMPCTRL1 registers

Value	Name (COMP0)	Name (COMP1)	Description
0x0	AIN0		AC Input
0x1	AIN1		AC Input
0x2	AIN2		AC Input
0x3	AIN3 (DAC1 VOUT1 if DAC is enabled)		AC Input (DAC1 Voltage Output if DAC is enabled)
0x4	AVSS		Ground
0x5	V_{SCALE}		AVDD Scaler
0x6	BANDGAP		Internal Bandgap Voltage Reference
0x7	DAC0 Output	OPAMP2 Output	DAC0 Voltage Output / OPAMP2 Output

Table 45-4. MUXNEG for COMPCTRL2 and COMPCTRL3 registers

Value	Name (COMP2)	Name (COMP3)	Description
0x0	AIN4		AC Input
0x1	AIN5		AC Input
0x2	AIN6		AC Input
0x3	AIN7		AC Input
0x4	AVSS		Ground
0x5	V_{SCALE}		AVDD Scaler
0x6	BANDGAP		Internal Bandgap Voltage Reference
0x7	DAC0 Output	OPAMP2 Output	DAC0 Voltage Output / OPAMP2 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 zero.
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 zero.
These bits are not synchronized.

Value	Description
0	Comparator n operates in continuous measurement mode.

Value	Description
1	Comparator n operates in single-shot mode.

Bit 1 – ENABLE Enable

Writing a zero to this bit disables comparator n.
Writing a one to this bit enables comparator n.
Due to synchronization, there is delay from updating the register until the comparator is enabled/disabled. The value written to COMPCTRLn.ENABLE will read back immediately after being written. SYNCBUSY.COMPCTRLn is set. SYNCBUSY.COMPCTRLn is cleared when the peripheral is enabled/disabled.
Writing a one to COMPCTRLn.ENABLE will prevent further changes to the other bits in COMPCTRLn. These bits remain protected until COMPCTRLn.ENABLE is written to zero and the write is synchronized.

45.7.13 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
		COMPCTRL3	COMPCTRL2	COMPCTRL1	COMPCTRL0	WINCTRL	ENABLE	SWRST
Access		R	R	R	R	R	R	R
Reset		0	0	0	0	0	0	0

Bits 3, 4, 5, 6 – COMPCTRL COMPCTRLn Synchronization Busy
This bit is cleared when the synchronization of the COMPCTRLn register between the clock domains is complete.
This bit is set when the synchronization of the COMPCTRLn 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.

46. Digital-to-Analog Converter (DAC)

46.1 Overview

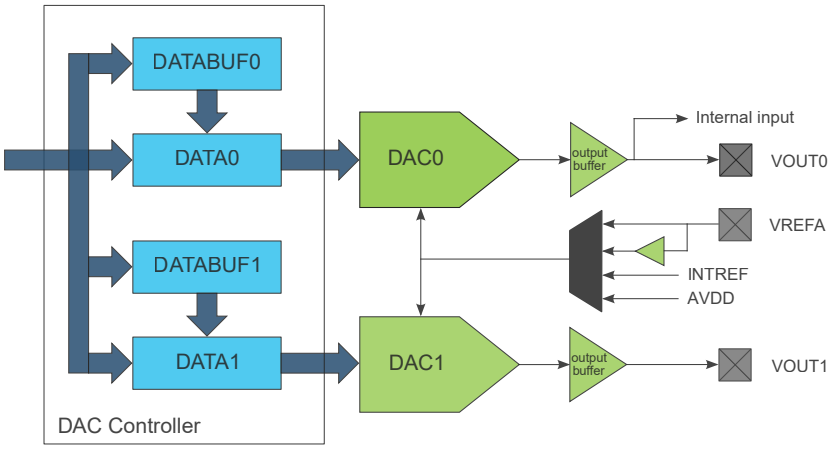
The Digital-to-Analog Converter (DAC) converts a digital value to a voltage. The DAC Controller controls two DACs, which can operate either as two independent DACs or as a single DAC in differential mode. Each DAC is 12-bit resolution and it is capable of converting up to 1,000,000 samples per second (MSPS).

46.2 Features

- Two independent DACs or single DAC in differential mode
- DAC with 12-bit resolution
- Up to 1MSPS conversion rate
- Hardware support for 16-bit using dithering
- Multiple trigger sources
- High-drive capabilities
- DAC0 used as internal input
- DMA support

46.3 Block Diagram

Figure 46-1. DAC Controller Block Diagram.



46.4 Signal Description

Signal	Description	Type
VOUT0	DAC0 output	Analog output
VOUT1	DAC1 output	Analog output
VREFA	External reference	Analog input

One signal can be mapped on several pins.

⚠ CAUTION

- Only one analog output function can be used on the same pin
- When DAC0 or DAC1 channel is enabled (DACCTRLx.ENABLE = 1), respective VOUT0 (PA02) and VOUT1 (PA07) pins cannot be used for other purposes (excluding analog inputs)

46.5 Peripheral Dependencies

Table 46-1. DAC Configuration Summary

Peripheral name	Base address	NVIC IRQ Index	MCLK AHB/APB Clocks	GCLK Peripheral Channel Index (GCLK.PCHCTRL)	PAC Peripheral Identifier Index (PAC.WRCTRL)	DMA Trigger Source Index (DMAC.CHCTRLB)	Events (EVSYS)		Power Domain (PM.STDBYCFG)
							Users (EVSYS.USER)	Generators (EVSYS.CHANNEL)	
DAC	0x42003C00	65: UNDERRUN0-1 66: EMPTY0-1	CLK_DAC_APB	30: GCLK_DAC	79	42-43: EMPTY0-1	44-45: START0-1	82-83: EMPTY0-1	PDSW

The DAC has up to two analog output pins (VOUT0, VOUT1) and one analog input pin (VREFA) that must be configured first.

When an internal input is used, it must be enabled before DAC Controller is enabled.

The analog signals of AC, ADC, DAC and OPAMP can be interconnected.

See Analog Connections of Peripherals for details.

46.6 Functional Description

46.6.1 Principle of Operation

Each DAC converts the digital value located in the Data register (DATA0 or DATA1) into an analog voltage on the DAC output (VOUT0 or VOUT1, respectively).

A conversion is started when new data is loaded to the Data register. The resulting voltage is available on the DAC output after the conversion time. A conversion can also be started by input events from Event System.

46.6.2 Basic Operation

46.6.2.1 Initialization

The following registers are enable-protected, meaning they can only be written when the DAC Controller is disabled (CTRLA.ENABLE=0):

- Control B register (CTRLB)
- Event Control register (EVCTRL)
- DAC0 Control (DACCTRL0)
- DAC1 Control (DACCTRL1)

Enable-protection is denoted by the Enable-Protected property in the register description.

46.6.2.2 Enabling, Disabling and Resetting

The DAC Controller is enabled by writing a '1' to the Enable bit in the Control A register (CTRLA.ENABLE). The DAC Controller is disabled by writing a '0' to CTRLA.ENABLE.

The DAC Controller is reset by writing '1' to the Software Reset bit in the Control A register (CTRLA.SWRST). All registers in the DAC will be reset to their initial state, and the DAC Controller will be disabled. Refer to 46.7.1. CTRLA for details.

46.6.2.3 DAC Configuration

Each individual DAC is configured by its respective DAC Control register (DACCTRLx)). These settings are applied when DAC Controller is enabled and can be changed only when DAC Controller is disabled.

- Enable the selected DAC by writing a '1' to DACCTRLx.ENABLE.
- Select the data alignment with DACCTRLx.LEFTADJ. Writing a '1' will left-align the data (DATABUFx/DATAx[31:20]). Writing a '0' to LEFTADJ will right-align the data (DATABUFx/DATAx[11:0]).
- If operation in standby mode is desired for DACx, write a '1' to the Run in Standby bit in the DAC Control register (DACCTRLx.RUNSTDBY). If RUNSTDBY=1, DACx continues normal operation when system is in standby mode. If RUNSTDBY=0, DACx is halted in standby mode.
- Select dithering mode with DACCTRLx.DITHER. Writing '1' to DITHER will enable dithering mode, writing a '0' will disable it. Refer to 46.6.3.5. Dithering Mode for details.
- Select the refresh period with the Refresh Period bit field in DACCTRLx.REFRESH[3:0]. Writing any value greater than '1' to the REFRESH bit field will enable and select the refresh mode. Refer to 46.6.3.3. Conversion Refresh for details.
- Select the output buffer current according to data rate (for low power application) with the Current Control bit field DACCTRLx.CCTRL[1:0]. Refer to 46.6.3.2. Output Buffer Current Control for details.

Once DAC Controller is enabled, DACx requires a startup time before the first conversion can start. The DACx Startup Ready bit in the Status register (STATUS.READYx) indicates that DACx is ready to convert a data when STATUS.READYx=1.

Conversions started while STATUS.READYx=0 are ignored.

VOUTx is at tri-state level if DACx is not enabled.

46.6.2.4 Digital to Analog Conversion

Each DAC converts a digital value (stored in DATAx register) into an analog voltage. The conversion range is between AVSS and the selected DAC voltage reference VREF. The default source for VREF is the internal reference voltage INTREF. Other voltage reference options are the analog supply voltage (AVDD) and the external voltage reference (VREFA). The voltage reference is selected by writing to the Reference Selection bits in the Control B register (CTRLB.REFSEL).

The output voltage from the DAC can be calculated using the following formula:

$$V_{\mathrm{OUTx}} = \frac{\mathrm{DATAx}}{4095} \times \mathrm{VREF}$$

A new conversion starts as soon as a new value is loaded into DATAx. DATAx can either be loaded via the APB bus during a CPU write operation, using DMA, or from the DATABUFx register when a STARTx event occurs.

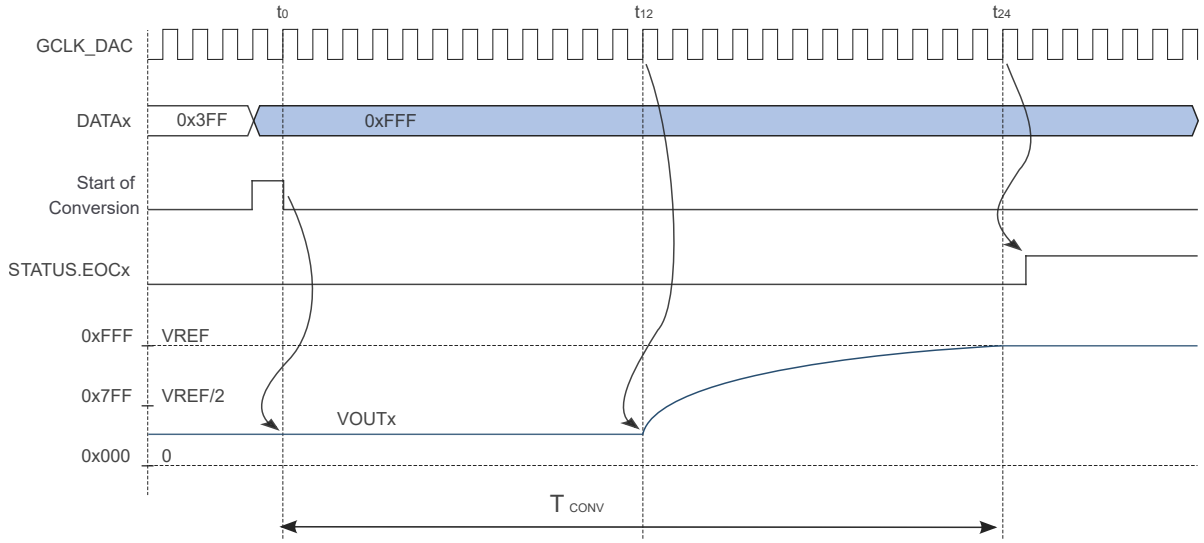
Refer to Events for details. Even if both DAC use the same GCLK, each data conversion can be started independently.

The conversion time is given by the period T_{GCLK} of the generic clock GCLK_DAC and the number of bits:

$$T_{\mathrm{CONV}} = 12 \times 2 \times T_{\mathrm{GCLK}}$$

The End Of Conversion bit in the Status register indicates that a conversion is completed (STATUS.EOCx=1). This means that VOUTx is stable.

Figure 46-2. Single DAC Conversion

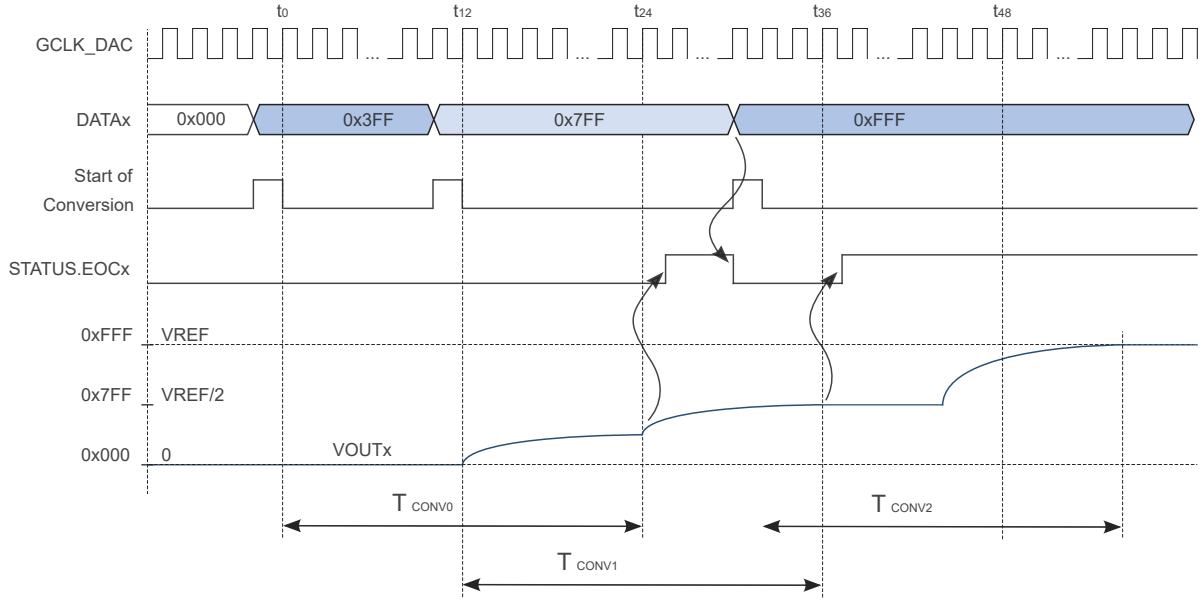


Since the DAC conversion is implemented as pipelined procedure, a new conversion can be started after only 12 GCLK_DAC periods. Therefore if DATAx is written while a conversion is ongoing, start of conversion is postponed until DACx is ready to start next conversion.

The maximum conversion rate (samples per second) is therefore:

$$\mathrm{CR}_{\max} = \frac{2}{T_{\mathrm{conv}}}$$

Figure 46-3. Multiple DAC Conversions



46.6.3 Additional Features

46.6.3.1 DAC0 as Internal Input

The analog output of DAC0, VOUT0, is internally available as input signal for other peripherals (AC, ADC, and OPAMP) when DAC0 is enabled.

Note: The pin VOUT0 will be dedicated as internal input and cannot be configured as alternate function.

46.6.3.2 Output Buffer Current Control

Power consumption can be reduced by controlling the output buffer current (DACCTRLx.CCTRL), according to conversion rate. Writing to the Current Control bits in DAC Control x register (DACCTRLx.[1:0]) will select an output buffer current.

46.6.3.3 Conversion Refresh

The DAC can only maintain its output within one LSB of the desired value for approximately 100µs. When a DAC is used to generate a static voltage or at a rate less than 20kSPS, the conversion must be refreshed periodically. The OSCULP32K clock can start new conversions automatically after a specified period. Write a value to the Refresh bit field in the DAC Control x register (DACCTRLx.REFRESH[3:0]) to select the refresh period according to the formula:

$$T_{\mathrm{REFRESH}} = \mathrm{REFRESH} \times T_{\mathrm{OSCULP32K}}$$

The actual period will depend on the tolerance of the OSCULP32K.

If DACCTRLx.REFRESH=0, there is no conversion refresh. DACCTRLx.REFRESH=1 is Reserved.

If no new conversion is started before the refresh period is completed, DACx will convert the DATAx value again.

In standby sleep mode, the refresh mode remains enabled if DACCTRLx.RUNSTDBY=1.

If DATAx is written while a refresh conversion is ongoing, the conversion of the new content of DATAx is postponed until DACx is ready to start the next conversion.

46.6.3.4 Differential Mode

DAC0 and DAC1 can be configured to operate in differential mode, i.e. the combined output is a voltage balanced around VREF/2, see also the figure below.

In differential mode, DAC0 and DAC1 are converting synchronously the DATA0 value. DATA0 must therefore be a signed value, represented in two's complement format with DATA0[11] as the signed bit. DATA0 has therefore the range [-2047:2047].

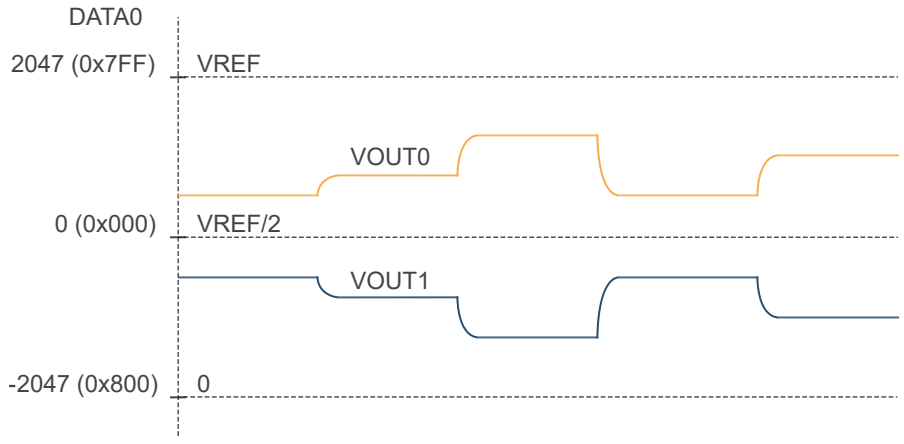
VOUT0 is the positive output and VOUT1 the negative output. The differential output voltage is therefore:

$$V_{\mathrm{OUT}} = \frac{\mathrm{DATA0}}{2047} \times \mathrm{VREF} = (V_{\mathrm{OUT0}} - V_{\mathrm{OUT1}})$$

DACCTRL0 serves as the configuration register for both DAC0 and DAC1. Therefore DACCTRL1 does not need to be written.

The differential mode is enabled by writing a '1' to the Differential bit in the Control B register (CTRLB.DIFF).

Figure 46-4. DAC Conversions in Differential Mode



46.6.3.5 Dithering Mode

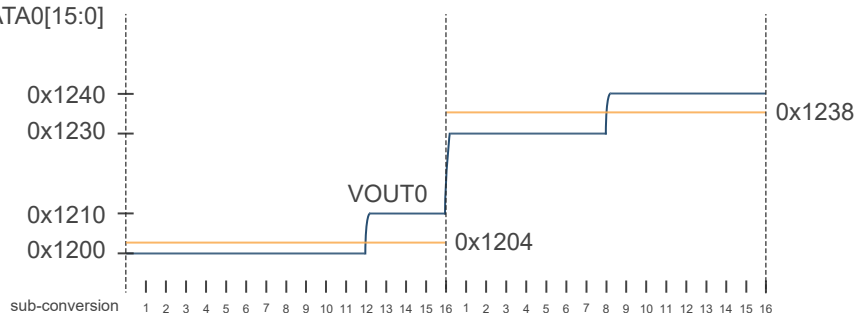
In dithering mode, DATAx is a 16-bit signed value where DATAx[15:4] is the 12-bit data converted by DAC and DATAx[3:0] represent the dither bits, used to minimize the quantization error.

The principle is to make 16 sub-conversions of the DATAx[15:4] value or the (DATAx[15:4] + 1) value, so that by averaging those two values, the conversion result of the 16-bit value (DATAx[15:0]) s accurate.

To operate, the STARTx event must be configured to generate 16 events for each DATAx[15:0] conversion, and DATABUFx must be loaded every 16 DAC conversions. EMPTYx event and DMA request are therefore generated every 16 DATABUFx to DATAx transfer. STATUS.EOCx still reports end of each sub-conversions.

Following timing diagram shows examples with DATA0[15:0] = 0x1204 followed by DATA0[15:0] = 0x1238.

Figure 46-5. DAC Conversions in Dithering Mode



46.6.4 Operating Conditions

- The DAC voltage reference must be below AVDD.
- The maximum conversion rate of 1MSPS can be achieved only if AVDD is above 2.4V.
- The frequency of GCLK_DAC must be equal or lower than 12MHz (corresponding to 1MSPS).

46.6.5 DMA Operation

In single mode (CTRLB.DIFF=0), DAC Controller generates the following DMA requests:

- Data Buffer 0 Empty (EMPTY0): The request is set when data is transferred from DATABUF0 or DATA0 to the internal data buffer of DAC0. The request is cleared when either DATA0 register or DATABUF0 register is written, or by writing a '1' to the EMPTY0 bit in the Interrupt Flag register (INTFLAG.EMPTY0).
- Data Buffer 1 Empty (EMPTY1): The request is set when data is transferred from DATABUF1 or DATA1 to the internal data buffer of DAC1. The request is cleared when either DATA0 register or DATABUF1 register is written, or by writing a one to the EMPTY1 bit in the Interrupt Flag register (INTFLAG.EMPTY1).

In differential mode (CTRLB.DIFF=1), DAC Controller generates the following DMA request:

- Data Buffer 0 Empty (EMPTY0): The request is set when data is transferred from DATABUF0 or DATA0 to the internal data buffer of DAC1. The request is cleared when either DATA0 register or DATABUF0 register is written, or by writing a one to the EMPTY0 bit in the Interrupt Flag register (INTFLAG.EMPTY0).

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, if enabled.

46.6.6 Interrupts

The DAC Controller has the following interrupt sources:

- DAC0 Data Buffer Empty (EMPTY0): Indicates that the internal data buffer of DAC0 is empty.
- DAC1 Data Buffer Empty (EMPTY1): Indicates that the internal data buffer of DAC1 is empty.
- DAC0 Underrun (UNDERRUN0): Indicates that the internal data buffer of DAC0 is empty and a DAC0 start of conversion event occurred. Refer to Events for details.
- DAC1 Underrun (UNDERRUN1): Indicates that the internal data buffer of DAC1 is empty and a DAC1 start of conversion event occurred. Refer to Events for details.

These interrupts are asynchronous wake-up sources. See Sleep Mode Controller for details.

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 DAC Controller is reset. See 46.7.6. INTFLAG 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. Refer to Nested Vector Interrupt Controller for details. 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. Refer to Nested Vector Interrupt Controller for details.

46.6.7 Events

The DAC Controller can generate the following output events:

- Data Buffer 0 Empty (EMPTY0): Generated when the internal data buffer of DAC0 is empty. Refer to 46.6.5. DMA Operation for details.
- Data Buffer 1 Empty (EMPTY1): Generated when the internal data buffer of DAC1 is empty. Refer to 46.6.5. DMA Operation for details.

Writing a '1' to an Event Output bit in the Event Control Register (EVCTRL.EMPTYEOx) enables the corresponding output event. Writing a '0' to this bit disables the corresponding output event. Refer to the Event System chapter for details on configuring the event system.

The DAC Controller can take the following actions on an input event:

- DAC0 Start Conversion (START0): DATABUF0 value is transferred into DATA0 as soon as DAC0 is ready for the next conversion, and then conversion is started. START0 is considered as asynchronous to GCLK_DAC, thus it is resynchronized in the DAC Controller. Refer to 46.6.2.4. Digital to Analog Conversion for details.
- DAC1 Start Conversion (START1): DATABUF1 value is transferred into DATA1 as soon as DAC1 is ready for the next conversion, and then conversion is started. START1 is considered as asynchronous to GCLK_DAC, thus it is resynchronized in the DAC Controller. Refer to 46.6.2.4. Digital to Analog Conversion for details.

Writing a '1' to an Event Input bit in the Event Control register (EVCTRL.STARTEIx) enables the corresponding action on input event. Writing a '0' to this bit will disable the corresponding action on input event.
Note: When a DACx Start Conversion event is enabled, only DATABUFx must be written (not DATAx).

Note: When several events are connected to the DAC Controller, the enabled action will be taken on any of the incoming events. Refer to EVSYS – Event System for details on configuring the event system.

By default, DAC Controller detects rising edge events. Falling edge detection can be enabled by writing '1' to EVCTRL.INVEIx.

46.6.8 Sleep Mode Operation

If the Run In Standby bit in the DAC Control x register DACCTRLx.RUNSTDBY=1, the DACx will continue the conversions in standby sleep mode.

If DACCTRLx.RUNSTDBY=0, the DACx will stop conversions in standby sleep mode.

If DACx conversion is stopped in standby sleep mode, DACx is also disabled to reduce power consumption. When exiting standby sleep mode, DACx is enabled again, therefore a certain startup time is required before starting a new conversion.

DAC Controller is compatible with SleepWalking: if RUNSTDBY=1, when an input event (STARTx) is detected in sleep mode, the DAC Controller will request GCLK_DAC in order to complete the conversion.

46.6.9 Debug Operation

When the CPU is halted in debug mode the DAC will halt normal operation. Any on-going conversions will be completed. The DAC can be forced to continue normal operation during debugging. If the DAC 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.

For further information, refer to the DBGCTRL Register.

46.6.10 Synchronization

Some registers (or bit fields within a register) require synchronization when read and/or written.

Synchronization is denoted by the "Read-Synchronized" (or "Read-Synchronized Bits”) and/or "Write-Synchronized" (or "Write-Synchronized Bits”) property in each individual register description.

For more details, refer to Register Synchronization.

46.7 Register Summary

Refer to the Registers Description section for more details on register properties and access permissions.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRLA	7:0							ENABLE	SWRST
0x01	CTRLB	7:0						REFSEL[1:0]		DIFF
0x02	EVCTRL	7:0			INVEI1	INVEI0	EMPTYEO1	EMPTYEO0	STARTEI1	STARTEI0
0x03	Reserved									
0x04	INTENCLR	7:0					EMPTY1	EMPTY0	UNDERRUN1	UNDERRUN0
0x05	INTENSET	7:0					EMPTY1	EMPTY0	UNDERRUN1	UNDERRUN0
0x06	INTFLAG	7:0					EMPTY1	EMPTY0	UNDERRUN1	UNDERRUN0
0x07	STATUS	7:0					EOC1	EOC0	READY1	READY0
0x08	SYNCBUSY	31:24								
		23:16								
		15:8								
		7:0			DATABUF1	DATABUF0	DATA1	DATA0	ENABLE	SWRST
0x0C	DACCTRL0	15:8					REFRESH[3:0]			
		7:0	DITHER	RUNSTDBY			CCTRL[1:0]		ENABLE	LEFTADJ
0x0E	DACCTRL1	15:8					REFRESH[3:0]			
		7:0	DITHER	RUNSTDBY			CCTRL[1:0]		ENABLE	LEFTADJ
0x10	DATA0	15:8	DATA[15:8]							
		7:0	DATA[7:0]							
0x12	DATA1	15:8	DATA[15:8]							
		7:0	DATA[7:0]							
0x14	DATABUF0	15:8	DATABUF[15:8]							
		7:0	DATABUF[7:0]							
0x16	DATABUF1	15:8	DATABUF[15:8]							
		7:0	DATABUF[7:0]							
0x18	DBGCTRL	7:0								DBGRUN

46.7.1 Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized Bits

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

Bit 1 – ENABLE Enable DAC Controller

Note: This bit is write-synchronized: SYNCBUSY.ENABLE must be checked to ensure the CTRLA.ENABLE synchronization is complete.

Value	Description
0	The peripheral is 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 DAC to their initial state, and the DAC will be disabled.
Writing a '1' to CTRLA.SWRST will always take precedence, meaning that all other writes in the same write-operation will be discarded.

Notes:

1. When the CTRLA.SWRST is written, the user must poll SYNCBUSY.SWRST bit to know when the reset operation is complete.
2. 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.

46.7.2 Control B

Name: CTRLB
Offset: 0x01
Reset: 0x02
Property: PAC Write-Protection, Enable-Protected

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

Bits 2:1 – REFSEL[1:0] Reference Selection
This bit field selects the Reference Voltage for both DACs.

Value	Name	Description
0x0	VREFAU	Unbuffered external voltage reference (not buffered in DAC, direct connection)
0x1	AVDD	Voltage supply
0x2	VREFAB	Buffered external voltage reference (buffered in DAC)
0x3	INTREF	Internal voltage reference

Bit 0 – DIFF Differential Mode Enable
This bit defines the conversion mode for both DACs.

Value	Description
0	Single mode
1	Differential mode

46.7.3 Event Control

Name: EVCTRL
Offset: 0x02
Reset: 0x00
Property: PAC Write-Protection, Enable-Protected

Bit	7	6	5	4	3	2	1	0
			INVEI1	INVEI0	EMPTYEO1	EMPTYEO0	STARTEI1	STARTEI0
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit 5 – INVEI1 Enable Inversion of DAC1 input event
This bit defines the edge detection of the input event for DAC1 (START1).

Value	Description
0	Rising edge.
1	Falling edge.

Bit 4 – INVEI0 Enable Inversion Data Buffer Empty Event Output DAC0
This bit defines the edge detection of the input event for DAC0 (START0).

Value	Description
0	Rising edge.
1	Falling edge.

Bit 3 – EMPTYEO1 Data Buffer Empty Event Output DAC1
This bit indicates if the Data Buffer Empty Event output for DAC1 is enabled.

Value	Description
0	Data Buffer Empty event is disabled.
1	Data Buffer Empty event is enabled.

Bit 2 – EMPTYEO0 Data Buffer Empty Event Output DAC0
This bit indicates if the Data Buffer Empty Event output for DAC0 is enabled.

Value	Description
0	Data Buffer Empty event is disabled.
1	Data Buffer Empty event is enabled.

Bit 1 – STARTEI1 Start Conversion Event Input DAC1
This bit indicates if the Start input event for DAC1 is enabled.

Value	Description
0	A new conversion will not be triggered on any incoming event. Only DATA1 must be written (not DATABUF1).
1	A new conversion will be triggered on any incoming event. Only DATABUF1 must be written (not DATA1).

Bit 0 – STARTEI0 Start Conversion Event Input DAC0
This bit indicates if the Start input event for DAC0 is enabled.

Value	Description
0	A new conversion will not be triggered on any incoming event. Only DATA0 must be written (not DATABUF0).
1	A new conversion will be triggered on any incoming event. Only DATABUF0 must be written (not DATA0).

46.7.4 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
					EMPTY1	EMPTY0	UNDERRUN1	UNDERRUN0
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit 3 – EMPTY1 Data Buffer 1 Empty Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Data Buffer 1 Empty Interrupt Enable bit, which disables the Data Buffer 1 Empty interrupt.

Value	Description
0	The Data Buffer 1 Empty interrupt is disabled.
1	The Data Buffer 1 Empty interrupt is enabled.

Bit 2 – EMPTY0 Data Buffer 0 Empty Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Data Buffer 0 Empty Interrupt Enable bit, which disables the Data Buffer 0 Empty interrupt.

Value	Description
0	The Data Buffer 0 Empty interrupt is disabled.
1	The Data Buffer 0 Empty interrupt is enabled.

Bit 1 – UNDERRUN1 Underrun Interrupt Enable for DAC1
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Data Buffer 1 Underrun Interrupt Enable bit, which disables the Data Buffer 1 Underrun interrupt.

Value	Description
0	The Data Buffer 1 Underrun interrupt is disabled.
1	The Data Buffer 1 Underrun interrupt is enabled.

Bit 0 – UNDERRUN0 Underrun Interrupt Enable for DAC0
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Data Buffer 0 Underrun Interrupt Enable bit, which disables the Data Buffer 0 Underrun interrupt.

Value	Description
0	The Data Buffer 0 Underrun interrupt is disabled.
1	The Data Buffer 0 Underrun interrupt is enabled.

46.7.5 Interrupt Enable Set

Name: INTENSET
Offset: 0x05
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 Clear register (INTENCLR).

Bit	7	6	5	4	3	2	1	0
					EMPTY1	EMPTY0	UNDERRUN1	UNDERRUN0
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit 3 – EMPTY1 Data Buffer 1 Empty Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Data Buffer 1 Empty Interrupt Enable bit, which enables the Data Buffer 1 Empty interrupt.

Value	Description
0	The Data Buffer 1 Empty interrupt is disabled.
1	The Data Buffer 1 Empty interrupt is enabled.

Bit 2 – EMPTY0 Data Buffer 0 Empty Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Data Buffer 0 Empty Interrupt Enable bit, which enables the Data Buffer 0 Empty interrupt.

Value	Description
0	The Data Buffer 0 Empty interrupt is disabled.
1	The Data Buffer 0 Empty interrupt is enabled.

Bit 1 – UNDERRUN1 Underrun Interrupt Enable for DAC1
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Data Buffer 1 Underrun Interrupt Enable bit, which enables the Data Buffer 1 Underrun interrupt.

Value	Description
0	The Data Buffer 1 Underrun interrupt is disabled.
1	The Data Buffer 1 Underrun interrupt is enabled.

Bit 0 – UNDERRUN0 Underrun Interrupt Enable for DAC0
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Data Buffer 0 Underrun Interrupt Enable bit, which enables the Data Buffer 0 Underrun interrupt.

Value	Description
0	The Data Buffer 0 Underrun interrupt is disabled.
1	The Data Buffer 0 Underrun interrupt is enabled.

46.7.6 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x06
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
					EMPTY1	EMPTY0	UNDERRUN1	UNDERRUN0
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit 3 – EMPTY1 Data Buffer 1 Empty
This flag is cleared by writing a '1' to it or by writing new data to DATA1 or DATABUF1.
This flag is set when the data buffer for DAC1 is empty and will generate an interrupt request if INTENSET.EMPTY1=1.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Data Buffer 1 Empty interrupt flag.

Bit 2 – EMPTY0 Data Buffer 0 Empty
This flag is cleared by writing a '1' to it or by writing new data to DATA0 or DATABUF0.
This flag is set when the data buffer for DAC0 is empty and will generate an interrupt request if INTENSET.EMPTY0=1.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Data Buffer 0 Empty interrupt flag.

Bit 1 – UNDERRUN1 DAC1 Underrun
This flag is cleared by writing a '1' to it.
This flag is set when a start conversion event (START1) occurred before new data is copied/written to the DAC1 data buffer and will generate an interrupt request if INTENSET.UNDERRUN1=1.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the DAC1 Underrun interrupt flag.

Bit 0 – UNDERRUN0 DAC0 Underrun
This flag is cleared by writing a '1' to it.
This flag is set when a start conversion event (START0) occurred before new data is copied/written to the DAC) data buffer and will generate an interrupt request if INTENSET.UNDERRUN0=1.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the DAC0 Underrun interrupt flag.

46.7.7 Status

Name: STATUS
Offset: 0x07
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
					EOC1	EOC0	READY1	READY0
Access					R	R	R	R
Reset					0	0	0	0

Bit 3 – EOC1 DAC1 End of Conversion
This bit is cleared when DATA1 register is written.

Value	Description
0	No conversion completed since last load of DATA1.
1	DAC1 conversion is complete, VOUT1 is stable.

Bit 2 – EOC0 DAC0 End of Conversion
This bit is cleared when DATA0 register is written.

Value	Description
0	No conversion completed since last load of DATA0.
1	DAC0 conversion is complete, VOUT0 is stable.

Bit 1 – READY1 DAC1 Startup Ready

Value	Description
0	DAC1 is not ready for conversion.
1	Startup time has elapsed, DAC1 is ready for conversion.

Bit 0 – READY0 DAC0 Startup Ready

Value	Description
0	DAC0 is not ready for conversion.
1	Startup time has elapsed, DAC0 is ready for conversion.

46.7.8 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
			DATABUF1	DATABUF0	DATA1	DATA0	ENABLE	SWRST
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bit 5 – DATABUF1 Data Buffer DAC1
This bit is set when DATABUF1 register is written.
This bit is cleared when DATABUF1 synchronization is completed.

Value	Description
0	No ongoing synchronized access.
1	Synchronized access is ongoing.

Bit 4 – DATABUF0 Data Buffer DAC0
This bit is set when DATABUF0 register is written.
This bit is cleared when DATABUF0 synchronization is completed.

Value	Description
0	No ongoing synchronized access.
1	Synchronized access is ongoing.

Bit 3 – DATA1 Data DAC1
This bit is set when DATA1 register is written.
This bit is cleared when DATA1 synchronization is completed.

Value	Description
0	No ongoing synchronized access.
1	Synchronized access is ongoing.

Bit 2 – DATA0 Data DAC0
This bit is set when DATA0 register is written.
This bit is cleared when DATA0 synchronization is completed.

Value	Description
0	No ongoing synchronized access.
1	Synchronized access is ongoing.

Bit 1 – ENABLE DAC Enable Status

This bit is set when CTRLA.ENABLE bit is written.

This bit is cleared when CTRLA.ENABLE synchronization is completed.

Value	Description
0	No ongoing synchronization.
1	Synchronization is ongoing.

Bit 0 – SWRST Software Reset

This bit is set when CTRLA.SWRST bit is written.

This bit is cleared when CTRLA.SWRST synchronization is completed.

Value	Description
0	No ongoing synchronization.
1	Synchronization is ongoing.

46.7.9 DAC0 Control

Name: DACCTRL0
Offset: 0x0C
Reset: 0x0000
Property: PAC Write-Protection, Enabled-Protected

Bit	15	14	13	12	11	10	9	8
					REFRESH[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
	DITHER	RUNSTDBY			CCTRL[1:0]		ENABLE	LEFTADJ
Access	R/W	R/W			R/W	R/W	R/W	R/W
Reset	0	0			0	0	0	0

Bits 11:8 – REFRESH[3:0] Refresh period
This field defines the refresh period.

Value	Description
0x0	Refresh is disabled.
0x1	Reserved
0x2 to 0xF	$T_{REFRESH} = REFRESH \times 30.52\mu s$

Bit 7 – DITHER Dithering Mode

Value	Description
0	Dithering mode is disabled.
1	Dithering mode is enabled.

Bit 6 – RUNSTDBY Run in Standby
This bit controls the behavior of DAC0 during standby sleep mode.

Value	Description
0	DAC0 is disabled during standby sleep mode.
1	DAC0 continues to operate during standby sleep mode.

Bits 3:2 – CCTRL[1:0] Current Control
This field defines the current in output buffer according to conversion rate.

Value	Name	Description
0x0	CC100K	GCLK_DAC ≤ 1.2MHz (100kSPS)
0x1	CC1M	1.2MHz < GCLK_DAC ≤ 6MHz (500kSPS)
0x2	CC12M	6MHz < GCLK_DAC ≤ 12MHz (1MSPS)
0x3	-	Reserved

Bit 1 – ENABLE Enable DAC0
This bit enables DAC0 when DAC Controller is enabled (CTRLA.ENABLE).

⚠ CAUTION When DAC0 channel is enabled (DACCTRL0.ENABLE = 1), VOUT0 (PA02) pin cannot be used for other purposes (excluding analog inputs).

Value	Description
0	DAC0 is disabled.
1	DAC0 is enabled.

Bit 0 – LEFTADJ Left Adjusted Data

This bit controls how the 12-bit conversion data is adjusted in the Data and Data Buffer registers.

Value	Description
0	DATA0 and DATABUF0 registers are right-adjusted.
1	DATA0 and DATABUF0 registers are left-adjusted.

46.7.10 DAC1 Control

Name: DACCTRL1
Offset: 0x0E
Reset: 0x0000
Property: PAC Write-Protection, Enabled-Protected

Bit	15	14	13	12	11	10	9	8
					REFRESH[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
	DITHER	RUNSTDBY			CCTRL[1:0]		ENABLE	LEFTADJ
Access	R/W	R/W			R/W	R/W	R/W	R/W
Reset	0	0			0	0	0	0

Bits 11:8 – REFRESH[3:0] Refresh period
This field defines the refresh period.

Value	Description
0x0	Refresh is disabled.
0x1	Reserved
0x2 to 0xF	$T_{REFRESH} = REFRESH \times 30.52\mu s$

Bit 7 – DITHER Dithering Mode

Value	Description
0	Dithering mode is disabled.
1	Dithering mode is enabled.

Bit 6 – RUNSTDBY Run in Standby
This bit controls the behavior of DAC1 during standby sleep mode.

Value	Description
0	DAC1 is disabled during standby sleep mode.
1	DAC1 continues to operate during standby sleep mode.

Bits 3:2 – CCTRL[1:0] Current Control
This field defines the current in output buffer.

Figure 46-6. Current Control

Value	Name	Description
0x0	CC100K	GCLK_DAC ≤ 1.2MHz (100kSPS)
0x1	CC1M	1.2MHz < GCLK_DAC ≤ 6MHz (500kSPS)
0x2	CC12M	6MHz < GCLK_DAC ≤ 12MHz (1MSPS)
0x3	-	Reserved

Bit 1 – ENABLE Enable DAC1
This bit enables DAC1 when DAC Controller is enabled (CTRLA.ENABLE).

⚠ CAUTION When DAC1 channel is enabled (DACCTRL1.ENABLE = 1), VOUT1 (PA07) pin cannot be used for other purposes (excluding analog inputs).

Value	Description
0	DAC1 is disabled.
1	DAC1 is enabled.

Bit 0 – LEFTADJ Left Adjusted Data

This bit controls how the 12-bit conversion data is adjusted in the Data and Data Buffer registers.

Value	Description
0	DATA1 and DATABUF1 registers are right-adjusted.
1	DATA1 and DATABUF1 registers are left-adjusted.

46.7.11 Data DAC0

Name: DATA0
Offset: 0x10
Reset: 0x0000
Property: PAC Write-Protection, Write-Synchronized

Note: This register is write-synchronized: SYNCBUSY.DATA0 must be checked to ensure the DATA0 register synchronization is complete.

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] DAC0 Data

DATA0 register contains the 12-bit value that is converted to a voltage by the DAC0. The adjustment of these 12 bits within the 16-bit register is controlled by DACCTRL0.LEFTADJ:
- DATA[11:0] when DACCTRL0.LEFTADJ=0.
- DATA[15:4] when DACCTRL0.LEFTADJ=1.

In dithering mode (whatever DACCTRL0.LEFTADJ value):
- DATA[15:4] are the 12-bit converted by DAC0.
- DATA[3:0] are the dither bits.

46.7.12 Data DAC1

Name: DATA1
Offset: 0x12
Reset: 0x0000
Property: PAC Write-Protection, Write-Synchronized

Note: This register is write-synchronized: SYNCBUSY.DATA1 must be checked to ensure the DATA1 register synchronization is complete.

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] DAC1 Data

DATA1 register contains the 12-bit value that is converted to a voltage by the DAC1. The adjustment of these 12 bits within the 16-bit register is controlled by DACCTRL1.LEFTADJ:
- DATA[11:0] when DACCTRL1.LEFTADJ=0.
- DATA[15:4] when DACCTRL1.LEFTADJ=1.

In dithering mode (whatever DACCTRL1.LEFTADJ value):
- DATA[15:4] are the 12-bit converted by DAC1.
- DATA[3:0] are the dither bits.

46.7.13 Data Buffer DAC0

Name: DATABUF0
Offset: 0x14
Reset: 0x0000
Property: Write-Synchronized

Note: This register is write-synchronized: SYNCBUSY.DATABUF0 must be checked to ensure the DATABUF0 register synchronization is complete.

Bit	15	14	13	12	11	10	9	8
	DATABUF[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
	DATABUF[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – DATABUF[15:0] DAC0 Data Buffer
DATABUF0 contains the value to be transferred into DATA0 when a START0 event occurs.

46.7.14 Data Buffer DAC1

Name: DATABUF1
Offset: 0x16
Reset: 0x0000
Property: Write-Synchronized

Note: This register is write-synchronized: SYNCBUSY.DATABUF1 must be checked to ensure the DATABUF1 register synchronization is complete.

Bit	15	14	13	12	11	10	9	8
	DATABUF[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
	DATABUF[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – DATABUF[15:0] DAC1 Data Buffer
DATABUF1 contains the value to be transferred into DATA1 when a START1 event occurs.

46.7.15 Debug Control

Name: DBGCTRL
Offset: 0x18
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
								DBGRUN
Access								
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 DAC is halted when the CPU is halted by an external debugger. Any ongoing conversion will complete.
1	The DAC continues normal operation when the CPU is halted by an external debugger.

47. Operational Amplifier Controller (OPAMP)

47.1 Overview

The Operational Amplifier (OPAMP) Controller configures and controls three low power, general purpose operational amplifiers offering a high degree of flexibility and rail-to-rail inputs.

Most common inverting or non-inverting programmable gain and hysteresis configurations can be selected by software, no external components are required for these configurations.

The OPAMPs can be cascaded for both standalone mode and built-in configurations.

Each OPAMP can be used as a standalone amplifier. External pins are available for filter configurations or other applications. A reference can be generated from the DAC to be used as selectable reference for inverting PGA (programmable gain amplifier) or instrumentation amplifier. Each OPAMP can be used as buffer or PGA for the ADC or an AC. The OPAMP offset voltage can be compensated when it is used in combination with the ADC.

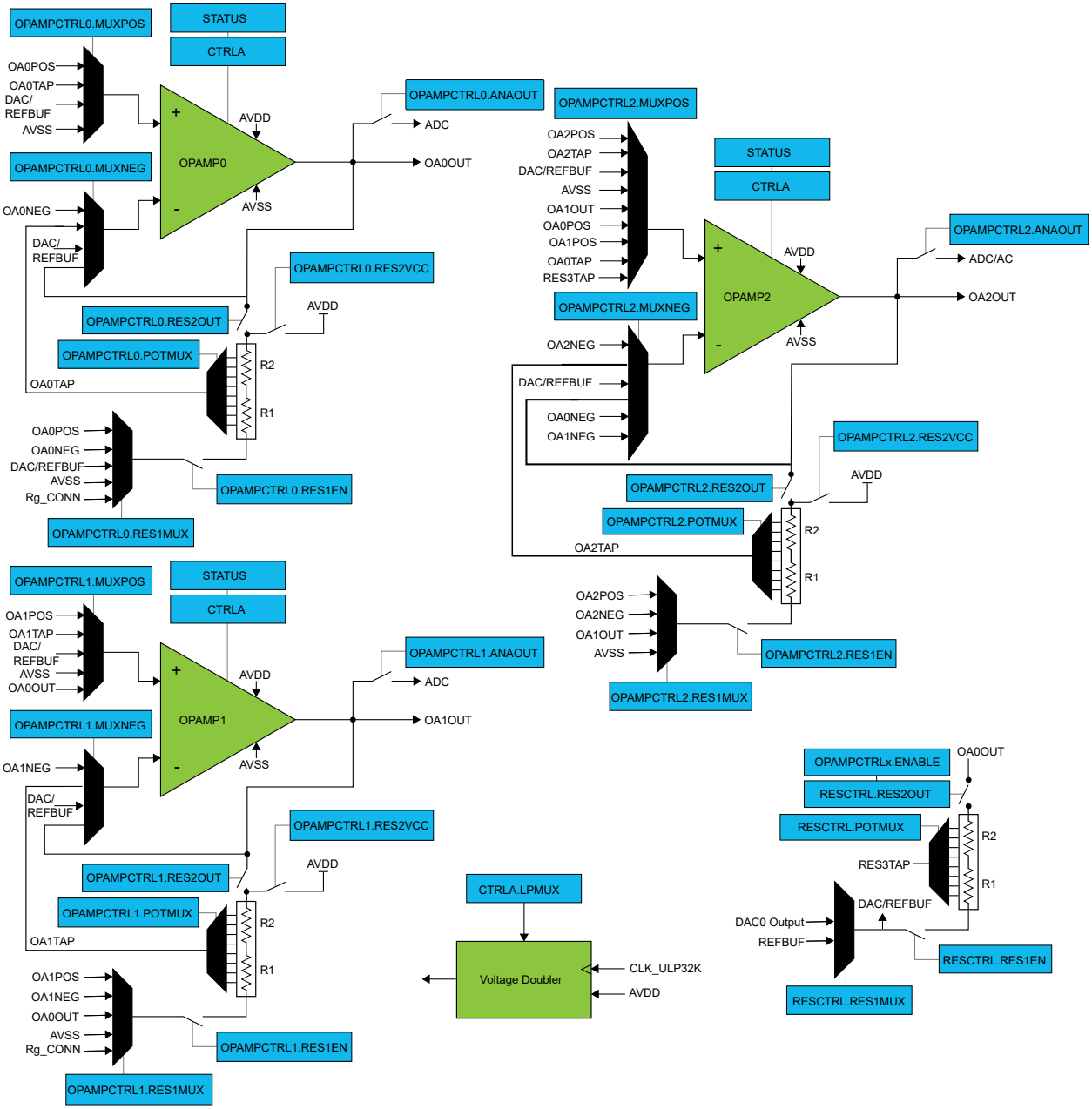
Four modes are available to select the trade-off between speed and power consumption to best fit the application requirements and optimize the power consumption.

47.2 Features

- Three individually configurable low power OPAMPs
- Rail-to-rail inputs
- Configurable resistor ladders for internal feedback
- Selectable configurations
  - Standalone OPAMP with flexible inputs
  - Unity gain amplifier
  - Non-inverting / inverting Programmable Gain Amplifier (PGA)
  - Cascaded PGAs
  - Instrumentation amplifier
  - Comparator with programmable hysteresis
- OPAMP output:
  - On I/O pins
  - As input for AC or ADC
- Flexible input selection:
  - I/O pins
  - DAC
  - Ground
- Low-power options:
  - Selectable voltage doubler and propagation delay versus current consumption
  - On demand start-up for ADC and AC operations
- Offset/Gain measurement for calibration when used with the ADC

47.3 Block Diagram

Figure 47-1. OPAMP Block Diagram



47.4 Signal Description

Signal	Description	Type
OA0POS	OPAMP0 positive input	Analog input
OA0NEG	OPAMP0 negative input	Analog input
OA1POS	OPAMP1 positive input	Analog input

.....continued		
Signal	**Description**	**Type**
OA1NEG	OPAMP1 negative input	Analog input
OA2POS	OPAMP2 positive input	Analog input
OA2NEG	OPAMP2 negative input	Analog input
OA0OUT	OPAMP0 output	Analog output
OA1OUT	OPAMP1 output	Analog output
OA2OUT	OPAMP2 output	Analog output

One signal can be mapped on several pins.

⚠ CAUTION Only one analog output function can be used on the same pin.

47.5 Peripheral Dependencies

Table 47-1. OPAMP Configuration Summary

Peripheral name	Base address	MCLK AHB/APB Clocks	PAC Peripheral Identifier Index (PAC.WRCTRL)	Power Domain (PM.STDBYCFG)
OPAMP	0x42005000	CLK_OPAMP_APB	84	PDSW

Each OPAMP has two I/O pins that can be used as analog inputs. These pins must be configured for analog operation before using them as OPAMP inputs.

If the DAC is to be used as OPAMP input, the DAC must be configured and enabled first.

Each OPAMP has one I/O pin that can be used as analog output. This pin must be configured for analog operation before using it as OPAMP output.

The analog signals of AC, ADC, DAC and OPAMP can be interconnected. The AC and ADC peripheral can request the OPAMP using an analog ONDEMAND functionality

See Analog Connections of Peripherals for details.

47.5.1 Clocks

The OPAMP bus clock (CLK_OPAMP_APB) can be enabled and disabled in the Power Manager, and the default state of CLK_OPAMP_APB can be found in the Peripheral Clock Masking.

A clock (CLK_ULP32K) is required by the voltage doubler for low voltage operation (VCC < 2.5V). The CLK_ULP32K is a 32.768 kHz clock which is provided by the OSCULP32K oscillator in the OSC32KCTRL module.

47.6 Functional Description

47.6.1 Principle of Operation

Each OPAMP has one positive and one negative input. Each input may be chosen from either a selection of analog input pins, or internal inputs such as the DAC, the resistor ladder, and the ground and output of another OPAMP.

Each OPAMP can be configured with built-in feedback to support various functions with programmable or unity gain.

I/O pins are externally accessible so that the operational amplifier can be configured with external feedback.

All OPAMPs can be cascaded to support circuits such as differential amplifiers.

47.6.2 Basic Operation

Each operational amplifier can be configured in different modes, selected by the OPAMP Control x register (OPAMPCTRLx):

- Standalone operational amplifier
- Operational amplifier with built-in feedback

After being enabled, a start-up delay is added before the output of the operational amplifier is available. This start-up time is measured internally to account for environmental changes such as temperature or voltage supply level.

When the OPAMP is ready, the respective Ready x bit in the Status register is set (STATUS.READYx=1).

If the supply voltage is below 2.5V, the start-up time is also dependent on the voltage doubler. If the supply voltage is always above 2.5V, the voltage doubler can be disabled by setting the Low-Power Mux bit in the Control A Register (CTRLA.LPMUX).

47.6.2.1 Initialization

The OPAMP must be configured with the desired properties and inputs before it is enabled.

A clock (CLK_ULP32K) is required by the voltage doubler for low voltage operation (VCC < 2.5V). The CLK_ULP32K is a 32.768 kHz clock which is provided by the OSCULP32K oscillator in the OSC32KCTRL module. See OSC32KCTRL – 32KHz Oscillators Controller for further details.

47.6.2.2 Enabling, Disabling, and Resetting

The OPAMP is enabled by writing a '1' to the Enable bit in the Control A register (CTRLA.ENABLE). The OPAMP is disabled by writing a '0' to CTRLA.ENABLE.

Each OPAMP sub-module is enabled by writing a '1' to the Enable bit in the OPAMP Control x register (OPAMPCTRLx.ENABLE). Each OPAMP sub-module is disabled by writing a '0' to OPAMPCTRLx.ENABLE.

The OPAMP module is reset by writing a '1' to the Software Reset bit in the Control A register (CTRLA.SWRST). All registers in the OPAMP will be reset to their initial state, and the OPAMP will be disabled. Refer to 47.7.1. CTRLA for details.

47.6.3 Sleep Mode Operation

The OPAMPs can also be used during sleep modes. The 32.768 kHz clock source used by the voltage doubler must remain active. See Voltage Doubler for more details.

Each OPAMP x can be configured to behave differently in different sleep modes. The behavior is determined by the individual Run in Standby and On Demand bits in the OPAMP Control x registers (OPAMPCTRLx.RUNSTDBY, and OPAMPCTRLx.ONDEMAND), as well as the common Enable bit in the Control A register (CTRLA.ENABLE).

Table 47-2. Individual OPAMP Sleep Mode Operation

OPAMPCTRLx.RUNSTDBY	OPAMPCTRLx.ONDEMAND	CTRLA.ENABLE	Sleep Behavior
-	-	0	Disabled
0	0	1	Always run in all sleep modes except STANDBY sleep mode
0	1	1	Only run in all sleep modes except STANDBY sleep mode if requested by a peripheral.
1	0	1	Always run in all sleep mode
1	1	1	Only run in all sleep modes if requested by a peripheral.

Note:
When OPAMPCTRLx.ONDEMAND=1, the analog block is powered off for the lowest power consumption if it is not requested.

When requested, a start-up time delay is necessary when the system returns from sleep. The start-up time is depending on the Bias Selection bits in the OPAMP Control x register (OPAMPCTRLx.BIAS) and the corresponding speed/current consumption requirements.

47.6.4 Debug Operation

When the CPU is halted in debug mode the OPAMP continues normal operation. If the OPAMP 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.

47.6.5 Configuring the Operational Amplifiers

Each individual operational amplifier is configured by its respective Operational Amplifier Control x register (OPAMPCTRLx). These settings must be configured before the amplifier is started.

- Select the positive input in OPAMPCTRLx.MUXPOS
- Select the negative input in OPAMPCTRLx.MUXNEG
- Select RES1EN if resistor ladder is used
- Select the input for the resistor ladder in OPAMPCTRLx.RES1MUX
- Select the potentiometer selection of the resistor ladder in OPAMPCTRLx.POTMUX
- Select the VCC input for the resistor ladder in OPAMPCTRLx.RES2VCC
- Connect the operational amplifier output to the resistor ladder using OPAMPCTRLx.RES2OUT
- Select the trade-off between speed and energy consumption in OPAMPCTRLx.BIAS

47.6.6 Standalone Mode

Each operational amplifier can be used as standalone amplifier. In this mode, positive input, negative input and the output are routed from/to external I/Os, requiring external feedback. OPAMPs can also be cascaded to support multiple OPAMP configurations. Refer to Operational Amplifier Control x register (OPAMPCTRLx) for further details on how to configure OPAMP I/Os.

47.6.7 Built-in Modes

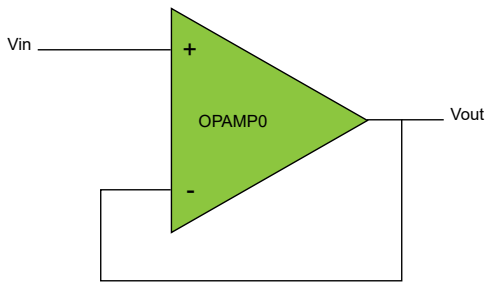
47.6.7.1 Voltage Follower

In this mode the unity gain path is selected for the negative input. The OPAMPCTRLx register can be configured as follows:

Table 47-3. Configuration - Three Independent Unitary Gain Followers

	MUXPOS	MUXNEG	RES1MUX	POTMUX	RES2VCC	RES2OUT	RES1EN	ANAOUT
OPAMP0	0000	011	-	-	0	0	0	0
OPAMP1	0000	011	-	-	0	0	0	0
OPAMP2	0000	011	-	-	0	0	0	0

Figure 47-2. Voltage follower



47.6.7.2 Inverting PGA

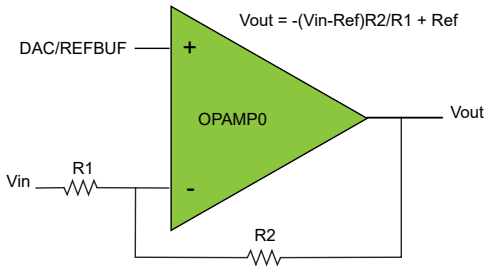
For inverting programmable gain amplifier operation, the OPAMPCTRLx registers can be configured as follows:

Table 47-4. Configuration - Three Independent Inverting PGAs

	MUXPOS	MUXNEG	RES1MUX	POTMUX	RES2VCC	RES2OUT	RES1EN	ANAOUT
OPAMP0	0010	001	001	100	0	1	1	0
OPAMP1	0010	001	001	100	0	1	1	0
OPAMP2	0010	001	001	100	0	1	1	0

Inverting PGA (Example: Vout=-3.Vin, R1=4R, R2=12R)

Figure 47-3. Inverting PGA



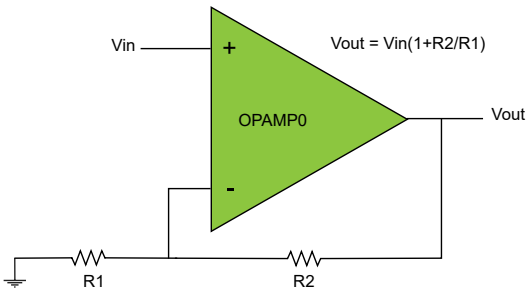
47.6.7.3 Non-Inverting PGA

For non-inverting programmable gain amplifier operation, the OPAMPCTRLx registers can be configured as follows:

Table 47-5. Configuration - Three Independent Non-Inverting PGAs (Example: Vout=4.Vin, R1=4R, R2=12R)

	MUXPOS	MUXNEG	RES1MUX	POTMUX	RES2VCC	RES2OUT	RES1EN	ANAOUT
OPAMP0	000	001	11	100	0	1	1	0
OPAMP1	000	001	11	100	0	1	1	0
OPAMP2	000	001	11	100	0	1	1	0

Figure 47-4. Non-Inverting PGA



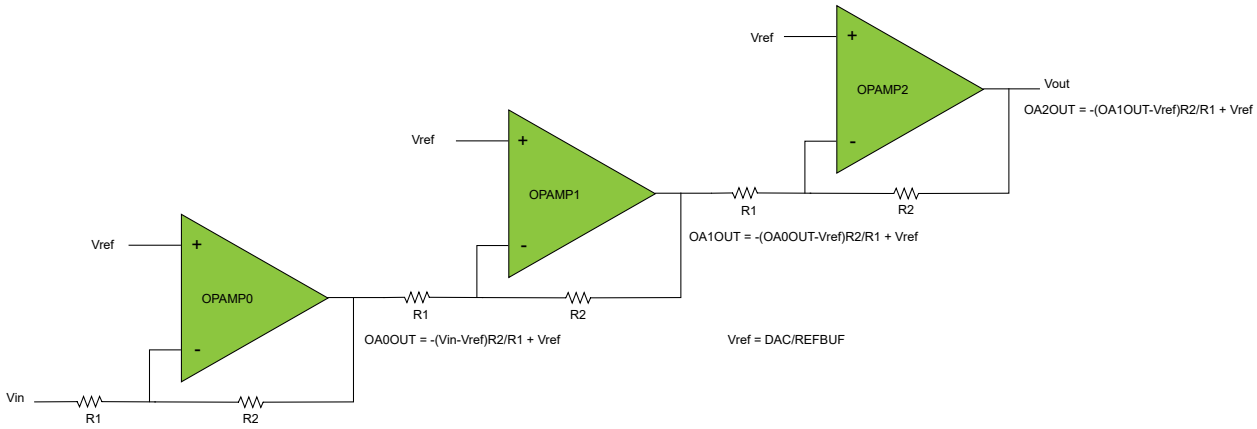
47.6.7.4 Cascaded Inverting PGA

The OPAMPs can be configured as three cascaded, inverting PGAs using these settings in OPAMPCTRLx:

Table 47-6. Cascade of three inverting PGAs (Example: R1=4R, R2=12R)

	MUXPOS	MUXNEG	RES1MUX	POTMUX	RES2VCC	RES2OUT	RES1EN	ANAOUT
OPAMP0	0010	001	001	100	0	1	1	0
OPAMP1	0010	001	010	100	0	1	1	0
OPAMP2	0010	001	010	100	0	1	1	0

Figure 47-5. Cascaded Inverting PGA



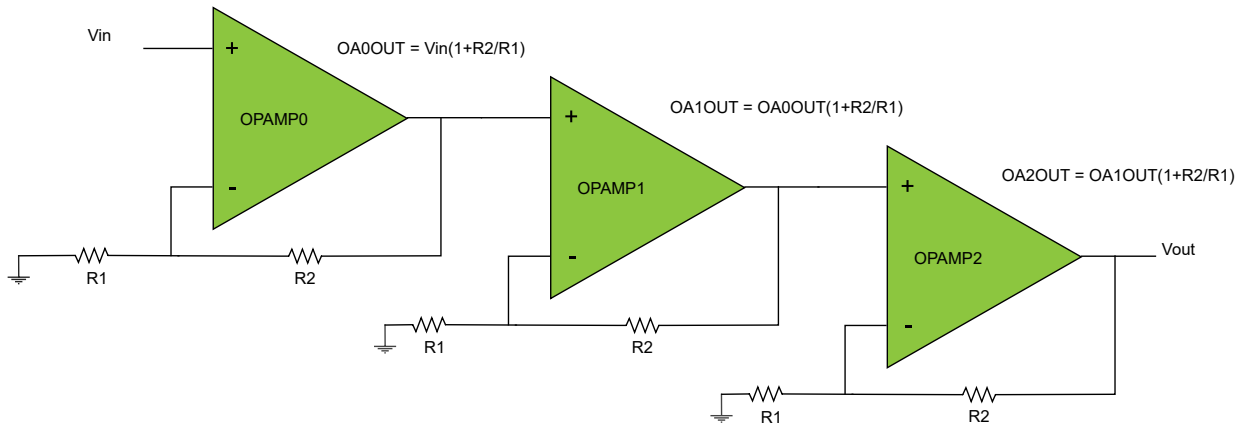
47.6.7.5 Cascaded Non-Inverting PGA

The OPAMPs can be configured as three cascaded, non-inverting PGAs using these settings in OPAMPCTRLx:

Table 47-7. Cascaded Non-Inverting PGA (Exemple: R1=4R, R2=12R)

	MUXPOS	MUXNEG	RES1MUX	POTMUX	RES2VCC	RES2OUT	RES1EN	ANAOUT
OPAMP0	0000	001	011	100	0	1	1	0
OPAMP1	0100	001	011	100	0	1	1	0
OPAMP2	0100	001	011	100	0	1	1	0

Figure 47-6. Cascaded Non-Inverting PGA



47.6.7.6 Two OPAMPs Differential Amplifier

In this mode, OPAMP0 can be coupled with OPAMP1 or OPAMP1 with OPAMP2 in order to amplify a differential signal.

To configure OPAMP0 and OPAMP1 as differential amplifier, the OPAMPCTRLx register can be configured as follows:

Table 47-8. OPAMP0 OPAMP1 Differential Amplifier (Example: R1=4R, R2=12R)

	MUXPOS	MUXNEG	RES1MUX	POTMUX	RES2VCC	RES2OUT	RES1EN	ANAOUT
OPAMP0	0000	011	011	000	0	0	0	0

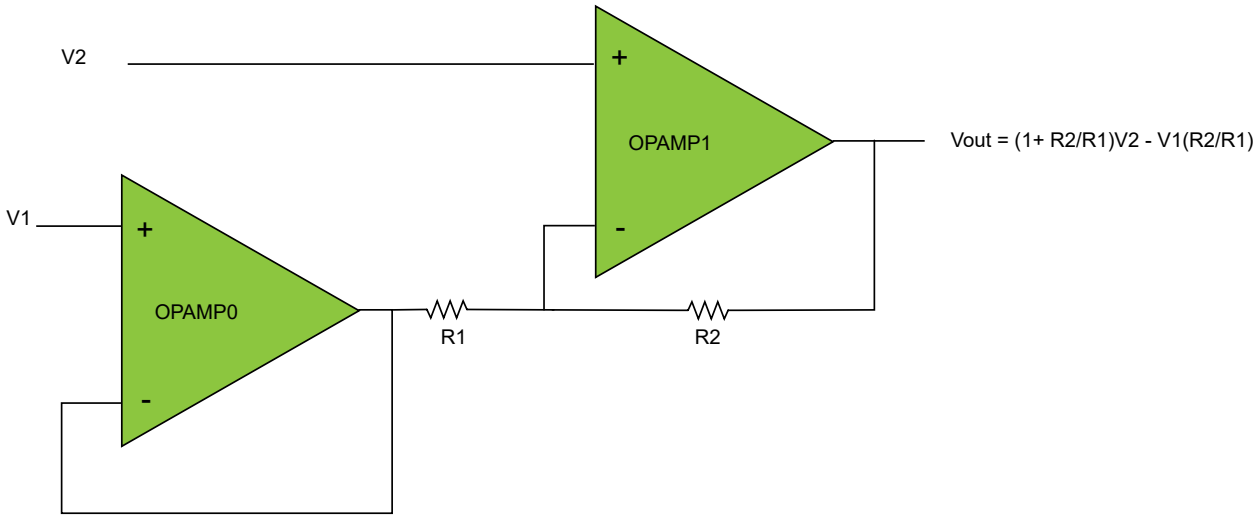
.....continued								
	MUXPOS	MUXNEG	RES1MUX	POTMUX	RES2VCC	RES2OUT	RES1EN	ANAOUT
OPAMP1	0000	001	010	100	0	1	1	0
OPAMP2	0000	000	000	000	0	0	0	0

To configure OPAMP1 and OPAMP2 as differential amplifier, the OPAMPCTRLx register can be configured as follows:

Table 47-9. OPAMP1 OPAMP2 Differential Amplifier (Example: R1=4R, R2=12R)

	MUXPOS	MUXNEG	RES1MUX	POTMUX	RES2VCC	RES2OUT	RES1EN	ANAOUT
OPAMP0	0000	000	000	000	0	0	0	0
OPAMP1	0000	011	011	000	0	0	0	0
OPAMP2	0000	001	010	100	0	1	1	0

Figure 47-7. OPAMP0 OPAMP1 Differential Amplifier



47.6.7.7 Instrumentation Amplifier

In this mode, OPAMP0 and OPAMP1 are configured as voltage followers. The OPAMPCTRLx register can be configured as follows:

Table 47-10. Instrumentation Amplifier Configuration

	MUXPOS	MUXNEG	RES1MUX	POTMUX	RES2VCC	RES2OUT	RES1EN	ANAOUT
OPAMP0	0000	011	010	010	0	1	1	0
OPAMP1	0000	011	011	000	0	0	0	0
OPAMP2	0111	001	010	010	0	1	1	0

The resistor ladders associated with OPAMP0 and OPAMP2 must be configured as follows in order to select the appropriate gain:

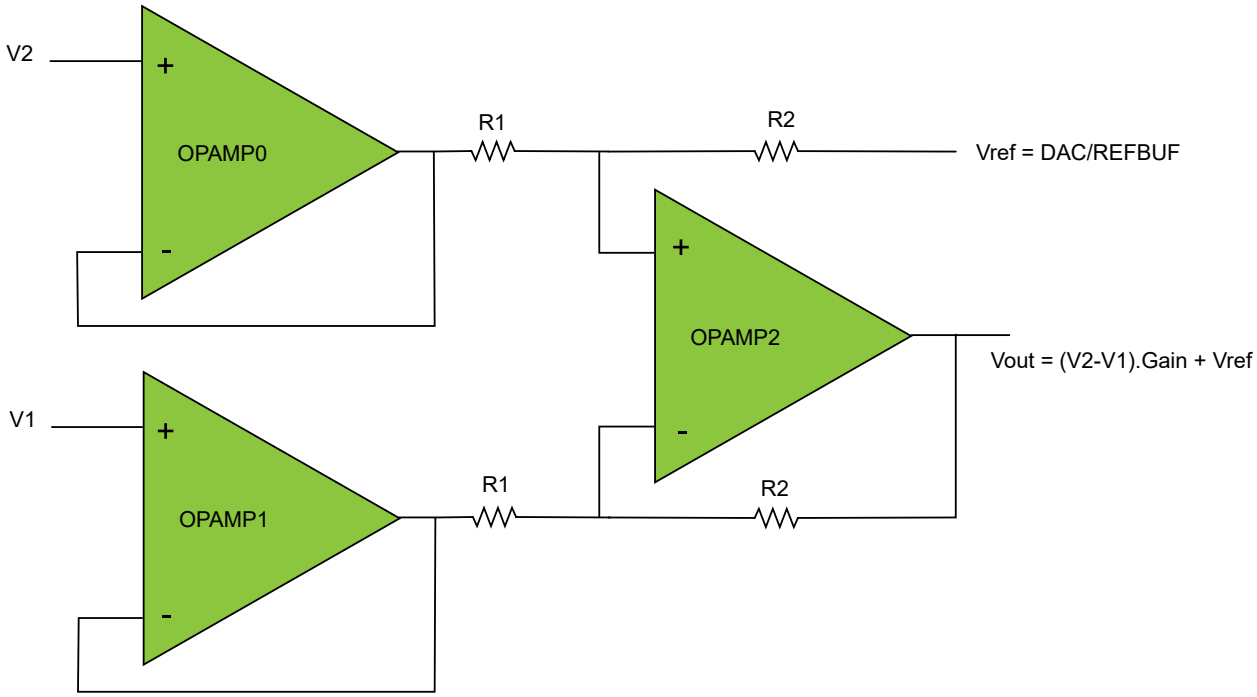
Table 47-11. Instrumentation Amplifier Gain Selection

OPAMPCTRL0.POTMUX	OPAMPCTRL2.POTMUX	GAIN
0x7	Reserved	Reserved
0x6	0x0	1/7

.....continued		
OPAMPCTRL0.POTMUX	OPAMPCTRL2.POTMUX	GAIN
0x5	Reserved	Reserved
0x4	0x1	1/3
0x3	Reserved	Reserved
0x2	0x2	1
0x1	0x4	3
0x0	0x6	7

Note: Either the DAC or AVSS must be the reference, selected by the OPAMPCTRL0.RES1MUX bits. Refer to the OPAMP Control 'x' register (47.7.3. OPAMPCTRL) for details.

Figure 47-8. Instrumentation amplifier



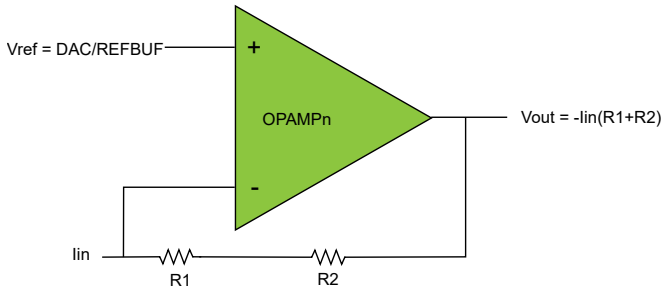
47.6.7.8 Transimpedance amplifier

Each OPAMP can be configured as a transimpedance amplifier (current to voltage converter). In this mode the positive input is connected to ground. The negative input is connected to the output through the resistor ladder. The OPAMPCTRLx register can be configured as follows:

Table 47-12. Transimpedance Amplifier

	MUXPOS	MUXNEG	RES1MUX	POTMUX	RES2VCC	RES2OUT	RES1EN	ANAOUT
OPAMP0	0010	000	-	-	0	0	0	0
OPAMP1	0010	000	-	-	0	0	0	0
OPAMP2	0010	000	-	-	0	0	0	0

Figure 47-9. Transimpedance Amplifier



Note: R1 and R2 are external user supplied resistors.

47.6.7.9 Programmable Hysteresis

OPAMP can be configured as an inverting or non-inverting comparator with programmable hysteresis. Applying hysteresis will prevent constant toggling of the output, caused by noise when the input signals are close to each other.

In both inverting and non-inverting comparator configurations the positive input is connected to the resistor ladder. When OPAMP is configured as an inverting comparator with programmable hysteresis, the input voltage must be applied to the negative input and RES1MUX must be connected to the ground. When an OPAMP is configured as a non-inverting comparator with programmable hysteresis, the input voltage must be applied to RES1MUX and the negative input must be connected to the ground.

To configure an OPAMP as an inverting comparator with programmable hysteresis, the OPAMPCTRLx register can be configured as follows:

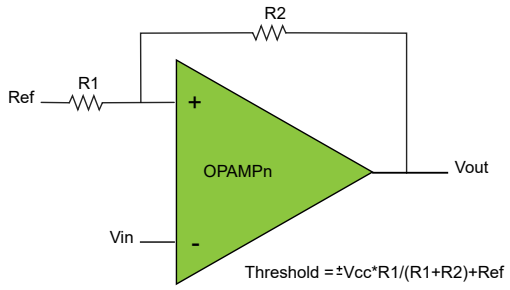
Table 47-13. Configuration of Input Multiplexes for OPAMP0 (Example: Vth = 3/4*Vcc, Ref = AVSS)

	MUXPOS	MUXNEG	RES1MUX	POTMUX	RES2VCC	RES2OUT	RES1EN	ANAOUT
OPAMP0	0001	000	010	001	0	1	1	0

Table 47-14. POTMUX [2:0]: Potentiometer Selection

Value	R1	R2	Threshold = Vcc * R1 / (R1 + R2)
0x0	14R	2R	7/8 * Vcc
0x1	12R	4R	3/4 * Vcc
0x2	8R	8R	1/2 * Vcc
0x3	6R	10R	3/8 * Vcc
0x4	4R	12R	1/4 * Vcc
0x5	3R	13R	3/16 * Vcc
0x6	2R	14R	1/8 * Vcc
0x7	R	15R	1/16 * Vcc

Figure 47-10. Inverting comparator with programmable hysteresis



To configure an OPAMP as a non-inverting comparator with programmable hysteresis, the OPAMPCTRLx register can be configured as follows:

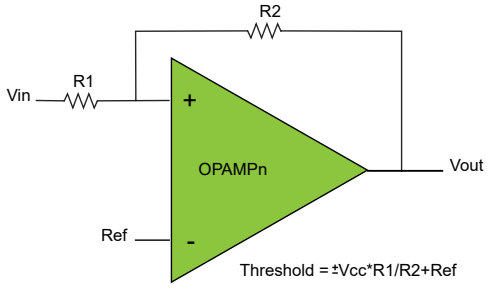
Table 47-15. Configuration of Input Multiplexes for OPAMP0 and OPAMP1 (Example: Vth = 1/3*Vcc, Ref = AVSS)

	MUXPOS	MUXNEG	RES1MUX	POTMUX	RES2VCC	RES2OUT	RES1EN	ANAOUT
OPAMP0	0001	010	000	100	0	1	1	0
OPAMP1	0001	010	000	100	0	1	1	0
OPAMP2	0001	010	000	100	0	1	1	0

Table 47-16. POTMUX [2:0]: Potentiometer Selection

Value	R1	R2	Threshold = Vcc * R1 / R2
0x0	14R	2R	Vcc * 7 (unused)
0x1	12R	4R	Vcc * 3 (unused)
0x2	8R	8R	Vcc (unused)
0x3	6R	10R	0.6* Vcc
0x4	4R	12R	1/3 * Vcc
0x5	3R	13R	3/13 *Vcc
0x6	2R	14R	1/7 * Vcc
0x7	R	15R	1/15 * Vcc

Figure 47-11. Non-Inverting comparator with programmable hysteresis



47.6.8 ADC Driver

47.6.8.1 Buffer/PGA for ADC

Each OPAMP can be configured as a buffer or a PGA for the other modules (such as ADC or AC). OPAMPs can also be cascaded to increase the programmable gain.

The output to the OPAMP must be enabled by writing a '1' to the Analog Output bit in the Operational Amplifier x Control register (OPAMPCTRLx.ANAOUT). The ADC input mux must be configured to select OPAMP as input. Refer to ADC – Analog-to-Digital Converter for details on configuring the ADC.

47.6.8.2 Offset and Gain Compensation

When the OPAMP is used in combination with the ADC, the OPAMP offset and gain errors can be compensated. To calculate offset and gain error compensation values

1. Configure OPAMP as Voltage Follower.
2. Route the OPAMP output to the ADC:
   - Write a '1' to the Analog Output bit in the Operational Amplifier x Control register (OPAMPCTRLx.ANAOUT)
   - Select the OPAMP as input for the ADC, see ADC – Analog-to-Digital Converter
3. Measure and set the Offset Correction value for the ADC OFFSETCORR register as in 47.6.8.3. Offset Compensation.
4. Measure and set the Gain Correction value for the ADC GAINCORR register as in 47.6.8.4. Gain Compensation.

The offset error compensation must be determined before gain error compensation.

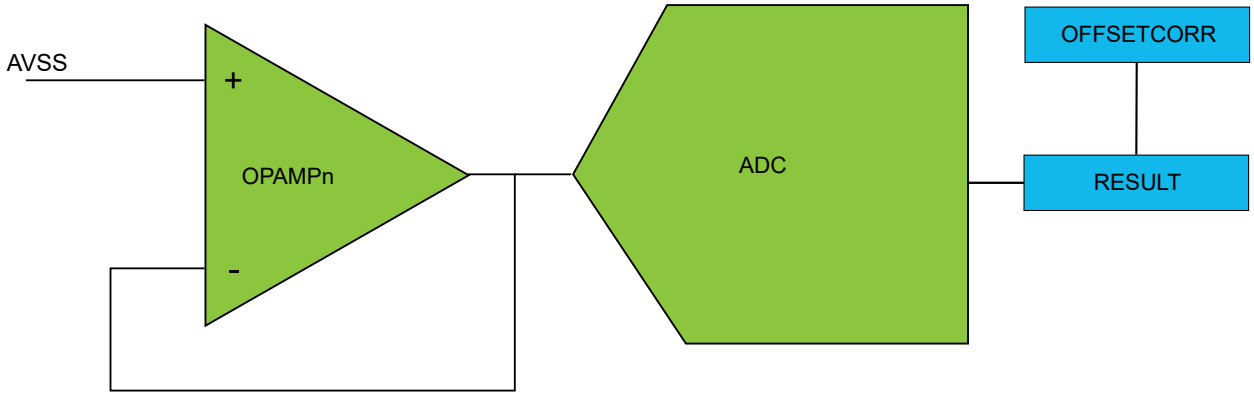
The relation for offset and gain error compensation is shown in this equation:

Result = (converted value + OFFSETCORR)*GAINCORR

47.6.8.3 Offset Compensation

To determine the offset compensation value, the positive input must be tied to ground. The result of the ADC conversion gives directly the offset compensation value that must be written in the ADC OFFSETCORR register.

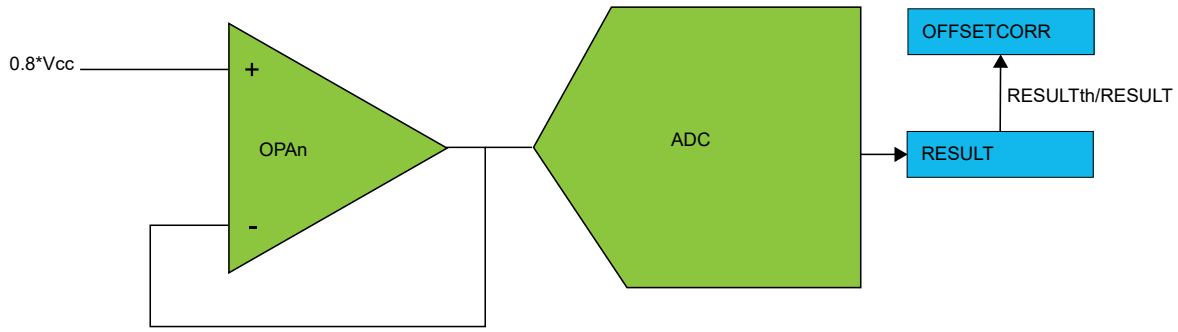
Figure 47-12. Offset Compensation



47.6.8.4 Gain Compensation

To perform gain compensation positive input must be close to AVDD, but less (e.g. 0.8*Vref for instance) to avoid ADC saturation. The value for gain error compensation is obtained by dividing the theoretical ADC conversion result by the result from measurement. The obtained value for gain error compensation must be written in the ADC GAINCORR register.

Figure 47-13. Gain Compensation



47.6.9 AC Driver

One or several OPAMPs can be configured as input for the AC. The AC input mux must be appropriately configured to select OPAMP as input.

47.6.10 Input Connection to DAC

The DAC can be used as a reference. This is configured by the corresponding OPAMPCTRLx.MUXPOS and OPAMPCTRLx.RES1MUX bits.

47.6.11 Reference Buffer (REFBUF)

The OPAMP has an internal reference voltage (REFBUF) which can be selected on each OPAMP (OPAMPCTRLx).

RESCTRL.REFBUFLEVEL bit field allows to select the Voltage Level of this internal reference.

47.6.12 Voltage Doubler

The OPAMP peripheral contains a voltage doubler for the analog multiplexer switches to ensure proper operation for a supply voltage below 2.5V. Aside from the multiplexers, no other supply voltages are affected by the voltage doubler.

The voltage doubler is normally switched on/off automatically, based on the supply level. If the supply voltage is guaranteed to be above 2.5V, the voltage doubler can be completely disabled by writing the Low-Power Mux bit in the Control Register (CTRLA.LPMUX).

Note: The voltage doubler must not be disabled when the internal REFBUF is used (CTRLA.LPMUX = 0).

When enabling OPAMPs, additional start-up time is required for the voltage doubler to settle. Disabling the voltage doubler saves power and reduces the startup time.

47.6.13 Performance vs. Power Consumption

It is possible to tradeoff speed versus power efficiency to get the shortest possible propagation delay or the lowest power consumption.

The speed setting is configured for each amplifier individually by the Bias Control field in the Operational Amplifier x Control register (OPAMPCTRLx.BIAS). The BIAS bits select the amount of bias current provided to the operational amplifiers. This will also affect the start-up time.

47.7 Register Summary

Refer to the Registers Description section for more details on register properties and access permissions.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRLA	7:0	LPMUX						ENABLE	SWRST
0x01	Reserved									
0x02	STATUS	7:0						READY2	READY1	READY0
0x03	Reserved									
0x04	OPAMPCTRL0	31:24								
		23:16	MUXNEG[3:0]				MUXPOS[3:0]			
		15:8	POTMUX[2:0]			RES1MUX[2:0]			RES1EN	RES2OUT
		7:0	ONDEMAND	RUNSTDBY	RES2VCC	BIAS[1:0]		ANAOUT	ENABLE	
0x08	OPAMPCTRL1	31:24								
		23:16	MUXNEG[3:0]				MUXPOS[3:0]			
		15:8	POTMUX[2:0]			RES1MUX[2:0]			RES1EN	RES2OUT
		7:0	ONDEMAND	RUNSTDBY	RES2VCC	BIAS[1:0]		ANAOUT	ENABLE	
0x0C	OPAMPCTRL2	31:24								
		23:16	MUXNEG[3:0]				MUXPOS[3:0]			
		15:8	POTMUX[2:0]			RES1MUX[2:0]			RES1EN	RES2OUT
		7:0	ONDEMAND	RUNSTDBY	RES2VCC	BIAS[1:0]		ANAOUT	ENABLE	
0x10	RESCTRL	7:0	REFBUFLEVEL[1:0]		POTMUX[2:0]			RES1MUX	RES1EN	RES2OUT

47.7.1 Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
	LPMUX						ENABLE	SWRST
Access	R/W						R/W	R/W
Reset	0						0	0

Bit 7 – LPMUX Low-Power Mux

Note: The voltage doubler must not be disabled when the internal REFBUF is used (CTRLA.LPMUX = 0).

Value	Description
0	The analog input muxes have low resistance, but consume more power at lower voltages (e.g., are driven by the voltage doubler).
1	The analog input muxes have high resistance, but consume less power at lower voltages (e.g., the voltage doubler is disabled).

Bit 1 – ENABLE Enable

Value	Description
0	The peripheral is disabled.
1	The peripheral is enabled. Each OPAMP must also be enabled individually by the Enable bit in the corresponding OPAMP Control register (OPAMPCTRLx.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 MODULE to their initial state, and the OPAMP will be disabled.
Writing a '1' to CTRLA.SWRST will always take precedence, meaning that all other writes in the same write-operation will be discarded.

Value	Description
0	There is no reset operation ongoing.
1	The reset operation is ongoing.

47.7.2 Status

Name: STATUS
Offset: 0x02
Reset: 0x00
Property: –

Bit	7	6	5	4	3	2	1	0
						READY2	READY1	READY0
Access						R	R	R
Reset						0	0	0

Bits 0, 1, 2 – READYx OPAMP x Ready [x=0..2]

This bit is set when the OPAMPx output is ready.
This bit is cleared when the output of OPAMPx is not ready.

47.7.3 OPAMP Control x

Name: OPAMPCTRL
Offset: 0x04 + n*0x04 [n=0..2]
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
	MUXNEG[3:0]				MUXPOS[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	15	14	13	12	11	10	9	8
	POTMUX[2:0]			RES1MUX[2:0]			RES1EN	RES2OUT
Access	R/W	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
	ONDEMAND	RUNSTDBY	RES2VCC	BIAS[1:0]		ANAOUT	ENABLE	
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	0	0	0	0	0	0	0	

Bits 23:20 – MUXNEG[3:0] Negative Input Mux Selection
Selection on negative input for operational amplifier x.

Value	OPAMPx	Name	Description
0x0	x=0,1,2	OAxNEG	OPAMPx Negative Input
0x1	x=0,1,2	OAxTAP	OPAMPx Resistor Ladder Taps
0x2	x=0,1,2	DAC/REFBUF	DAC0 VOUT0 or OPAMP Reference Output Voltage (REFBUF)
0x3	x=0,1,2	OAxOUT	-
0x4	x=0,1	Reserved	-
	x=2	OA0NEG	OPAMP0 Negative Input
0x5	x=0,1	Reserved	-
	x=2	OA1NEG	OPAMP1 Negative Input
0x6 - 0xF	-	Reserved	-

Bits 19:16 – MUXPOS[3:0] Positive Input Mux Selection
Selection on positive input for operational amplifier x.

Value	OPAMPx	Name	Description
0x0	x=0,1,2	OAxPOS	OPAMPx Positive Input
0x1	x=0,1,2	OAxTAP	OPAMPx Resistor Ladder Taps
0x2	x=0,1,2	DAC/REFBUF	DAC0 VOUT0 or OPAMP Reference Output Voltage (REFBUF)
0x3	x=0,1,2	AVSS	Ground
0x4	x=0	Reserved	-
	x=1	OA0OUT	OPAMP0 output
	x=2	OA1OUT	OPAMP1 output

.....continued			
Value	**OPAMPx**	**Name**	**Description**
0x5	x=0,1	Reserved	-
	x=2	OA0POS	OPAMP0 Positive Input
0x6	x=0,1	Reserved	-
	x=2	OA1POS	OPAMP1 Positive Input
0x7	x=0,1	Reserved	-
	x=2	OA0TAP	OPAMP0 Resistor Ladder Taps
0x8	x=0,1	Reserved	-
	x=2	RES3TAP	RES3TAP Potentiometer
0x9 - 0xF	-	Reserved	-

Bits 15:13 – POTMUX[2:0] Potentiometer selection

Resistor selection bits control a numeric potentiometer with eight fixed values.

Value	R1	R2
0x0	14R	2R
0x1	12R	4R
0x2	8R	8R
0x3	6R	10R
0x4	4R	12R
0x5	3R	13R
0x6	2R	14R
0x7	R	15R

Bits 12:10 – RES1MUX[2:0] Resistor 1 Mux

These bits select the connection of R1 resistor of the potentiometer.

Value	OPAMPx	Name	Description
0x0	x=0,1,2	OAxPOS	OPAMPx Positive Input
0x1	x=0,1,2	OAxNEG	OPAMPx Negative Input
0x2	x=0	DAC/REFBUF	DAC0 VOUT0 or OPAMP Reference Output Voltage (REFBUF)
	x=1	OA0OUT	OPAMP0 output
	x=2	OA1OUT	OPAMP1 output
0x3	x=0,1,2	AVSS	Ground
0x4	x=0,1	Rg_CONN	Rg_CONN
	x=2	Reserved	-

Bit 9 – RES1EN Resistor 1 Enable

Value	Description
0	R1 disconnected from RES1MUX.
1	R1 connected to RES1MUX.

Bit 8 – RES2OUT Resistor ladder To Output

Value	Description
0	Swith open.
1	Switch closed.

Bit 7 – ONDEMAND On Demand Control

The On Demand operation mode allows the OPAMPx to be enabled or disabled, depending on other peripheral requests.

Value	Description
0	The OPAMPx is always on, if enabled.

Value	Description
1	The OPAMPx is enabled when a peripheral is requesting the OPAMPx to be used as an input. The OPAMPx is disabled if no peripheral is requesting it as an input.

Bit 6 – RUNSTDBY Run in Standby

This bit controls how the OPAMPx behaves during standby sleep mode:

Value	Description
0	The OPAMPx is disabled in standby sleep mode.
1	The OPAMPx is not stopped in standby sleep mode. If OPAMPCTRLx.ONDEMAND=1, the OPAMPx will be running when a peripheral is requesting it as an input. If OPAMPCTRLx.ONDEMAND=0, OPAMPx will always be running in standby sleep mode.

Bit 5 – RES2VCC Resistor ladder To VCC

Value	Description
0	Swith open.
1	Switch closed.

Bits 4:3 – BIAS[1:0] Bias Selection

These bits are used to select the bias mode.

Value	Name	Description
0x0	MODE0	Minimum current consumption, but the slowest mode
0x1	MODE1	Low current consumption, slow speed
0x2	MODE2	High current consumption, fast speed
0x3	MODE3	Maximum current consumption but the fastest mode

Bit 2 – ANAOUT Analog Output

This bit controls a switch connected to the OPAMP output.

Value	Description
0	Switch open. No ADC or AC connection.
1	Switch closed. OPAMP output is connected to the ADC or AC input.

Bit 1 – ENABLE Operational Amplifier Enable

Value	Description
0	The OPAMPx is disabled
1	The OPAMPx is enabled

47.7.4 Resistor Control

Name: RESCTRL
Offset: 0x10
Reset: 0x0
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
	REFBUFLEVEL[1:0]		POTMUX[2:0]			RES1MUX	RES1EN	RES2OUT
Access	R/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 7:6 – REFBUFLEVEL[1:0] OPAMP Reference Output Voltage (REFBUF) Level Selection
REFBUFLEVEL allows to select the OPAMP Reference Output Voltage (REFBUF) Level.

Value	Reference Level
0x0	1.1V
0x1	1.25V
0x2	1.6V
0x3	Reserved

Bits 5:3 – POTMUX[2:0] Potentiometer selection
Resistor selection bits control a numeric potentiometer with eight fixed values.

Value	R1	R2
0x0	14R	2R
0x1	12R	4R
0x2	8R	8R
0x3	6R	10R
0x4	4R	12R
0x5	3R	13R
0x6	2R	14R
0x7	R	15R

Bit 2 – RES1MUX Resistor 1 Mux
These bits select the connection of R1 resistor of the potentiometer.

Value	Name	Description
0x0	DAC	DAC VOUT0
0x1	REFBUF	OPAMP Reference Output Voltage (REFBUF)

Bit 1 – RES1EN Resistor 1 Enable
RES1EN = 1 is required in order to provide DAC/REFBUF to POSMUX, NEGMUX and RES1MUX.

Value	Description
0	R1 disconnected from RES1MUX.
1	R1 connected to RES1MUX.

Bit 0 – RES2OUT Resistor ladder To Output
RES2OUT switch can only be closed if all OPAMPs are enabled.

Value	Description
0	Swith open.
1	Switch closed.

48. Peripheral Touch Controller (PTC)

48.1 Overview

The Peripheral Touch Controller (PTC) acquires signals in order to detect touch on capacitive sensors. The external capacitive touch sensor is typically formed on a PCB, and the sensor electrodes are connected to the analog front end of the PTC through the I/O pins in the device. The PTC supports both self- and mutual-capacitance sensors.

In mutual-capacitance mode, sensing is done using capacitive touch matrices in various X-Y configurations, including indium tin oxide (ITO) sensor grids. The PTC requires one pin per X-line and one pin per Y-line.

In self-capacitance mode, the PTC requires only one pin (Y-line) for each touch sensor.

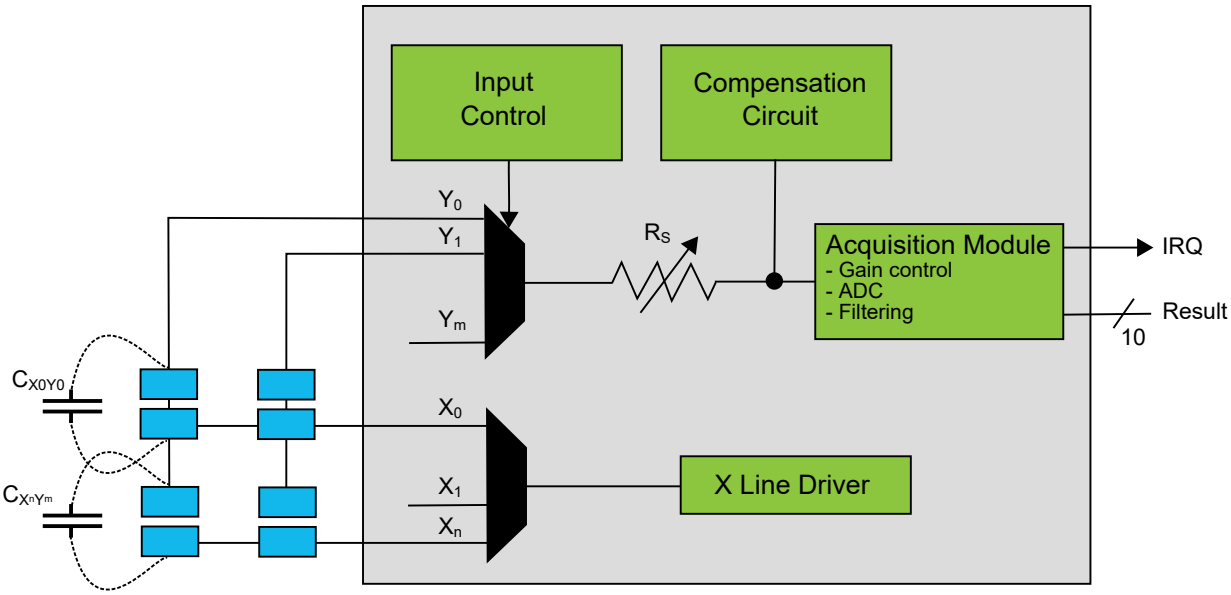
The number of available pins and the assignment of X- and Y-lines is depending on both package type and device configuration. Refer to the Configuration Summary and Pinout for details.

48.2 Features

- Low-power, high-sensitivity, environmentally robust capacitive touch buttons, sliders, and wheels
- Driven Shield Plus for better noise immunity and moisture tolerance
  - Any PTC X/Y line can be used for the driven shield
  - All enabled sensors will be driven at the same potential as the sensor scanned
- Supports wake-up on touch from standby Sleep mode
- Supports mutual capacitance and self-capacitance sensing
  - Mix-and-match mutual-and self-capacitance sensors
- One pin per electrode – no external components
- Load compensating charge sensing
  - Parasitic capacitance compensation and adjustable gain for superior sensitivity
- Zero drift over the temperature and AVDD range
  - Auto calibration and recalibration of sensors
- Single-shot and free-running charge measurement
- Hardware noise filtering and noise signal desynchronization for high conducted immunity
- Polarity control, allowing Parallel Acquisition (through the Touch Library) individually controls the polarity of each line
- Selectable channel change delay allows choosing the settling time on a new channel, as required
- Acquisition-start triggered by command or through auto-triggering feature
- Low CPU utilization through interrupt on acquisition-complete

48.3 Block Diagram

Figure 48-1. PTC Block Diagram Mutual Capacitance



Note: For PIC32CM LE00/LS00/LS60 the R_S = 100 KΩ.

A mutual-capacitance sensor is formed between two I/O lines - an X electrode for transmitting and Y electrode for sensing. The mutual capacitance between the X and Y electrode is measured by the Peripheral Touch Controller.

Figure 48-2. Mutual Capacitance Sensor Arrangement

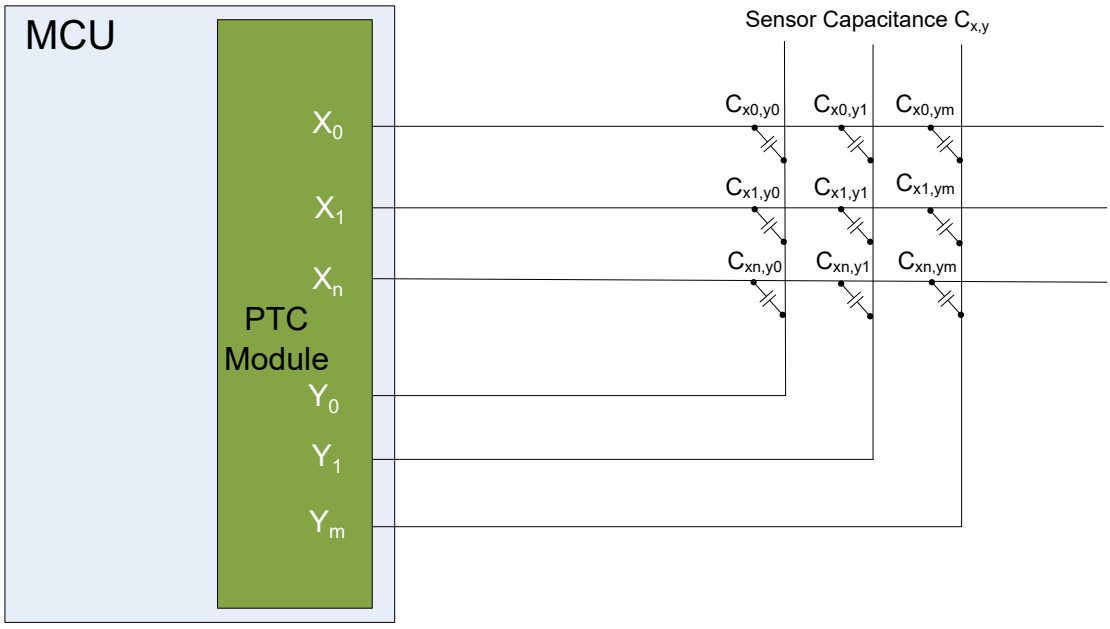
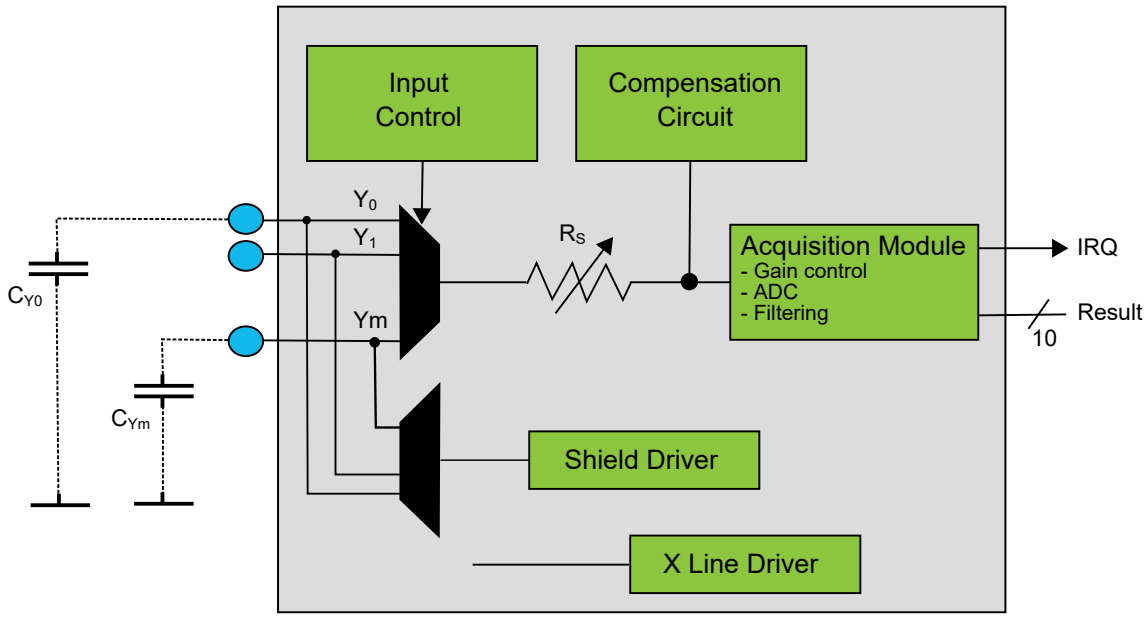


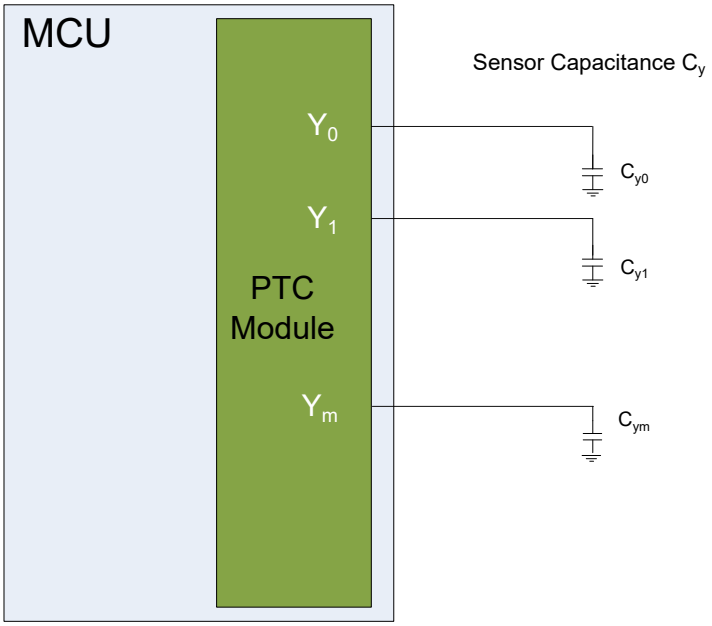
Figure 48-3. PTC Block Diagram Self Capacitance



Note: For PIC32CM LE00/LS00/LS60 the RS = 100 KΩ.

A self-capacitance sensor is connected to a single pin on the Peripheral Touch Controller through the Y electrode for sensing the signal. The sense electrode capacitance is measured by the Peripheral Touch Controller.

Figure 48-4. Self-Capacitance Sensor Arrangement



For more information about designing the touch sensor, refer to Buttons, Sliders and Wheels Touch Sensor Design Guide.

48.4 Signal Description

Table 48-1. Signal Description for PTC

Name	Type	Description
Y[m:0]	Analog	Y-line (Input/Output)
X[n:0]	Digital	X-line (Output)

Note: The number of X- and Y-lines are device dependent. Refer to Configuration Summary for details.

Refer to the Pinout for details on the pin mapping for this peripheral. One signal can be mapped on several pins.

48.5 Peripheral Dependencies

Table 48-2. PTC Configuration Summary

Peripheral name	Base address	NVIC IRQ Index	MCLK AHB/APB Clocks	GCLK Peripheral Channel Index (GCLK.PCHCTRL)	PAC Peripheral Identifier Index (PAC.WRCTRL)	DMA Trigger Source Index (DMAC.CHCTRLB)	Events (EVSYS)		Power Domain (PM.STDBYCFG)
							Users (EVSYS.USER)	Generators (EVSYS.CHANNEL)	
PTC	0x42004000	67: EOC, WCOMP	CLK_PTC_APB	31: GCLK_PTC	80	44 : EOC 45 : SEQ 46 : WCOMP	46 : STCONV 47 : DSEQR	84: EOC 85: WCOMP	PDSW

48.6 Functional Description

In order to access the PTC, the user must use the MPLAB Harmony v3 Configurator (MHC) to configure and link the Touch Library firmware with the application software. Touch Library can be used to implement buttons, sliders, wheels in a variety of combinations on a single interface.

For more information about Touch Library, refer to the Touch Modular Library Peripheral Touch Controller User's Guide.

49. Electrical Characteristics

49.1 Disclaimer

Unless otherwise specified:

- Typical data are measured at T_A = 25°C. They are for design guidance only and are not tested.
- All minimum and maximum values are valid across operating temperature and voltage

49.2 Absolute Maximum Ratings

Absolute maximum ratings are listed below. 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 49-1. Absolute Maximum Electrical Characteristics

Absolute Maximum Ratings (1)	
Ambient temperature under bias	-40°C to +125°C
Storage temperature	-60°C to +150°C
Voltage on (VDD/AVDD) with respect to VSS	-0.3V to +3.80V
Voltage on any pins, with respect to VSS (2)	-0.3V to (VDD+0.3V) -0.3V to (AVDD+0.3V)
Maximum current out of VSS pins	357 mA
Maximum current into (VDD/AVDD) pins (3)	184 mA
Maximum output current sourced/sunk by any Non-High Sink I/O pin (4)	27 mA
Maximum output current sourced/sunk by any High Sink I/O pin (4)	50 mA
Maximum current sunk by all ports	287 mA
Maximum current sourced by all ports (3)	114 mA
Maximum Junction Temperature	+145°C
ESD Qualification	
Human Body Model (HBM) per JESD22-A114	2000 V
Charged Device Model (CDM) (ANSI/ESD STM 5.3.1) (All pins / Corner pins)	500 V / 750 V

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. When applying higher or lower voltage than those specified on IO pins, please refer to DI_17 / DI_19 to respect the maximum injection current specification.
3. Maximum allowable current is a function of device maximum power dissipation (See Thermal Operating Conditions).
4. Refer to I/O Pin Electrical Specifications to get High Sink I/O pins list.

49.3 General Operating Ratings and Thermal Conditions

Table 49-2. Operating Frequency vs. Voltage (PIC32CM LE00 / PIC32CM LS00)

Param. No.	VDD, AVDD Range	Temp. Range (in °C)	Max CPU Frequency	Comments
DC_5	1.62V to 3.63V (1)	-40°C to +85°C	12 MHz	Industrial Performance Level 0 Mode (PL0)
DC_5A			48 MHz	Industrial Performance Level 2 Mode (PL2)
Note: The same voltage must be applied to VDD and AVDD.				

Table 49-3. Operating Frequency vs. Voltage (PIC32CM LS60)

Param. No.	VDD, AVDD Range	Temp. Range (in °C)	Max CPU Frequency	Comments
DC_15	2.0V to 3.63V (1)	-40°C to +85°C	12 MHz	Industrial Performance Level 0 Mode (PL0)
DC_15A			48 MHz	Industrial Performance Level 2 Mode (PL2)
Note: The same voltage must be applied to VDD and AVDD.				

Table 49-4. Thermal Operating Conditions

Rating	Symbol	Min.	Typ.	Max.	Unit
Industrial Temperature Devices: Operating Ambient Temperature Range Operating Junction Temperature Range	 T_A T_J	 -40 —	 — —	 85 105	°C
Power Dissipation: Internal Chip Power Dissipation: $P_{INT} = VDD \times (IDD - \Sigma IOH_{VDD}) + AVDD \times (IDDANA - \Sigma IOH_{AVDD})$ **I/O Pin Power Dissipation:** $P_{I/O} = \Sigma ((VDD - VOH) \times IOH_{VDD}) + \Sigma (VOL \times IOL_{VDD}) + \Sigma ((AVDD - VOH) \times IOH_{AVDD}) + \Sigma (VOL \times IOL_{AVDD})$	P_D	$P_{INT} + P_{I/O}$			W
Maximum Allowed Power Dissipation	P_{DMAX}	$(T_J - T_A)/\theta_{JA}$			W

Table 49-5. Thermal Packaging Characteristics (PIC32CM LE00/LS00) (1)

Characteristics	Symbol	Typ.	Max.	Unit
Thermal Resistance, 48-pin TQFP (7x7x1.0 mm) Package	θ_{JA}	50	-	°C/W
Thermal Resistance, 48-pin VQFN (7x7x0.9 mm) Package		25	-	
Thermal Resistance, 64-pin TQFP (10x10x1.0 mm) Package		46	-	
Thermal Resistance, 64-pin VQFN (9x9x1.0 mm) Package		22	-	
Thermal Resistance, 100-pin TQFP (14x14x1.0 mm) Package		42	-	

Note:
1. Junction to ambient thermal resistance, Theta-JA (θ_{JA}) numbers are achieved by package simulations.

Table 49-6. Thermal Packaging Characteristics (PIC32CM LS60) (1)

Characteristics	Symbol	Typ.	Max.	Unit
Thermal Resistance, 48-pin VQFN (7x7x0.9 mm) Package	θ_{JA}	30	-	°C/W
Thermal Resistance, 64-pin TQFP (10x10x1.0 mm) Package		51	-	
Thermal Resistance, 64-pin VQFN (9x9x1.0 mm) Package		27	-	
Thermal Resistance, 100-pin TQFP (14x14x1.0 mm) Package		47	-	

Note:

1. Junction to ambient thermal resistance, Theta-JA (θ_{JA}) numbers are achieved by package simulations.

49.4 Power Supply

Table 49-7. Power Supply Electrical Specifications

DC CHARACTERISTICS			Standard Operating Conditions: VDD=AVDD = 1.62V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ T_A ≤ +85°C for Industrial				
Param. No.	**Symbol**	**Characteristics**	**Min.**	**Typ.**	**Max.**	**Units**	**Conditions**
REG_1	VDDCORE_C_{IN} (5)	VDDCORE Input Bypass parallel Capacitor pair	0.8	1	1.2	µF	Bulk Ceramic or solid Tantalum with ESR <0.5Ω
			80	100	—	nF	Ceramic XR7 with ESR <0.5Ω
REG_3	VDDPLL_C_{IN} (5)	VDDPLL Input Bypass parallel Capacitor pair	0.8	1	1.2	µF	Bulk Ceramic or solid Tantalum with ESR <0.5Ω
			80	100	—	nF	Ceramic XR7 with ESR <0.5Ω
REG_5	VDD_C_{IN} (5)	VDD Input Bypass parallel Capacitor pair	80	100	—	nF	Ceramic XR7 with ESR <0.5Ω **on all VDD pins**
REG_9	VREFx_C_{IN} (5)	External VREFA/VREFB Input Bypass parallel Capacitor pair	3.76	4.7	—	µF	Bulk Ceramic or solid Tantalum with ESR <0.5Ω
			80	100	—	nF	Ceramic XR7 with ESR <0.5Ω
REG_17	AVDD_C_{IN} (5)	AVDD Input Bypass parallel Capacitor pair	8	10	—	µF	Bulk Ceramic or solid Tantalum with ESR <0.5Ω
			80	100	—	nF	Ceramic XR7 with ESR <0.5Ω **on all AVDD pins**
REG_23	AVDD_L_{EXT} (1)	AVDD series Ferrite Bead DCR (DC Resistance)	—	—	0.1	Ω	≥ 600 Ohms @ 100MHz
REG_25		Ferrite Bead Current Rating	500	—	—	mA	—
REG_27	VDDOUT_LEXT (2,3)	Buck Switch Mode Regulator Inductor Inductance (If LDO Mode not used)	8	10	12	µH	Shielded Inductor **ONLY** (If used in BUCK mode else No Connect)
REG_29		Inductor DCR (DC Resistance)	—	—	0.7	Ω	—
REG_31		Inductor ISAT Rating	275	—	—	mA	—
REG_33	BUCK_P_{EFF}	Buck Mode Power Efficiency	—	86	—	%	IOUT = 5mA
REG_35			—	85	—	%	IOUT = 50mA
REG_36	VDDCORE	VDDCORE Voltage Range (PL0 mode)	—	0.9	—	V	CPU in Active or Idle Mode
		VDDCORE Voltage Range (PL2 mode)	—	1.2	—		

.....continued

DC CHARACTERISTICS				Standard Operating Conditions: VDD=AVDD = 1.62V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ T_A ≤ +85°C for Industrial				
Param. No.	**Symbol**	**Characteristics**		**Min.**	**Typ.**	**Max.**	**Units**	**Conditions**
REG_36A	$V_{SCALING\_STEP}$	VDDCORE / VDDPLL Voltage Scaling Step Size		—	5	—	mV	Step size for PL0 to PL2 (or PL2 to PL0) transition
REG_37	VDD (4)	VDD Input Voltage Range		1.62	3.3	3.63	V	PIC32CM LE00 / PIC32CM LS00
				2.0				PIC32CM LS60
REG_38	VDDPLL	VDDPLL Voltage Range (PL0 mode)		—	0.9	—	V	CPU in Active or Idle Mode
		VDDPLL Voltage Range (PL2 mode)		—	1.2	—		
REG_39	AVDD (4)	AVDD Input Voltage Range		1.62	3.3	3.63	V	PIC32CM LE00 / PIC32CM LS00
				2.0				PIC32CM LS60
REG_43	SVDD_R	VDD/AVDD Rise Ramp Rate to ensure internal Power-on Reset Signal		—	—	0.1	V/µs	Failure to meet this specification may lead to start-up or unexpected behaviors
REG_44	SVDD_F	VDD/AVDD Falling Ramp Rate to ensure internal Power-on Reset Signal		—	—	0.05	V/µs	Failure to meet this specification may cause the device to not detect reset
REG_45	VPOR	Power-on Reset		0.6	—	1.63	V	VDD/AVDD Power up/Down
REG_47	VBOD33 (6,7)	BOD33.VREFSEL=0		1.57	—	1.60	V	BOD33.LEVEL = 0x4 (Default Factory Value)
				3.60	—	3.68		BOD33.LEVEL = 0x3F
		BOD33.VREFSEL=1		1.64	—	1.79		BOD33.LEVEL = 0x11
				2.91	—	3.18		BOD33.LEVEL = 0x3F
REG_51	$VBOD33_{LEVEL\_STEP}$	VBOD33 step size (BOD33.LEVEL)	BOD33.VREFSEL=0	—	34	—	mV	Step size
			BOD33.VREFSEL=1	—	28	—		
REG_52	$VBOD33_{HYST\_STEP}$	VBOD33 Hysteresis step size (BOD33.HYST)		—	Note (7)	—	mV	Step size
REG_53	T_{RST}	External RESET valid active pulse width		1	—	—	µs	Minimum reset active time to guarantee CPU reset

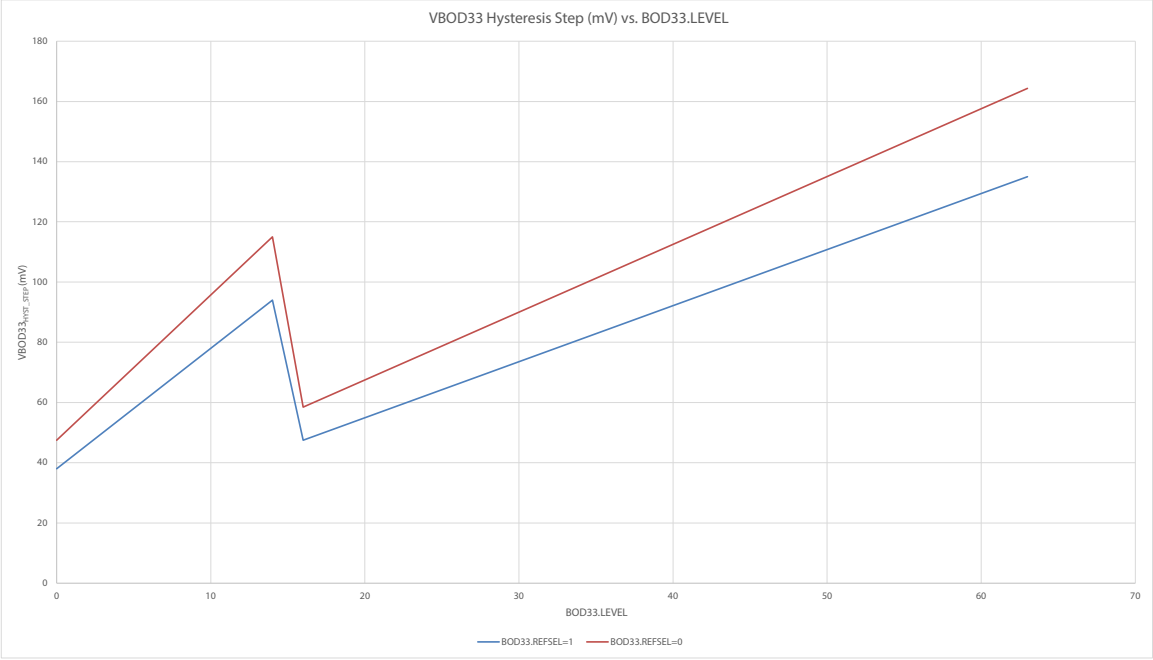
.....continued

DC CHARACTERISTICS			Standard Operating Conditions: VDD=AVDD = 1.62V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ T_A ≤ +85°C for Industrial				
Param. No.	Symbol	Characteristics	Min.	Typ.	Max.	Units	Conditions

Notes:

1. Ferrite Bead ISAT(min) ≥ (IDDANA(max) * 1.15).
2. Buck Inductor ISAT(min) ≥ ((ICAPACITOR + IVDDCORE_MAX) * 1.2) when BUCK mode enabled.
3. User must select either LDO or BUCK Mode. The modes are mutually exclusive.
4. VDD and AVDD must be at the same voltage level.
5. All bypass caps should be located immediately adjacent to pins and on the same side of the PCB as the MCU. Each primary power supply group VDD, AVDD, VDDCORE and VDDPLL should have one bulk capacitor and all power pins everywhere a 100nF bypass cap. Min and max represent absolute values including cap tolerances.
6. VBOD33(min):
   - BOD33.VREFSEL=0 => VBOD33(min) = 1.43 + (BOD33.LEVEL[5:0]) * 0.0344
   - BOD33.VREFSEL=1 => VBOD33(min) = 1.17 + (BOD33.LEVEL[5:0]) * 0.0276
7. VBOD33(max):
   (VBOD33(max)@BOD33.HYST=1) = (VBOD33(max)@BOD33.HYST=0) + $VBOD33_{HYST\_STEP}$

   where $VBOD33_{HYST\_STEP}$ can be obtained for both BOD33.VREFSEL settings using the following graph:



49.5 MCU Active Current Consumption

Table 49-8. MCU Active Current Consumption Electrical Specifications (1, 2, 3, 4)

DC CHARACTERISTICS				Standard Operating Conditions: VDD = AVDD = 1.62V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ T_A ≤ +85°C for Industrial			
Param. No.	**Symbol**	**Characteristics**	**Clock/Freq**	**Typ.**	**Max.**	**Units**	**Conditions**
APWR_1	I_{DD_ACTIVE}	Current consumption in Active mode (LDO mode)	FDPLL96M=12MHz (48 MHz/4)	120	145	µA/MHz	VDD = AVDD = 3.3V XOSC32K @ 32.768 kHz Performance Level Mode = PL0
APWR_2				119	145		VDD = AVDD = 1.8V XOSC32K @ 32.768 kHz Performance Level Mode = PL0
APWR_3			DFLLULP=12MHz	88	123		VDD = AVDD = 3.3V XOSC32K @ 32.768 kHz Performance Level Mode = PL0
APWR_4				88	124		VDD = AVDD = 1.8V XOSC32K @ 32.768 kHz Performance Level Mode = PL0
APWR_5			OSC16M=12MHz	93	114		VDD = AVDD = 3.3V XOSC32K OFF Performance Level Mode = PL0
APWR_6				91	113		VDD = AVDD = 1.8V XOSC32K OFF Performance Level Mode = PL0
APWR_7			OSC16M=4MHz	106	168		VDD = AVDD = 3.3V XOSC32K OFF Performance Level Mode = PL0
APWR_8				103	164		VDD = AVDD = 1.8V XOSC32K OFF Performance Level Mode = PL0

.....continued

DC CHARACTERISTICS				Standard Operating Conditions: VDD = AVDD = 1.62V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ T_A ≤ +85°C for Industrial			
Param. No.	Symbol	Characteristics	Clock/Freq	Typ.	Max.	Units	Conditions
APWR_9	I_{DD_ACTIVE}	Current consumption in Active mode (LDO mode)	FDPLL96M=48MHz	114	126	µA/MHz	VDD = AVDD = 3.3V XOSC32K @ 32.768 kHz Performance Level Mode = PL2
APWR_10				114	126		VDD = AVDD = 1.8V XOSC32K @ 32.768 kHz Performance Level Mode = PL2
APWR_11			DFLL48M=48MHz	111	121		VDD = AVDD = 3.3V XOSC32K @ 32.768 kHz Performance Level Mode = PL2
APWR_12				110	121		VDD = AVDD = 1.8V XOSC32K @ 32.768 kHz Performance Level Mode = PL2
APWR_13			DFLLULP=32MHz	109	135		VDD = AVDD = 3.3V XOSC32K @ 32.768 kHz Performance Level Mode = PL2
APWR_14				109	135		VDD = AVDD = 1.8V XOSC32K @ 32.768 kHz Performance Level Mode = PL2
APWR_15		Current consumption in Active mode (BUCK mode)	FDPLL96M=12MHz (48 MHz/4)	63	85		VDD = AVDD = 3.3V XOSC32K @ 32.768 kHz Performance Level Mode = PL0
APWR_16				85	105		VDD = AVDD = 1.8V XOSC32K @ 32.768 kHz Performance Level Mode = PL0
APWR_17			DFLLULP=12MHz	38	59		VDD = AVDD = 3.3V XOSC32K @ 32.768 kHz Performance Level Mode = PL0
APWR_18				58	86		VDD = AVDD = 1.8V XOSC32K @ 32.768 kHz Performance Level Mode = PL0

.....continued

DC CHARACTERISTICS				Standard Operating Conditions: VDD = AVDD = 1.62V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ T_A ≤ +85°C for Industrial			
Param. No.	Symbol	Characteristics	Clock/Freq	Typ.	Max.	Units	Conditions
APWR_19	I_{DD_ACTIVE}	Current consumption in Active mode (BUCK mode)	OSC16M=12MHz	41	56	µA/MHz	VDD = AVDD = 3.3V XOSC32K OFF Performance Level Mode = PL0
APWR_20				61	79		VDD = AVDD = 1.8V XOSC32K OFF Performance Level Mode = PL0
APWR_21			OSC16M=4MHz	52	84		VDD = AVDD = 3.3V XOSC32K OFF Performance Level Mode = PL0
APWR_22				70	114		VDD = AVDD = 1.8V XOSC32K OFF Performance Level Mode = PL0
APWR_23			FDPLL96M=48MHz	58	71		VDD = AVDD = 3.3V XOSC32K @ 32.768 kHz Performance Level Mode = PL2
APWR_24				89	99		VDD = AVDD = 1.8V XOSC32K @ 32.768 kHz Performance Level Mode = PL2
APWR_25			DFLL48M=48MHz	54	67		VDD = AVDD = 3.3V XOSC32K @ 32.768 kHz Performance Level Mode = PL2
APWR_26				86	94		VDD = AVDD = 1.8V XOSC32K @ 32.768 kHz Performance Level Mode = PL2
APWR_27			DFLLULP=32MHz	52	75		VDD = AVDD = 3.3V XOSC32K @ 32.768 kHz Performance Level Mode = PL2
APWR_28				83	107		VDD = AVDD = 1.8V XOSC32K @ 32.768 kHz Performance Level Mode = PL2

.....continued

DC CHARACTERISTICS				Standard Operating Conditions: VDD = AVDD = 1.62V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ T_A ≤ +85°C for Industrial			
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 (i.e. all peripherals inactive)
   - NVM Cache enabled
   - All clock generation sources disabled unless otherwise specified (i.e. XOSC32=Off...)
   - All I/O pins are tri-stated with input buffers disabled
   - WDT, RTC, CFD Clock Fail Detection disabled
   - $\overline{\text{RESET}}$ = VDD
   - Note: µA/Mhz varies over temperature. Worst case is given by max.
3. CPU Running CoreMark® Test Suite.
4. For PIC32CM LS60 MCUs, the current consumption of the ATECC608B is not taken into account and must be added. Refer to the ATECC608B-TFLXTLS CryptoAuthentication™ Data Sheet (DS40002138) for additional information.

Figure 49-1. Typical Power Consumption over Temperature in Active Mode (PL0 @12MHz)

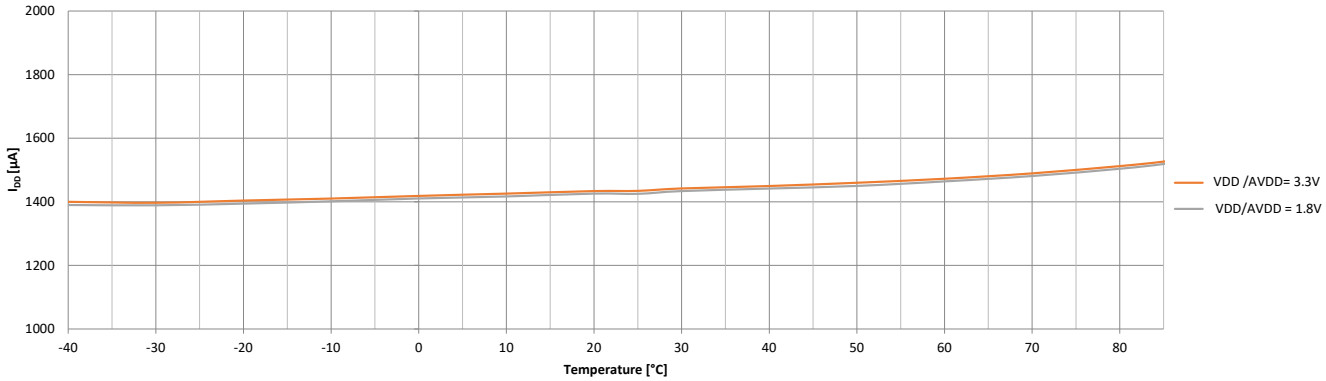
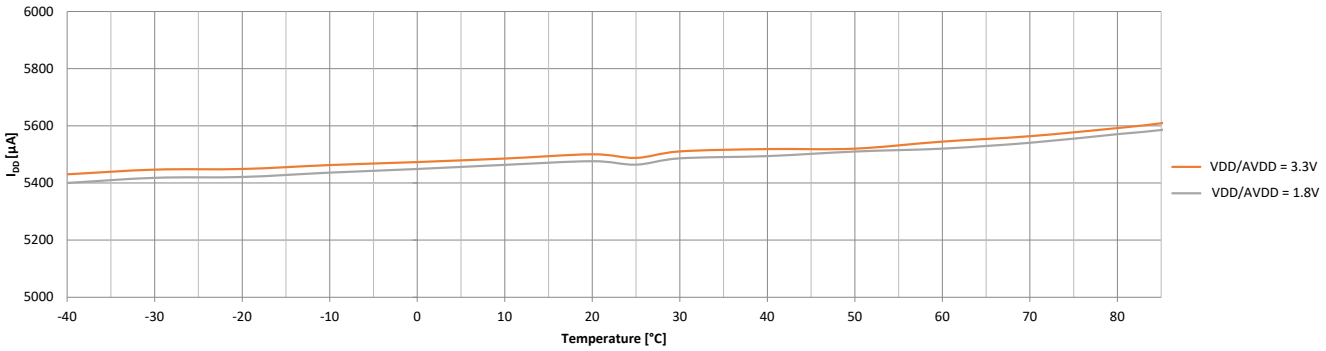


Figure 49-2. Power Consumption over Temperature in Active Mode (PL2 @48MHz)



49.6 MCU Idle Current Consumption

Table 49-9. MCU Idle Current Consumption Electrical Specifications (1, 2, 3)

DC CHARACTERISTICS				Standard Operating Conditions: VDD = AVDD = 1.62V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ T_A ≤ +85°C for Industrial			
Param. No.	**Symbol**	**Characteristics**	**Clock/Freq**	**Typ.**	**Max.**	**Units**	**Conditions**
IPWR_1	I_{DD_IDLE}	Current consumption in IDLE mode (LDO mode)	FDPLL96M=12MHz (48 MHz/4)	51	75	µA/MHz	VDD = AVDD = 3.3V XOSC32K @ 32.768 kHz Performance Level Mode = PL0
IPWR_2				50	74		VDD = AVDD = 1.8V XOSC32K @ 32.768 kHz Performance Level Mode = PL0
IPWR_3			DFLLULP=12MHz	20	43		VDD = AVDD = 3.3V XOSC32K @ 32.768 kHz Performance Level Mode = PL0
IPWR_4				20	43		VDD = AVDD = 1.8V XOSC32K @ 32.768 kHz Performance Level Mode = PL0
IPWR_5			OSC16M=12MHz	24	44		VDD = AVDD = 3.3V XOSC32K OFF Performance Level Mode = PL0
IPWR_6				23	43		VDD = AVDD = 1.8V XOSC32K OFF Performance Level Mode = PL0
IPWR_7			OSC16M=4MHz	36	96		VDD = AVDD = 3.3V XOSC32K OFF Performance Level Mode = PL0
IPWR_8				33	93		VDD = AVDD = 1.8V XOSC32K OFF Performance Level Mode = PL0

.....continued

DC CHARACTERISTICS				Standard Operating Conditions: VDD = AVDD = 1.62V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ T_A ≤ +85°C for Industrial			
Param. No.	Symbol	Characteristics	Clock/Freq	Typ.	Max.	Units	Conditions
IPWR_9	I_{DD_IDLE}	Current consumption in IDLE mode (LDO mode)	FDPLL96M=48MHz	25	33	µA/MHz	VDD = AVDD = 3.3V XOSC32K @ 32.768 kHz Performance Level Mode = PL2
IPWR_10				25	33		VDD = AVDD = 1.8V XOSC32K @ 32.768 kHz Performance Level Mode = PL2
IPWR_11			DFLL48M=48MHz	22	28		VDD = AVDD = 3.3V XOSC32K @ 32.768 kHz Performance Level Mode = PL2
IPWR_12				22	28		VDD = AVDD = 1.8V XOSC32K @ 32.768 kHz Performance Level Mode = PL2
IPWR_13			DFLLULP=32MHz	21	33		VDD = AVDD = 3.3V XOSC32K @ 32.768 kHz Performance Level Mode = PL2
IPWR_14				21	33		VDD = AVDD = 1.8V XOSC32K @ 32.768 kHz Performance Level Mode = PL2
IPWR_15		Current consumption in IDLE mode (BUCK mode)	FDPLL96M=12MHz (48 MHz/4)	36	51		VDD = AVDD = 3.3V XOSC32K @ 32.768 kHz Performance Level Mode = PL0
IPWR_16				41	59		VDD = AVDD = 1.8V XOSC32K @ 32.768 kHz Performance Level Mode = PL0
IPWR_17			DFLLULP=12MHz	11	23		VDD = AVDD = 3.3V XOSC32K @ 32.768 kHz Performance Level Mode = PL0
IPWR_18				14	30		VDD = AVDD = 1.8V XOSC32K @ 32.768 kHz Performance Level Mode = PL0

.....continued

DC CHARACTERISTICS				Standard Operating Conditions: VDD = AVDD = 1.62V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ T_A ≤ +85°C for Industrial			
Param. No.	**Symbol**	**Characteristics**	**Clock/Freq**	**Typ.**	**Max.**	**Units**	**Conditions**
IPWR_19	I_{DD_IDLE}	Current consumption in IDLE mode (BUCK mode)	OSC16M=12MHz	14	24	µA/MHz	VDD = AVDD = 3.3V XOSC32K OFF Performance Level Mode = PL0
IPWR_20				17	30		VDD = AVDD = 1.8V XOSC32K OFF Performance Level Mode = PL0
IPWR_21			OSC16M=4MHz	24	51		VDD = AVDD = 3.3V XOSC32K OFF Performance Level Mode = PL0
IPWR_22				26	64		VDD = AVDD = 1.8V XOSC32K OFF Performance Level Mode = PL0
IPWR_23			FDPLL96M=48MHz	16	23		VDD = AVDD = 3.3V XOSC32K @ 32.768 kHz Performance Level Mode = PL2
IPWR_24				21	27		VDD = AVDD = 1.8V XOSC32K @ 32.768 kHz Performance Level Mode = PL2
IPWR_25			DFLL48M=48MHz	12	18		VDD = AVDD = 3.3V XOSC32K @ 32.768 kHz Performance Level Mode = PL2
IPWR_26				18	23		VDD = AVDD = 1.8V XOSC32K @ 32.768 kHz Performance Level Mode = PL2
IPWR_27			DFLLULP=32MHz	12	20		VDD = AVDD = 3.3V XOSC32K @ 32.768 kHz Performance Level Mode = PL2
IPWR_28				17	27		VDD = AVDD = 1.8V XOSC32K @ 32.768 kHz Performance Level Mode = PL2

Notes:

1. Typical values at 25°C only.
2. Conditions:
   - No peripheral modules are operating (i.e. all peripherals inactive)
   - All clock generation sources disabled unless otherwise specified (i.e. XOSC32=Off...)
   - All I/O pins are tri-stated with input buffers disabled
   - WDT, RTC, CFD Clock Fail Detection disabled
   - $\overline{\text{RESET}}$ = VDD
   - Note: µA/Mhz varies over temperature. Worst case is given by max.
3. For PIC32CM LS60 MCUs, the current consumption of the ATECC608B is not taken into account and must be added. Refer to the ATECC608B-TFLXTLS CryptoAuthentication™ Data Sheet (DS40002138) for additional information.

49.7 MCU Standby Current Consumption

Table 49-10. MCU Standby Current Consumption Electrical Specifications (1,2)

DC CHARACTERISTICS			Standard Operating Conditions: VDD = AVDD = 1.62V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ T_A ≤ +85°C for Industrial					
Param. No.	Symbol	Characteristics	VDD/ AVDD	T_A	Typ.	Max.	Units	Conditions
SPWR_1	$I_{DD\_STANDBY}$	Current consumption in Standby mode 64 kB SRAM retained PDSW Domain Active	3.3V	25°C	1.68	5.10	µA	LPVREG with VREG.LPEFF enable
SPWR_2				85°C	36.47	97.11		
SPWR_3			1.8V	25°C	2.07	6.70		LPVREG with VREG.LPEFF disable
SPWR_4				85°C	56.51	164.57		
SPWR_5			3.3V	25°C	1.52	4.02		BUCK in Standby with MAINVREG in PL0 Mode (VREG.RUNSTDBY = 1 and VREG.STDBYPL0 = 1)
SPWR_6				85°C	26.53	81.62		
SPWR_7			1.8V	25°C	1.57	5.72		
SPWR_8				85°C	40.30	113.67		
SPWR_97		Current consumption in Standby mode 64 kB SRAM retained PDSW Domain in Retention	3.3V	25°C	1.05	2.89	µA	LPVREG with VREG.LPEFF enable
SPWR_98				85°C	21.17	52.06		
SPWR_99			1.8V	25°C	1.25	3.46		LPVREG with VREG.LPEFF disable
SPWR_100				85°C	27.88	84.56		
SPWR_101			3.3V	25°C	1.09	2.52		BUCK in Standby with MAINVREG in PL0 Mode (VREG.RUNSTDBY = 1 and VREG.STDBYPL0 = 1)
SPWR_102				85°C	15.30	38.30		
SPWR_103			1.8V	25°C	1.11	2.96		
SPWR_104				85°C	22.90	58.21		
SPWR_9		Current consumption in Standby mode 64 kB SRAM retained PDSW Domain Active RTC running on XOSC32K	3.3V	25°C	2.09	5.56	µA	LPVREG with VREG.LPEFF enable
SPWR_10				85°C	37.06	97.66		
SPWR_11			1.8V	25°C	2.50	7.14		LPVREG with VREG.LPEFF disable
SPWR_12				85°C	57.43	165.48		
SPWR_13			3.3V	25°C	1.83	4.38		BUCK in Standby with MAINVREG in PL0 Mode (VREG.RUNSTDBY = 1 and VREG.STDBYPL0 = 1)
SPWR_14				85°C	27.01	82.34		
SPWR_15			1.8V	25°C	1.98	6.10		
SPWR_16				85°C	40.79	114.23		

.....continued

DC CHARACTERISTICS			Standard Operating Conditions: VDD = AVDD = 1.62V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ T_A ≤ +85°C for Industrial					
Param. No.	Symbol	Characteristics	VDD/ AVDD	T_A	Typ.	Max.	Units	Conditions
SPWR_105	$I_{DD\_STANDBY}$	Current consumption in Standby mode 64 kB SRAM retained PDSW Domain in Retention RTC running on XOSC32K	3.3V	25°C	1.46	3.32	µA	LPVREG with VREG.LPEFF enable
SPWR_106				85°C	21.73	52.70		
SPWR_107			1.8V	25°C	1.68	3.88		LPVREG with VREG.LPEFF disable
SPWR_108				85°C	28.74	85.42		
SPWR_109			3.3V	25°C	1.40	2.89		BUCK in Standby with MAINVREG in PL0 Mode (VREG.RUNSTDBY = 1 and VREG.STDBYPL0 = 1)
SPWR_110				85°C	15.81	38.90		
SPWR_111			1.8V	25°C	1.38	3.36		
SPWR_112				85°C	23.37	58.87		
SPWR_17		Current consumption in Standby mode 64 kB SRAM retained PDSW Domain Active RTC running on OSCULP32K	3.3V	25°C	1.75	5.24	µA	LPVREG with VREG.LPEFF enable
SPWR_18				85°C	36.59	97.27		
SPWR_19			1.8V	25°C	2.15	6.81		LPVREG with VREG.LPEFF disable
SPWR_20				85°C	57.01	165.07		
SPWR_21			3.3V	25°C	1.55	4.02		BUCK in Standby with MAINVREG in PL0 Mode (VREG.RUNSTDBY = 1 and VREG.STDBYPL0 = 1)
SPWR_22				85°C	26.61	81.77		
SPWR_23			1.8V	25°C	1.62	6.00		
SPWR_24				85°C	40.37	113.71		
SPWR_113		Current consumption in Standby mode 64 kB SRAM retained PDSW Domain in Retention RTC running on OSCULP32K	3.3V	25°C	1.11	2.98	µA	LPVREG with VREG.LPEFF Enable
SPWR_114				85°C	21.27	52.27		
SPWR_115			1.8V	25°C	1.33	3.53		LPVREG with VREG.LPEFF Disable
SPWR_116				85°C	28.20	84.99		
SPWR_117			3.3V	25°C	1.20	2.57		BUCK in Standby with MAINVREG in PL0 Mode (VREG.RUNSTDBY = 1 and VREG.STDBYPL0 = 1)
SPWR_118				85°C	15.34	38.33		
SPWR_119			1.8V	25°C	1.14	3.02		
SPWR_120				85°C	22.91	58.54		

.....continued

DC CHARACTERISTICS			Standard Operating Conditions: VDD = AVDD = 1.62V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ T_A ≤ +85°C for Industrial					
Param. No.	Symbol	Characteristics	VDD/ AVDD	T_A	Typ.	Max.	Units	Conditions
SPWR_25	$I_{DD\_STANDBY}$	Current consumption in Standby mode 48 kB SRAM retained PDSW Domain Active	3.3V	25°C	1.60	5.04	µA	LPVREG with VREG.LPEFF enable
SPWR_26				85°C	33.99	93.58		
SPWR_27			1.8V	25°C	1.96	6.51		LPVREG with VREG.LPEFF disable
SPWR_28				85°C	51.82	158.92		
SPWR_29			3.3V	25°C	1.38	3.92		BUCK in Standby with MAINVREG in PL0 Mode (VREG.RUNSTDBY = 1 and VREG.STDBYPL0 = 1)
SPWR_30				85°C	24.73	78.93		
SPWR_31			1.8V	25°C	1.53	5.58		
SPWR_32				85°C	37.41	109.46		
SPWR_121		Current consumption in Standby mode 48 kB SRAM retained PDSW Domain in Retention	3.3V	25°C	0.96	2.66	µA	LPVREG with VREG.LPEFF Enable
SPWR_122				85°C	18.64	46.81		
SPWR_123			1.8V	25°C	1.13	3.16		LPVREG with VREG.LPEFF Disable
SPWR_124				85°C	22.81	74.71		
SPWR_125			3.3V	25°C	1.15	2.38		BUCK in Standby with MAINVREG in PL0 Mode (VREG.RUNSTDBY = 1 and VREG.STDBYPL0 = 1)
SPWR_126				85°C	13.61	34.68		
SPWR_127			1.8V	25°C	1.09	2.71		
SPWR_128				85°C	19.92	52.14		
SPWR_33		Current consumption in Standby mode 48 kB SRAM retained PDSW Domain Active RTC running on XOSC32K	3.3V	25°C	2.01	5.42	µA	LPVREG with VREG.LPEFF enable
SPWR_34				85°C	34.60	94.19		
SPWR_35			1.8V	25°C	2.38	6.94		LPVREG with VREG.LPEFF disable
SPWR_36				85°C	52.85	159.96		
SPWR_37			3.3V	25°C	1.74	4.28		BUCK in StandBy with MAINVREG in PL0 Mode (VREG.RUNSTDBY = 1 and VREG.STDBYPL0 = 1)
SPWR_38				85°C	25.26	79.18		
SPWR_39			1.8V	25°C	1.96	5.96		
SPWR_40				85°C	37.89	109.89		

.....continued

DC CHARACTERISTICS			Standard Operating Conditions: VDD = AVDD = 1.62V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ T_A ≤ +85°C for Industrial					
Param. No.	Symbol	Characteristics	VDD/ AVDD	T_A	Typ.	Max.	Units	Conditions
SPWR_129	$I_{DD\_STANDBY}$	Current consumption in Standby mode 48 kB SRAM retained PDSW Domain in Retention RTC running on XOSC32K	3.3V	25°C	1.37	3.08	µA	LPVREG with VREG.LPEFF Enable
SPWR_130				85°C	19.21	47.40		
SPWR_131			1.8V	25°C	1.56	3.56		LPVREG with VREG.LPEFF Disable
SPWR_132				85°C	23.42	75.83		
SPWR_133			3.3V	25°C	1.40	2.74		BUCK in Standby with MAINVREG in PL0 Mode (VREG.RUNSTDBY = 1 and VREG.STDBYPL0 = 1)
SPWR_134				85°C	14.07	35.48		
SPWR_135			1.8V	25°C	1.37	2.96		
SPWR_136				85°C	20.46	52.66		
SPWR_41		Current consumption in Standby mode 48 kB SRAM retained PDSW Domain Active RTC running on OSCULP32K	3.3V	25°C	1.66	5.10	µA	LPVREG with VREG.LPEFF enable
SPWR_42				85°C	34.14	93.83		
SPWR_43			1.8V	25°C	2.03	6.61		LPVREG with VREG.LPEFF disable
SPWR_44				85°C	52.40	158.97		
SPWR_45			3.3V	25°C	1.45	3.96		BUCK in StandBy with MAINVREG in PL0 Mode (VREG.RUNSTDBY = 1 and VREG.STDBYPL0 = 1)
SPWR_46				85°C	24.73	78.94		
SPWR_47			1.8V	25°C	1.54	5.64		
SPWR_48				85°C	37.46	109.47		
SPWR_137		Current consumption in Standby mode 48 kB SRAM retained PDSW Domain in Retention RTC running on OSCULP32K	3.3V	25°C	1.02	2.73	µA	LPVREG with VREG.LPEFF enable
SPWR_138				85°C	18.75	46.90		
SPWR_139			1.8V	25°C	1.21	3.24		LPVREG with VREG.LPEFF disable
SPWR_140				85°C	23.00	75.09		
SPWR_141			3.3V	25°C	1.13	2.42		BUCK in Standby with MAINVREG in PL0 Mode (VREG.RUNSTDBY = 1 and VREG.STDBYPL0 = 1)
SPWR_142				85°C	13.57	34.67		
SPWR_143			1.8V	25°C	1.03	2.79		
SPWR_144				85°C	20.00	51.89		

.....continued

DC CHARACTERISTICS			Standard Operating Conditions: VDD = AVDD = 1.62V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ T_A ≤ +85°C for Industrial					
Param. No.	Symbol	Characteristics	VDD/ AVDD	T_A	Typ.	Max.	Units	Conditions
SPWR_49	$I_{DD\_STANDBY}$	Current consumption in Standby mode 32 kB SRAM retained PDSW Domain Active	3.3V	25°C	1.51	4.90	µA	LPVREG with VREG.LPEFF enable
SPWR_50				85°C	31.51	90.35		
SPWR_51			1.8V	25°C	1.83	6.40		LPVREG with VREG.LPEFF disable
SPWR_52				85°C	47.14	152.39		
SPWR_53			3.3V	25°C	1.41	3.82		BUCK in Standby with MAINVREG in PL0 Mode (VREG.RUNSTDBY = 1 and VREG.STDBYPL0 = 1)
SPWR_54				85°C	23.01	76.27		
SPWR_55			1.8V	25°C	1.39	5.38		
SPWR_56				85°C	34.44	105.26		
SPWR_145		Current consumption in Standby mode 32 kB SRAM retained PDSW Domain in Retention	3.3V	25°C	0.87	2.39	µA	LPVREG with VREG.LPEFF enable
SPWR_146				85°C	16.09	41.29		
SPWR_147			1.8V	25°C	1.01	2.80		LPVREG with VREG.LPEFF disable
SPWR_148				85°C	19.93	64.03		
SPWR_149			3.3V	25°C	1.05	2.24		BUCK in Standby with MAINVREG in PL0 Mode (VREG.RUNSTDBY = 1 and VREG.STDBYPL0 = 1)
SPWR_150				85°C	11.75	31.22		
SPWR_151			1.8V	25°C	0.91	2.44		
SPWR_152				85°C	17.10	45.03		
SPWR_57		Current consumption in Standby mode 32 kB SRAM retained PDSW Domain Active RTC running on XOSC32K	3.3V	25°C	1.92	5.28	µA	LPVREG with VREG.LPEFF enable
SPWR_58				85°C	32.11	90.89		
SPWR_59			1.8V	25°C	2.26	6.74		LPVREG with VREG.LPEFF disable
SPWR_60				85°C	48.17	153.24		
SPWR_61			3.3V	25°C	1.65	4.25		BUCK in Standby with MAINVREG in PL0 Mode (VREG.RUNSTDBY = 1 and VREG.STDBYPL0 = 1)
SPWR_62				85°C	23.48	76.56		
SPWR_63			1.8V	25°C	1.82	5.78		
SPWR_64				85°C	35.08	105.69		

.....continued

DC CHARACTERISTICS			Standard Operating Conditions: VDD = AVDD = 1.62V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ T_A ≤ +85°C for Industrial					
Param. No.	Symbol	Characteristics	VDD/ AVDD	T_A	Typ.	Max.	Units	Conditions
SPWR_153	$I_{DD\_STANDBY}$	Current consumption in Standby mode 32 kB SRAM retained PDSW Domain in Retention RTC running on XOSC32K	3.3V	25°C	1.28	2.80	µA	LPVREG with VREG.LPEFF enable
SPWR_154				85°C	16.67	41.86		
SPWR_155			1.8V	25°C	1.44	3.25		LPVREG with VREG.LPEFF disable
SPWR_156				85°C	20.53	64.87		
SPWR_157			3.3V	25°C	1.33	2.68		BUCK in Standby with MAINVREG in PL0 Mode (VREG.RUNSTDBY = 1 and VREG.STDBYPL0 = 1)
SPWR_158				85°C	12.22	31.83		
SPWR_159			1.8V	25°C	1.25	2.84		
SPWR_160				85°C	17.12	45.61		
SPWR_65		Current consumption in Standby mode 32 kB SRAM retained PDSW Domain Active RTC running on OSCULP32K	3.3V	25°C	1.57	4.97	µA	LPVREG with VREG.LPEFF enable
SPWR_66				85°C	31.65	90.53		
SPWR_67			1.8V	25°C	1.92	6.41		LPVREG with VREG.LPEFF disable
SPWR_68				85°C	47.61	152.70		
SPWR_69			3.3V	25°C	1.46	3.87		BUCK in Standby with MAINVREG in PL0 Mode (VREG.RUNSTDBY = 1 and VREG.STDBYPL0 = 1)
SPWR_70				85°C	23.07	76.06		
SPWR_71			1.8V	25°C	1.45	5.44		
SPWR_72				85°C	34.55	105.24		
SPWR_161		Current consumption in Standby mode 32 kB SRAM retained PDSW Domain in Retention RTC running on OSCULP32K	3.3V	25°C	0.94	2.46	µA	LPVREG with VREG.LPEFF enable
SPWR_162				85°C	16.20	41.41		
SPWR_163			1.8V	25°C	1.10	2.90		LPVREG with VREG.LPEFF disable
SPWR_164				85°C	20.09	64.58		
SPWR_165			3.3V	25°C	1.07	1.88		BUCK in Standby with MAINVREG in PL0 Mode (VREG.RUNSTDBY = 1 and VREG.STDBYPL0 = 1)
SPWR_166				85°C	11.28	31.22		
SPWR_167			1.8V	25°C	0.97	2.52		
SPWR_168				85°C	16.34	45.34		

.....continued

DC CHARACTERISTICS			Standard Operating Conditions: VDD = AVDD = 1.62V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ T_A ≤ +85°C for Industrial					
Param. No.	Symbol	Characteristics	VDD/ AVDD	T_A	Typ.	Max.	Units	Conditions
SPWR_73	$I_{DD\_STANDBY}$	Current consumption in Standby mode 16 kB SRAM retained PDSW Domain Active	3.3V	25°C	1.42	4.58	µA	LPVREG with VREG.LPEFF enable
SPWR_74				85°C	29.04	86.89		
SPWR_75			1.8V	25°C	1.73	5.92		LPVREG with VREG.LPEFF disable
SPWR_76				85°C	42.43	145.93		
SPWR_77			3.3V	25°C	1.30	3.61		BUCK in Standby with MAINVREG in PL0 Mode (VREG.RUNSTDBY = 1 and VREG.STDBYPL0 = 1)
SPWR_78				85°C	21.17	73.34		
SPWR_79			1.8V	25°C	1.35	5.10		
SPWR_80				85°C	31.60	100.95		
SPWR_169		Current consumption in Standby mode 16 kB SRAM retained PDSW Domain in Retention	3.3V	25°C	0.78	2.12	µA	LPVREG with VREG.LPEFF enable
SPWR_170				85°C	13.55	35.62		
SPWR_171			1.8V	25°C	0.90	2.47		LPVREG with VREG.LPEFF disable
SPWR_172				85°C	16.91	53.29		
SPWR_173			3.3V	25°C	0.93	2.09		BUCK in Standby with MAINVREG in PL0 Mode (VREG.RUNSTDBY = 1 and VREG.STDBYPL0 = 1)
SPWR_174				85°C	9.99	27.55		
SPWR_175			1.8V	25°C	0.88	2.18		
SPWR_176				85°C	14.23	38.66		
SPWR_81		Current consumption in Standby mode 16 kB SRAM retained PDSW Domain Active RTC running on XOSC32K	3.3V	25°C	1.83	4.96	µA	LPVREG with VREG.LPEFF enable
SPWR_82				85°C	29.58	87.40		
SPWR_83			1.8V	25°C	2.15	6.34		LPVREG with VREG.LPEFF disable
SPWR_84				85°C	43.16	146.51		
SPWR_85			3.3V	25°C	1.66	4.00		BUCK in Standby with MAINVREG in PL0 Mode (VREG.RUNSTDBY = 1 and VREG.STDBYPL0 = 1)
SPWR_86				85°C	21.68	73.76		
SPWR_87			1.8V	25°C	1.68	5.46		
SPWR_88				85°C	32.14	101.44		

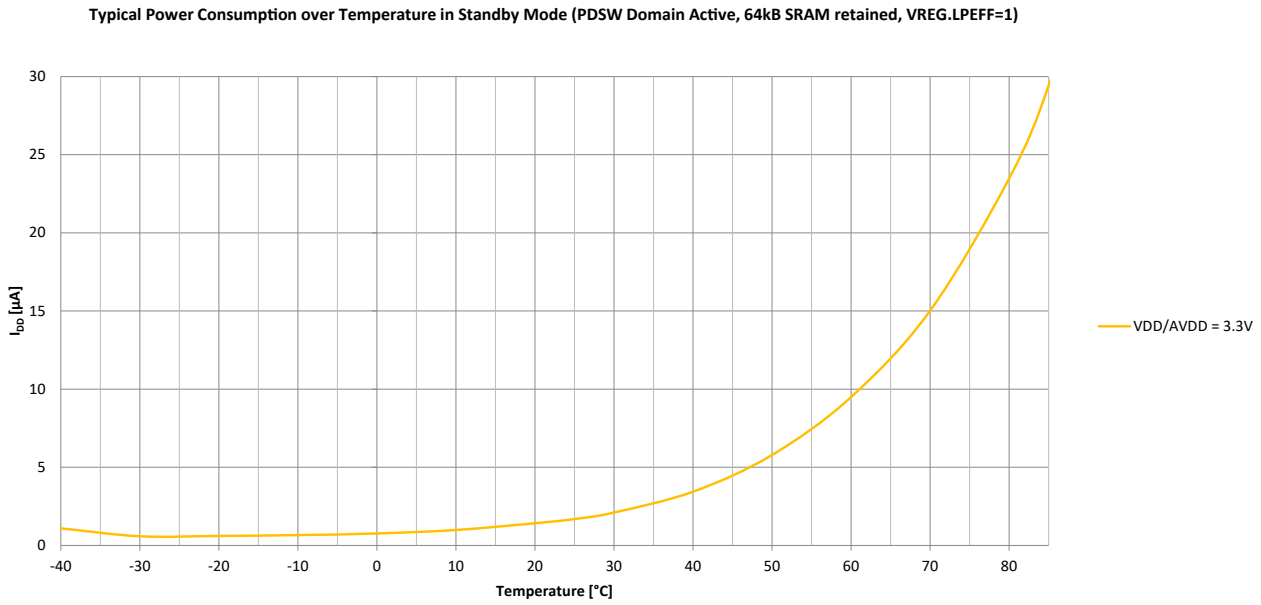
.....continued

DC CHARACTERISTICS			Standard Operating Conditions: VDD = AVDD = 1.62V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ T_A ≤ +85°C for Industrial					
Param. No.	Symbol	Characteristics	VDD/ AVDD	T_A	Typ.	Max.	Units	Conditions
SPWR_177	$I_{DD\_STANDBY}$	Current consumption in Standby mode 16 kB SRAM retained PDSW Domain in Retention RTC running on XOSC32K	3.3V	25°C	1.20	2.53	µA	LPVREG with VREG.LPEFF enable
SPWR_178				85°C	14.09	36.18		
SPWR_179			1.8V	25°C	1.33	2.90		LPVREG with VREG.LPEFF disable
SPWR_180				85°C	17.49	53.95		
SPWR_181			3.3V	25°C	1.27	2.49		BUCK in Standby with MAINVREG in PL0 Mode (VREG.RUNSTDBY = 1 and VREG.STDBYPL0 = 1)
SPWR_182				85°C	10.54	28.21		
SPWR_183			1.8V	25°C	1.13	2.58		
SPWR_184				85°C	14.74	39.02		
SPWR_89		Current consumption in Standby mode 16 kB SRAM retained PDSW Domain Active RTC running on OSCULP32K	3.3V	25°C	1.49	4.66	µA	LPVREG with VREG.LPEFF enable
SPWR_90				85°C	29.13	86.98		
SPWR_91			1.8V	25°C	1.79	6.00		LPVREG with VREG.LPEFF disable
SPWR_92				85°C	42.73	145.95		
SPWR_93			3.3V	25°C	1.35	3.67		BUCK in Standby with MAINVREG in PL0 Mode (VREG.RUNSTDBY = 1 and VREG.STDBYPL0 = 1)
SPWR_94				85°C	21.26	73.35		
SPWR_95			1.8V	25°C	1.46	5.16		
SPWR_96				85°C	31.71	101.04		
SPWR_185		Current consumption in Standby mode 16 kB SRAM retained PDSW Domain in Retention RTC running on OSCULP32K	3.3V	25°C	0.85	2.21	µA	LPVREG with VREG.LPEFF enable
SPWR_186				85°C	13.62	35.72		
SPWR_187			1.8V	25°C	0.98	2.54		LPVREG with VREG.LPEFF disable
SPWR_188				85°C	17.04	53.52		
SPWR_189			3.3V	25°C	0.96	2.25		BUCK in Standby with MAINVREG in PL0 Mode (VREG.RUNSTDBY = 1 and VREG.STDBYPL0 = 1)
SPWR_190				85°C	10.10	27.29		
SPWR_191			1.8V	25°C	0.91	2.24		
SPWR_192				85°C	14.34	38.69		

Notes:

1. Conditions:
   - All peripherals inactive unless otherwise specified
   - All clock generation sources disabled unless otherwise specified
   - All I/O pins are tri-stated with input buffers disabled
   - WDT, CFD Clock Fail Detect disabled
   - $\overline{\text{RESET}}$ = VDD
2. For PIC32CMLS60 variant, the current consumption of the ATECC608B is not taken into account and must be added according to its operating mode. Refer to the ATECC608B-TFLXTLS CryptoAuthentication™ Data Sheet (DS40002138) for additional information.

Figure 49-3. Typical Power Consumption over Temperature in Standby Mode (PDSW Domain Active, 64 kB SRAM retained, VREG.LPEFF = 1)



49.8 MCU Off Current Consumption

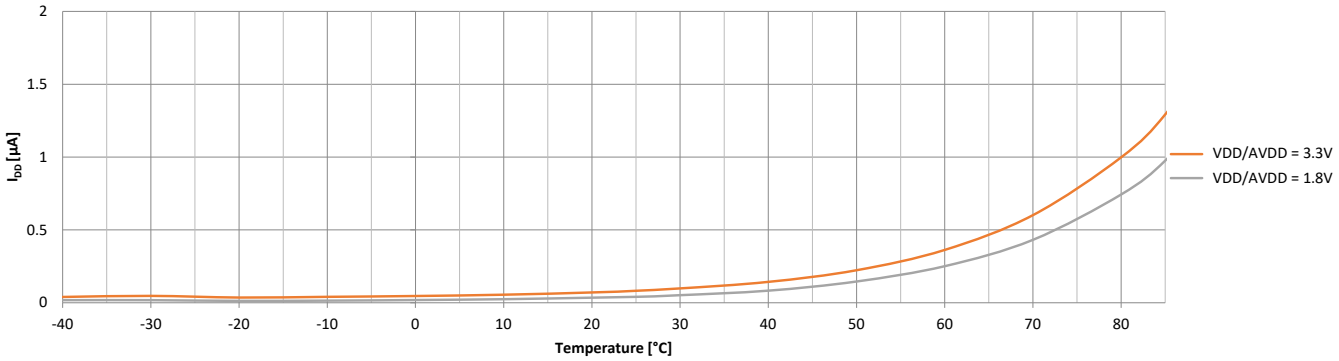
Table 49-11. MCU Off Current Consumption Electrical Specifications (1)

DC CHARACTERISTICS			Standard Operating Conditions: VDD = AVDD = 1.62V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ T_A ≤ +85°C for Industrial					
Param. No.	**Symbol**	**Characteristics**	**VDD/AVDD**	**T_A**	**Typ.**	**Max.**	**Units**	**Conditions**
OPWR_1	IDD_OFF	Current consumption in OFF mode	3.3V	25°C	84	161	nA	-
OPWR_2				85°C	1458	4163		-
OPWR_3			1.8V	25°C	43	91		-
OPWR_4				85°C	1086	3037		-

Note:

1. For PIC32CM LS60 MCUs, the current consumption of the ATECC608B is not taken into account and must be added. Refer to the "ATECC608B-TFLXTLS CryptoAuthentication™ Data Sheet" (DS40002138) for additional information.

Figure 49-4. Typical Power Consumption over Temperature in Off Mode



49.9 Wake-Up Timings Electrical Specifications

Table 49-12. Wake-Up Timings From Low Power Modes Electrical Specifications (1,2,3)

AC CHARACTERISTICS			Standard Operating Conditions: VDD = AVDD = 1.62V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ T_A ≤ +85°C for Industrial				
Param. No.	**Symbol**	**Characteristics**	**Min.**	**Typ.**	**Max.**	**Units**	**Conditions**
WKUP_1	WKUP_IDLE	Wake from IDLE mode	—	1.5	—	µs	Performance Level 0 and 2
WKUP_3	WKUP_STDBY	Wake from STANDBY Mode	—	2.7	—		PDSW Domain in Active Performance Level 0
WKUP_5			—	5.4	—		PDSW Domain in Retention Performance Level 0
WKUP_7			—	75	—		PDSW Domain in Active Performance Level 2
WKUP_9			—	76	—		PDSW Domain in Retention Performance Level 2
WKUP_10	WKUP_OFF	Wake from OFF Mode <=> Start-up Time from POR (VDD≥VDD_{min})	—	4	—	ms	PIC32CM LE
WKUP_11			—	5	—		PIC32CM LS, BOOTOPT=0
WKUP_12			—	1.7	—	s	PIC32CM LS, BOOTOPT=1, BOOTPROT=0x200
WKUP_13			—	3.4	—		PIC32CM LS, BOOTOPT=1, BOOTPROT=0x400
WKUP_14			—	6.8	—		PIC32CM LS, BOOTOPT=1, BOOTPROT=0x7FF
WKUP_15			—	1.7	—		PIC32CM LS, BOOTOPT=2, BOOTPROT=0x200
WKUP_16			—	3.4	—		PIC32CM LS, BOOTOPT=2, BOOTPROT=0x400
WKUP_17			—	6.8	—		PIC32CM LS, BOOTOPT=2, BOOTPROT=0x7FF
WKUP_18			—	1.7	—		PIC32CM LS, BOOTOPT=3, BOOTPROT=0x200
WKUP_19			—	3.4	—		PIC32CM LS, BOOTOPT=3, BOOTPROT=0x400

.....continued

AC CHARACTERISTICS			Standard Operating Conditions: VDD = AVDD = 1.62V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ T_A ≤ +85°C for Industrial				
Param. No.	Symbol	Characteristics	Min.	Typ.	Max.	Units	Conditions
WKUP_20	WKUP_OFF	Wake from OFF Mode <=> Start-up Time from POR (VDD≥$VDD_{(min)}$)	—	6.8	—	s	PIC32CM LS, BOOTOPT=3, BOOTPROT=0x7FF
WKUP_21			—	1.6	—		PIC32CM LS60, BOOTOPT=4, BOOTPROT=0x200
WKUP_22			—	2.9	—		PIC32CM LS60, BOOTOPT=4, BOOTPROT=0x400
WKUP_23			—	5.7	—		PIC32CM LS60, BOOTOPT=4, BOOTPROT=0x7FF
WKUP_24			—	1.6	—		PIC32CM LS60, BOOTOPT=5, BOOTPROT=0x200
WKUP_25			—	2.9	—		PIC32CM LS60, BOOTOPT=5, BOOTPROT=0x400
WKUP_26			—	5.7	—		PIC32CM LS60, BOOTOPT=5, BOOTPROT=0x7FF
WKUP_27			—	1.6	—		PIC32CM LS60, BOOTOPT=6, BOOTPROT=0x200
WKUP_28			—	2.9	—		PIC32CM LS60, BOOTOPT=6, BOOTPROT=0x400
WKUP_29			—	5.7	—		PIC32CM LS60, BOOTOPT=6, BOOTPROT=0x7FF

Notes:

1. Typical values at 25°C only.
2. Conditions:
   - VDD = AVDD = 3.3V
   - LDO Regulation mode
   - CPU clock = OSC16M at 4 MHz
   - One Wait-state
   - Cache enabled
   - Flash Fast Wake-up enabled (NVMCTRL.CTRLB.FWUP = 1)
   - Flash in WAKEUPINSTANT mode (NVMCTRL.CTRLB.SLEEPPRM = 1)
3. Measurement Method:
   - For IDLE and STANDBY modes, the CPU sets an I/O by writing the ARM® CPU IOBUS with interrupts disabled at ARM Core level (Cortex-M23 PRIMASK register). The wake-up time is measured between the edge of the wake-up input signal and the edge of the I/O pin.
   - For OFF mode, the exit of the mode is done through the $\overline{\text{RESET}}$ pin, the time is measured between the falling edge of the $\overline{\text{RESET}}$ signal (with the minimum reset pulse length), and the set of the I/O which is done by the first executed instructions after reset

49.10 Peripheral Active Current

Table 49-13. Peripheral Active Current Electrical Specifications (1,2)

DC CHARACTERISTICS		Standard Operating Conditions: VDD = AVDD = 1.62V to 3.63V (unless otherwise stated) Operating temperature -40°C ≤ T_A ≤ +85°C for Industrial	
Param. No.	**Symbol**	**Characteristics**	**Conditions**
MODULES/PERIPHERALS ACTIVE CURRENTS ≤ 500 nA			
PAI_01	$I_{XOSC32K}$	XOSC32K current	—
MODULES/PERIPHERALS ACTIVE CURRENTS ≤ 10 µA			
PAI_1	I_{OPAMP_MODE0}	OPAMP Active current (1 of 4)	OPAMPCTRLx.BIAS = 0
MODULES/PERIPHERALS ACTIVE CURRENTS ≤ 15 µA			
PAI_11	I_{AC_SCALER}	AC Voltage Scaler current	Voltage scaler only,
MODULES/PERIPHERALS ACTIVE CURRENTS ≤ 25 µA			
PAI_21	I_{OPAMP_MODE1}	OPAMP Active current (2 of 4)	OPAMPCTRLx.BIAS = 1
MODULES/PERIPHERALS ACTIVE CURRENTS ≤ 35 µA			
PAI_31	I_{BOD33}	BOD33 Active current	BOD in continuous mode
MODULES/PERIPHERALS ACTIVE CURRENTS ≤ 40 µA			
PAI_41	$I_{DFLLULP\_PL0}$	DFLLULP current (1 of 2)	Fout = 12 MHz, PL0 mode
PAI_42	I_{EVSYS}	EVSYS Active current	One channel operating, Synchronous mode
MODULES/PERIPHERALS ACTIVE CURRENTS ≤ 55 µA			
PAI_51	I_{CCL}	CCL Active current	All LUT Active, Filter Enabled
PAI_52	I_{AC_SPEED0}	AC Active current (1 of 4)	Voltage scaler disable, COMPCTRLn.SPEED = 0
MODULES/PERIPHERALS ACTIVE CURRENTS ≤ 70 µA			
PAI_71	$I_{XOSC\_2MHZ\_AMPGC\_ON}$	XOSC current (1 of 10)	F = 2 MHz, C_L=20 pF, XOSCCTRL GAIN=0, AMPGC = ON
PAI_72	$I_{XOSC\_2MHZ\_AMPGC\_OFF}$	XOSC current (2 of 10)	F = 2 MHz, C_L=20 pF, XOSCCTRL GAIN=0, AMPGC = OFF
MODULES/PERIPHERALS ACTIVE CURRENTS ≤ 80 µA			
PAI_81	$I_{XOSC\_4MHZ\_AMPGC\_ON}$	XOSC current (3 of 10)	F = 4 MHz, C_L = 20 pF, XOSCCTRL GAIN=0, AMPGC = ON
PAI_82	I_{OPAMP_MODE2}	OPAMP Active current (3 of 4)	OPAMPCTRLx.BIAS = 2
PAI_83	$I_{RC\_4\ MHZ}$	OSC16M Multi-RC current	F = 4 MHz
MODULES/PERIPHERALS ACTIVE CURRENTS ≤ 125 µA			
PAI_121	$I_{RC\_8\ MHZ}$	OSC16M Multi-RC current	F = 8 MHz
PAI_122	$I_{XOSC\_4MHZ\_AMPGC\_OFF}$	XOSC current (4 of 10)	F = 4 MHz, C_L = 20 pF, XOSCCTRL GAIN = 0, AMPGC = OFF
MODULES/PERIPHERALS ACTIVE CURRENTS ≤ 150 µA			
PAI_151	$I_{XOSC\_8MHZ\_AMPGC\_ON}$	XOSC current (5 of 10)	F = 8 MHz, C_L = 20 pF, XOSCCTRL GAIN = 0, AMPGC = ON

.....continued

DC CHARACTERISTICS		Standard Operating Conditions: VDD = AVDD = 1.62V to 3.63V (unless otherwise stated) Operating temperature -40°C ≤ T_A ≤ +85°C for Industrial	
Param. No.	**Symbol**	**Characteristics**	**Conditions**
PAI_152	I_{RC_12MHZ}	OSC16M Multi-RC current	F = 12 MHz
MODULES/PERIPHERALS ACTIVE CURRENTS ≤ 175 µA			
PAI_171	$I_{DFLLULP\_PL2}$	DFLLULP current (2 of 2)	Fout = 32 MHz, PL2 mode
PAI_172	I_{EIC}	EIC/NMI Active current	One EIC line operating, Filter Enabled
MODULES/PERIPHERALS ACTIVE CURRENTS ≤ 200 µA			
PAI_201	I_{RC_16MHZ}	OSC16M Multi-RC current	F = 16 MHz
PAI_202	I_{OPAMP_MODE3}	OPAMP Active current (4 of 4)	OPAMPCTRLx.BIAS = 3
MODULES/PERIPHERALS ACTIVE CURRENTS ≤ 250 µA			
PAI_251	$I_{XOSC\_8MHZ\_AMPGC\_OFF}$	XOSC current (6 of 10)	F = 8 MHz, C_L = 20 pF, XOSCCTRL GAIN=0, AMPGC = OFF
PAI_252	I_{AC_SPEED1}	AC Active current (2 of 4)	Voltage scaler disable, COMPCTRLn.SPEED = 1
MODULES/PERIPHERALS ACTIVE CURRENTS ≤ 300 µA			
PAI_301	$I_{XOSC\_16MHZ\_AMPGC\_ON}$	XOSC current (7 of 10)	F = 16 MHz, C_L = 20 pF, XOSCCTRL GAIN = 0, AMPGC = ON
PAI_302	$I_{SERCOMx}$	SERCOMx Active current	USART Mode, no communication
MODULES/PERIPHERALS ACTIVE CURRENTS ≤ 400 µA			
PAI_401	I_{DPLL_PL0}	DPLL current (1 of 2)	Fout = 48 MHz, PL0 mode
PAI_402	I_{TCx}	TCx Active current	Normal Frequency mode, Counter in 16 bits mode,
PAI_403	I_{DFLL48}	DFLL48M current	Fout = 48 MHz, Closed loop mode
MODULES/PERIPHERALS ACTIVE CURRENTS ≤ 500 µA			
PAI_501	I_{DAC} (3)	DAC Active current	VREF = AVDD, DAC DATA = 0x3FF
PAI_502	$I_{XOSC\_16MHZ\_AMPGC\_OFF}$	XOSC current (8 of 10)	F = 16 MHz, C_L = 20 pF, XOSCCTRL GAIN=0, AMPGC = OFF
PAI_503	I_{AC_SPEED2}	AC Active current (3 of 4)	Voltage scaler disable, COMPCTRLn.SPEED=2
MODULES/PERIPHERALS ACTIVE CURRENTS ≤ 700 µA			
PAI_701	$I_{XOSC\_32MHZ\_AMPGC\_ON}$	XOSC current (9 of 10)	F = 32 MHz, C_L = 20 pF, XOSCCTRL GAIN = 0, AMPGC = ON
PAI_702	I_{FREQM} (4)	FREQM Active current	Reference running at 48MHz and Measure clock running at 96 MHz,
MODULES/PERIPHERALS ACTIVE CURRENTS ≤ 750 µA			
PAI_751	I_{DPLL_PL2}	DPLL current (2 of 2)	Fout = 96 MHz, PL2 mode
MODULES/PERIPHERALS ACTIVE CURRENTS ≤ 1 mA			
PAI_1001	I_{AC_SPEED3}	AC Active current (4 of 4)	Voltage scaler disable, COMPCTRLn.SPEED = 3
MODULES/PERIPHERALS ACTIVE CURRENTS ≤ 1.5 mA			

.....continued

DC CHARACTERISTICS		Standard Operating Conditions: VDD = AVDD = 1.62V to 3.63V (unless otherwise stated) Operating temperature -40°C ≤ T_A ≤ +85°C for Industrial	
Param. No.	**Symbol**	**Characteristics**	**Conditions**
PAI_1501	$I_{XOSC\_32MHZ\_AMPGC\_OFF}$	XOSC current (10 of 10)	F = 32MHz, C_L = 20pF XOSCCTRL GAIN=0, AMPGC = OFF
PAI_1502	I_{TCCx}	TCCx Active current	Normal Frequency mode
MODULES/PERIPHERALS ACTIVE CURRENTS ≤ 2 mA			
PAI_2001	I_{ADC} (5)	ADC Active current	Free-Running, 1 MSPS
MODULES/PERIPHERALS ACTIVE CURRENTS ≤ 2.5 mA			
PAI_2501	I_{DMA}	DMA Active current	RAM-to-RAM transfer, one channel operating

Notes:

1. These values are for worst case guidance only unless otherwise specified, actual values depend on user peripheral configuration and operating frequency.
2. Conditions (Unless otherwise stated):
   - VDD = AVDD = 3.3V
   - LDO regulator used
   - Performance Level 2 (PL2) selected
   - Only mentioned peripheral modules is operating (i.e. rest of the peripherals are inactive)
   - MCLK all APB clock masked except MCLK and NVMCTRL and selected peripheral
   - All clock generation sources disabled unless otherwise specified. (i.e. XOSC32 = Off)
   - All clock sources disabled except XOSC32K running with external 32kHz crystal and FDPLL96M using XOSC32K as reference and running at 96MHz divided by 2 on GCLK0
   - GCLK clock generators 1 to 8 stopped (GENCTRL[8:1].GENEN = 0)
   - AHB/APB clocks not needed are masked: AHBMASK = 0x70, APB(A/B/C/D)MASK = 0x0
   - CPU and AHB clocks undivided
   - All I/O pins configured as input pins pulled down or tied to VSS
   - WDT, RTC, CFD Clock Fail Detect disabled
   - $\overline{\text{RESET}}$ = VDD
   - CPU is running on Flash with 3 Wait States
   - NVMCTRL cache enabled
   - BOD33 disabled
   - Measure is differential between active and inactive module in those conditions
3. Conditions:
   - Same Conditions as Note 2
   - GCLK1 running on FDPLL96M at 96 MHz is divided by 96
   - Measure is differential between active and inactive module in those conditions
4. Conditions:
   - Same Conditions as Note 2
   - GCLK1 running on FDPLL96M at 96 MHz is not divided
   - Measure is differential between active and inactive module in those conditions
5. Conditions:
   - Same Conditions as Note 2
   - GCLK1 running on FDPLL96M at 96 MHz is divided by 6
   - Measure is differential between active and inactive module in those conditions

49.11 I/O Pin Electrical Specifications

Table 49-14. I/O Pin Electrical Specifications (1)

DC CHARACTERISTICS			Standard Operating Conditions: VDD = AVDD = 1.62V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ T_A ≤ +85°C for Industrial				
Param. No.	**Symbol**	**Characteristics**	**Min.**	**Typ.**	**Max.**	**Units**	**Conditions**
DI_1	VIL	Input Low Voltage I/O Pins	VSS AVSS	—	0.2 VDD 0.2 AVDD	V	—
DI_3	VIH	Input High Voltage	0.7 VDD 0.7 AVDD	—	VDD AVDD	V	—
DI_5	VOL (5)	Output Voltage Low (Normal Pins) (Low Drive Strength, DRVSTR=0)	—	—	0.4	V	VDD/AVDD < 3.0V @ I_{OL}= 1.5 mA
DI_5A		Output Voltage Low (Normal Pins) (Low Drive Strength, DRVSTR=0)	—	—			VDD/AVDD ≥ 3.0V @ I_{OL}= 2 mA
DI_6		Output Voltage Low (Normal Pins) (High Drive Strength, DRVSTR=1)	—	—			VDD/AVDD < 3.0V @ I_{OL}= 2.5 mA
DI_6A		Output Voltage Low (Normal Pins) (High Drive Strength, DRVSTR=1)	—	—			VDD/AVDD ≥ 3.0V @ I_{OL}= 3 mA
DI_7		Output Voltage Low (High Sink Pins) (Low Drive Strength, DRVSTR=0)	—	—			VDD/AVDD < 3.0V @ I_{OL}= 2.5 mA
DI_7A		Output Voltage Low (High Sink Pins) (Low Drive Strength, DRVSTR=0)	—	—			VDD/AVDD ≥ 3.0V @ I_{OL}= 3.5 mA
DI_8		Output Voltage Low (High Sink Pins) (High Drive Strength, DRVSTR=1)	—	—			VDD/AVDD < 3.0V @ I_{OL}= 4 mA
DI_8A		Output Voltage Low (High Sink Pins) (High Drive Strength, DRVSTR=1)	—	—			VDD/AVDD ≥ 3.0V @ I_{OL}= 6 mA

.....continued

DC CHARACTERISTICS			Standard Operating Conditions: VDD = AVDD = 1.62V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ T_A ≤ +85°C for Industrial				
Param. No.	**Symbol**	**Characteristics**	**Min.**	**Typ.**	**Max.**	**Units**	**Conditions**
DI_9	V_{OH} (5)	Output Voltage High (Normal Pins) (Low Drive Strength, DRVSTR=0)	0.72 VDD 0.72 AVDD	—	—	V	VDD/AVDD < 3.0V @ I_{OH}= 1 mA
DI_9A		Output Voltage High (Normal Pins) (Low Drive Strength, DRVSTR=0)		—	—		VDD/AVDD ≥ 3.0V @ I_{OH}= 4 mA
DI_10		Output Voltage High (Normal Pins) (High Drive Strength, DRVSTR=1)		—	—		VDD/AVDD < 3.0V @ I_{OH}= 2 mA
DI_10A		Output Voltage High (Normal Pins) (High Drive Strength, DRVSTR=1)		—	—		VDD/AVDD ≥ 3.0V @ I_{OH}= 7 mA
DI_11		Output Voltage High (High Sink Pins) (Low Drive Strength, DRVSTR=0)		—	—		VDD/AVDD < 3.0V @ I_{OH}= 2 mA
DI_11A		Output Voltage High (High Sink Pins) (Low Drive Strength, DRVSTR=0)		—	—		VDD/AVDD ≥ 3.0V @ I_{OH}= 7.5 mA
DI_12		Output Voltage High (High Sink Pins) (High Drive Strength, DRVSTR=1)		—	—		VDD/AVDD < 3.0V @ I_{OH}= 4 mA
DI_12A		Output Voltage High (High Sink Pins) (High Drive Strength, DRVSTR=1)		—	—		VDD/AVDD ≥ 3.0V @ I_{OH}= 13 mA
DI_13	I_{IL}	Input pin leakage current	-1	—	1	µA	VSS ≤ VPIN ≤ VDD/AVDD(max) (VPIN = Voltage present on Pin)
DI_15	R_{PDWN}	Internal Pull-Down (DIR=OUT=0, PULLEN=1)	20	—	63	kΩ	—
DI_17	R_{PUP}	Internal Pull-Up (DIR=0, OUT=PULLEN=1)	20	—	63	kΩ	
DI_19	I_{ICL} (1,3,4)	Input Low Injection Current	-1	—	—	mA	—
DI_21	I_{ICH} (2,3,4)	Input High Injection Current	—	—	1	mA	—
DI_23	∑IICT	Total Input Injection Current (sum of all I/O and control pins) Absolute value of | ∑IICT |	—	—	45	mA	Absolute instantaneous sum of all ± input injection currents from all I/O pins. (| IICL | + | IICH |) ≤ ∑IICT

.....continued

DC CHARACTERISTICS			Standard Operating Conditions: VDD = AVDD = 1.62V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ T_A ≤ +85°C for Industrial				
Param. No.	Symbol	Characteristics	Min.	Typ.	Max.	Units	Conditions
DI_25	T_{RISE} (5)	I/O pin Rise Time (Normal Pins) (Low Drive Strength, DRVSTR=0)	—	—	13	ns	VDD/AVDD(min), C_{LOAD}=30pF(MAX)
DI_25A		I/O pin Rise Time (High Sink Pins) (Low Drive Strength, DRVSTR=0)	—	—	6		
DI_25B		I/O pin Rise Time (Normal Pins) (High Drive Strength, DRVSTR=1)	—	—	6		
DI_25C		I/O pin Rise Time (High Sink Pins) (High Drive Strength, DRVSTR=1)	—	—	4.5		
DI_27	T_{FALL} (5)	I/O pin Fall Time (Normal Pins) (Low Drive Strength, DRVSTR=0)	—	—	12	ns	
DI_27A		I/O pin Fall Time (High Sink Pins) (Low Drive Strength, DRVSTR=0)	—	—	7		
DI_27B		I/O pin Fall Time (Normal Pins) (High Drive Strength, DRVSTR=1)	—	—	7		
DI_27C		I/O pin Fall Time (High Sink Pins) (High Drive Strength, DRVSTR=1)	—	—	4.5		

Notes:

1. VIL source < (VSS - 0.3). Characterized but not tested.
2. VIH source > (VDD + 0.3).
3. If the sum of all injection currents are > | ∑IICT | it can affect the ADC results by approximately 4 to 6 counts (i.e., VIH Source > (VDD + 0.3) or VIL source < (VSS - 0.3)).
4. Any number and/or 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, (i.e. RS), between input source voltage and device pin. The resistor value is calculated according to:
   - For negative Input voltages less than (VSS - 0.3): RS ≥ absolute value of | ((VIL source - (VSS - 0.3)) / IICL) |
   - For positive input voltages greater than (VDD + 0.3): RS ≥ ((VIH source - (VDD + 0.3))/ IICH)
   - For Vpin voltages >VDD + 0.3 and <VSS - 0.3 then RS = the larger of the values calculated above
5. High Sink I/O pins are: PA08, PA09, PA12, PA13, PA16, PA17, PA22, PA23, PA31, PB12, PB13, PB16, PB17, PB30, PB31.

49.12 Internal Voltage Reference Electrical Specifications

Table 49-15. Internal Voltage Reference Electrical Specifications (1)

DC CHARACTERISTICS			Standard Operating Conditions: VDD = AVDD = 1.62V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ T_A ≤ +85°C for Industrial				
Param. No.	**Symbol**	**Characteristics**	**Min.**	**Typ.**	**Max.**	**Units**	**Conditions**
VR_1	I_{REF}	Internal Voltage Reference ADC.REFCTRL.REFSEL = INTREF	2.344	2.4	2.455	V	AVDD=3.3V, T_A=25°C SUPC VREF.SEL = 0x6
VR_3		DAC.CTRLB.REFSEL = INTREF	2.433	2.5	2.565		AVDD=3.3V, T_A=25°C SUPC VREF.SEL = 0x7
VR_9	AC_{IREF}	Internal Voltage Reference Comparator AC.COMPCTRLn.MUXNEG = BANDGAP	1.086	1.1	1.122	V	AVDD=3.3V, T_A=25°C
VR_25	T_{DRIFT}	Internal Voltage Reference Temperature Drift	-0.012	—	0.030	%/°C	AVDD=3.3V
VR_27	V_{DRIFT}	Internal Voltage Reference Voltage Drift	-0.856	—	0.856	%/V	T_A=25°C

Note:

1. Typical values at 25°C only.

49.13 Maximum Clock Frequencies Electrical Specifications

Table 49-16. Maximum Clock Frequencies Electrical Specifications

AC CHARACTERISTICS		Standard Operating Conditions: VDD = AVDD = 1.62V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ T_A ≤ +85°C for Industrial			
Param. No.	Symbol	Characteristics	Max.		Units
			PL0	PL2	
FCLK_1	f_{CPU}	CPU clock frequency	12	48	MHz
FCLK_3	f_{AHB}	AHB clock frequency	12	48	MHz
FCLK_5	f_{APBA}, f_{APBB}, f_{APBC}	APBA, APBB, APBC clock frequency	12	48	MHz
FCLK_6	f_{GCLK_GENn}	Output frequency (UNDIVIDED)	25	96	MHz
		Output frequency (DIVIDED)	12	48	
FCLK_7	$f_{GCLK\_FDPLL96M}$	FDPLL96M reference clock frequency	2	2	MHz
FCLK_9	$f_{GCLK\_FDPLL96M\_32K}$	FDPLL96M 32K reference slow clock frequency	33	33	kHz
FCLK_11	$f_{GCLK\_DFLL48M}$	DFLL48M reference clock frequency	33	33	kHz
FCLK_12	$f_{GCLK\_DFLLULP}$	DFLLULP reference clock frequency	1	4	MHz
FCLK_13	f_{GCLK_EIC}	EIC input clock frequency	12	48	MHz
FCLK_15	$f_{GCLK\_FREQM\_MSR}$	FREQM Measure clock frequency	25	100	MHz
FCLK_17	$f_{GCLK\_FREQM\_REF}$	FREQM Reference clock frequency	25	100	MHz
FCLK_19	$f_{GCLK\_EVSYS\_CHANNELx}$	EVSYS channel x input clock frequency	12	48	MHz
FCLK_21	$f_{GCLK\_SERCOMx\_SLOW}$	Common SERCOM slow input clock frequency	1	5	MHz
FCLK_23	$f_{GCLK\_SERCOMx\_CORE}$	SERCOMx input clock frequency	6	24	MHz
FCLK_27	f_{GCLK_USB}	USB input clock frequency	N/A	60	MHz
FCLK_29	f_{GCLK_I2S}	I^2S input clock frequency	12	12	MHz
FCLK_35	f_{GCLK_TCCx}	TCCx input clock frequency	25	100	MHz
FCLK_37	f_{GCLK_TCx}	TCx input clock frequency	12	48	MHz
FCLK_43	f_{GCLK_CCL}	CCL input clock frequency	12	48	MHz
FCLK_45	$f_{GCLK\_GCLKINx}$	External GCLKx input clock frequency	12	48	MHz
FCLK_49	f_{GCLK_AC}	AC input clock frequency	12	48	MHz
FCLK_51	f_{GCLK_ADC}	ADC input clock frequency	12	48	MHz
FCLK_53	f_{GCLK_DAC}	DAC input clock frequency	12	12	MHz
FCLK_55	f_{GCLK_PTC}	PTC Input Clock Frequency	12	48	MHz

49.14 XOSC Electrical Specifications

Table 49-17. External XTAL and Clock Electrical Specifications

AC CHARACTERISTICS			Standard Operating Conditions: VDD =AVDD = 1.62V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ T_A ≤ +85°C for Industrial				
Param. No.	**Symbol**	**Characteristics**	**Min.**	**Typ.**	**Max.**	**Units**	**Conditions**
XOSC_1	F_{OSC_XOSC}	X_{OSC} Crystal Frequency	0.4	—	32	MHz	XOSCCTRL.XTALEN=1 XIN, XOUT Primary Osc
XOSC_1A	T_{OSC}	T_{OSC} = 1/ F_{OSC_XOSC}	31.25	—	2500	ns	—
XOSC_2	X_{OSC_ST} (1)	X_{OSC} Crystal Start-up Time	—	15600	Note (2)	T_{OSC}	C_{LOAD} = 20pF
XOSC_3	C_{XIN}	X_{OSC} X_{IN} parasitic pin capacitance	—	7	—	pF	—
XOSC_5	C_{XOUT}	X_{OSC} X_{OUT} parasitic pin capacitance	—	4.5	—	pF	—
XOSC_11	C_{LOAD} (3)	X_{OSC} Crystal F_{OSC} = 0.4 MHz	—	—	100	pF	XOSCCTRL.GAIN = 0x0
XOSC_13		XOSC Crystal FOSC = 2 MHz	—	—	20		XOSCCTRL.GAIN = 0x0
XOSC_15		X_{OSC} Crystal F_{OSC} = 4 MHz	—	—	20		XOSCCTRL.GAIN = 0x1
XOSC_17		X_{OSC} Crystal F_{OSC} = 8 MHz	—	—	20		XOSCCTRL.GAIN = 0x2
XOSC_19		X_{OSC} Crystal F_{OSC} = 16 MHz	—	—	20		XOSCCTRL.GAIN = 0x3
XOSC_20		X_{OSC} Crystal F_{OSC} = 32 MHz	—	—	20		XOSCCTRL.GAIN = 0x4
XOSC_21	ESR	X_{OSC} Crystal F_{OSC} = 0.4 MHz	—	—	5600	Ω	XOSCCTRL.GAIN = 0x0 C_{LOAD}= 20pF
XOSC_23		X_{OSC} Crystal F_{OSC} = 2 MHz	—	—	330		XOSCCTRL.GAIN = 0x0 C_{LOAD}= 20pF
XOSC_25		X_{OSC} Crystal F_{OSC} = 4 MHz	—	—	240		XOSCCTRL.GAIN = 0x1 C_{LOAD}= 20pF
XOSC_27		X_{OSC} Crystal F_{OSC} = 8 MHz	—	—	105		XOSCCTRL.GAIN = 0x2 C_{LOAD}= 20pF
XOSC_29		X_{OSC} Crystal F_{OSC} = 16 MHz	—	—	60		XOSCCTRL.GAIN = 0x3 C_{LOAD}= 20pF
XOSC_30		X_{OSC} Crystal F_{OSC} = 32 MHz	—	—	55		XOSCCTRL.GAIN = 0x4 C_{LOAD}= 20pF
XOSC_35	F_{OSC_XCLK}	Ext Clock Oscillator Input Freq (X_{IN} pin)	0	—	24	MHz	XOSCCTRL.XTALEN=0
XOSC_37	XCLK_DC	Ext Clock Oscillator (X_{IN}) Duty Cycle	40	—	60	%	XOSCCTRL.XTALEN=0

.....continued

AC CHARACTERISTICS			Standard Operating Conditions: VDD =AVDD = 1.62V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ T_A ≤ +85°C for Industrial				
Param. No.	Symbol	Characteristics	Min.	Typ.	Max.	Units	Conditions
XOSC_39	XCLK_FST	Primary X_{IN} Clock Fail Safe Time-out Period	—	4 / (RC16M_1C / 2^(CFDPRESC.CFDPRESC))	—	µs	RC16M_1C: Refer to OSC16M Electrical Specifications

Notes:

1. This is for guidance only. A major component of crystal start-up time is based on the second party crystal MFG parasitics that are outside the scope of this specification. If this is a major concern the customer would need to characterize this based on their design choices.
2. Maximum start up time user selectable in XOSCCTRL.STARTUP.
3. CRYSTAL LOAD CAPACITOR CALCULATION GIVEN:
   - Standard PCB trace capacitance = 1.5 pF per 12.5 mm(0.5 inches) (i.e. PCB STD TRACE W=0.175 mm, H=36 µm, T=113 µm)
   - Xtal PCB capacitance typical therefore ~= 2.5pF for a tight PCB xtal layout
   - For C_{XIN} and C_{XOUT} within 4pF of each other, Assume CXTAL_EFF = ((C_{XIN}+C_{XOUT}) / 2)
     Note: Averaging C_{XIN} and C_{XOUT} will effect final calculated C_{LOAD} value by less than 0.25pF.

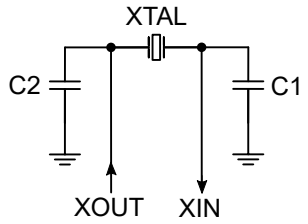
EQUATION 1:

MFG C_{LOAD} Spec = {([C_{XIN} + C1] * [C_{XOUT} + C2]) / [C_{XIN} + C1 + C2 + C_{XOUT}] } + estimated oscillator PCB stray capacitance

- Assuming C1 = C2 and C_{XIN} ~= C_{XOUT}, the formula can be further simplified and restated to solve for C1 and C2 by:

EQUATION 2: (Simplified version of equation 1)

C1 = C2 = ((2 * MFG C_{LOAD} spec) - C_{XTAL_EFF} - (2 * PCB capacitance))



EXAMPLE ONLY:

- XTAL Mfg C_{LOAD} Data Sheet Spec = 12pF
- PCB XTAL trace Capacitance = 2.5pF
- C_{XIN} pin = 6.5pF, C_{XOUT} pin = 4.5pF. Therefore C_{XTAL_EFF} = ((C_{XIN}+C_{XOUT}) / 2)

C_{XTAL_EFF} = ((6.5 + 4.5)/2) = 5.5pF

C1 = C2 = ((2 * MFG C_{LOAD} spec) - C_{XTAL_EFF} - (2 * PCB capacitance))

C1 = C2 = (24 - 5.5 - (2 * 2.5))

C1 = C2 = 13.5pF (Always rounded down)

C1 = C2 = 13pF (i.e. for hypothetical example crystal external load capacitors)

User C1=C2=13pF ≤ C_{LOAD}(max) spec

49.15 XOSC32K Electrical Specifications

Table 49-18. External 32.768 kHz XTAL and Clock Electrical Specifications

AC CHARACTERISTICS			Standard Operating Conditions: VDD = AVDD = 1.62V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ T_A ≤ +85°C for Industrial				
Param. No.	**Symbol**	**Characteristics**	**Min.**	**Typ.**	**Max.**	**Units**	**Conditions**
XOSC32_1	F_{OSC_XOSC32}	X_{OSC32} Oscillator Crystal Frequency	—	32.768	—	kHz	X_{IN32}, X_{OUT32} Secondary Osc
XOSC32_3	C_{XIN32}	X_{OSC32} X_{IN32} parasitic pin capacitance	—	4.2	—	pF	—
XOSC32_5	C_{XOUT32}	X_{OSC32} X_{OUT32} parasitic pin capacitance	—	4.2	—	pF	—
XOSC32_11	C_{LOAD_X32} (3)	32.768kz Crystal Load Capacitance	—	—	9	pF	XOSC32K.XTALEN = 1 XOSC32K.ENABLE = 1
XOSC32_13	ESR_X32	32.768kz Crystal ESR	—	—	70	kΩ	XOSC32K.XTALEN = 1 XOSC32K.ENABLE = 1
XOSC32_15	T_{OSC32}	T_{OSC32} = 1/ F_{OSC_XOSC32}	—	30.518	—	µs	See parameter XOSC32_1 for F_{OSC_XOSC32} value
XOSC32_17	XOSC32_ST (1)	X_{OSC32} Crystal Start-up Time	—	10000	Note (2)	T_{OSC32}	—
XOSC32_19	F_{OSC_XCLK32}	Ext Clock Oscillator Input Freq (X_{IN32} pin)	—	32.768	—	kHz	±100ppm max XOSC32K.XTALEN=0 XOSC32K.ENABLE = 1
XOSC32_21	XCLK32_DC	Ext Clock Oscillator Duty Cycle	40	—	60	%	XOSC32K.XTALEN=0 XOSC32K.ENABLE = 1
XOSC32_23	XCLK32_FST	X_{IN32} Clock Fail Safe Time-out Period	—	4 / (LP32K_1 / 2^(CFDCTRL.CFDPRESC))	—	ms	LP32K_1: Refer to OSCULP32K Electrical Specifications

.....continued

AC CHARACTERISTICS			Standard Operating Conditions: VDD = AVDD = 1.62V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ T_A ≤ +85°C for Industrial				
Param. No.	Symbol	Characteristics	Min.	Typ.	Max.	Units	Conditions

Notes:

1. This is for guidance only. A major component of crystal start-up time is based on the second party crystal MFG parasitics that are outside the scope of this specification. If this is a major concern the customer would need to characterize this based on their design choices.
2. Maximum start up time user selectable in XOSC32K.STARTUP.
3. CRYSTAL LOAD CAPACITOR CALCULATION GIVEN:
   - Standard PCB trace capacitance = 1.5 pF per 12.5 mm(0.5 inches) (i.e. PCB STD TRACE W=0.175 mm, H=36 µm, T=113 µm)
   - Xtal PCB capacitance typical therefore ~= 2.5pF for a tight PCB Xtal layout
   - For C_{XIN32} and C_{XOUT32} within 4pF of each other, Assume CXTAL_EFF = ((CXIN32+CXOUT32) / 2)
     Note: Averaging C_{XIN32} and C_{XOUT32} will effect final calculated C_{LOAD} value by less than 0.25pF.

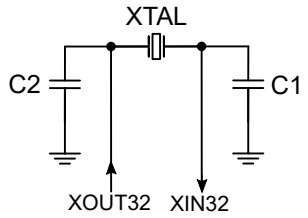
EQUATION 1:

MFG C_{LOAD} Spec = {([C_{XIN32} + C1] * [C_{XOUT32} + C2]) / [C_{XIN32} + C1 + C2 + C_{XOUT32}] } + estimated oscillator PCB stray capacitance

- Assuming C1 = C2 and C_{XIN32} ~= C_{XOUT32}, the formula can be further simplified and restated to solve for C1 and C2 by:

EQUATION 2: (Simplified Equation 1)

C1 = C2 = ((2 * MFG C_{LOAD} spec) - C_{XTAL_EFF} - (2 * PCB capacitance))



EXAMPLE ONLY:

- XTAL Mfg C_{LOAD} Data Sheet Spec = 12pF
- PCB XTAL trace Capacitance = 2.5pF
- C_{XIN32} pin = 6.5pF, C_{XOUT32} pin = 4.5pF. Therefore C_{XTAL_EFF} = ((C_{XIN32}+C_{XOUT32}) / 2)

C_{XTAL_EFF} = ((6.5 + 4.5)/2) = 5.5pF

C1 = C2 = ((2 * MFG C_{LOAD} spec) - C_{XTAL_EFF} - (2 * PCB capacitance))

C1 = C2 = (24 - 5.5 - (2 * 2.5))

C1 = C2 = (24 - 5.5 - 5)

C1 = C2 = 13.5pF (Always rounded down)

C1 = C2 = 13pF (i.e. for hypothetical example crystal external load capacitors)

User C1=C2=13pF ≤ C_{LOAD}_X32(max) spec

49.16 OSCULP32K Electrical Specifications

Table 49-19. Low Power Internal 32.768 kHz RC Oscillator Electrical Specifications

AC CHARACTERISTICS			Standard Operating Conditions: VDD = AVDD = 1.62V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ T_A ≤ +85°C for Industrial				
Param. No.	Symbol	Characteristics	Min.	Typ.	Max.	Units	Conditions
LP32K_1	FOSC_LPRC32K	Output Frequency	—	32.768	—	kHz	T_A = 25°C , AVDD = 3.3V
LP32K_3	LPRC32K_ACC	Accuracy	-23	—	25	%	AVDD = 3.3V
LP32K_9	RC32K_Duty	Oscillator Duty Cycle	—	50	—	%	T_A = 25°C , AVDD = 3.3V

49.17 OSC16M Electrical Specifications

Table 49-20. 4/8/12/16 MHz Internal Multi-RC Oscillator Electrical Specifications

AC CHARACTERISTICS			Standard Operating Conditions: VDD = AVDD = 1.62V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ T_A ≤ +85°C for Industrial				
Param. No.	**Symbol**	**Characteristics**	**Min.**	**Typ.**	**Max.**	**Units**	**Conditions**
RC16M_1	FOSC_RC16M	Output Frequency	—	4	—	MHz	T_A = 25°C , AVDD = 3.3V OSC16MCTRL.FSEL = 0x00
RC16M_1A				8			T_A = 25°C , AVDD = 3.3V OSC16MCTRL.FSEL = 0x01
RC16M_1B				12			T_A = 25°C , AVDD = 3.3V OSC16MCTRL.FSEL = 0x10
RC16M_1C				16			T_A = 25°C , AVDD = 3.3V OSC16MCTRL.FSEL = 0x11
RC16M_3	RC16M_ACC	Accuracy	-7.5	—	7.5	%	AVDD = 3.3V FOSC_RM16M = 4 MHz
RC16M_3A			-5.6		5.6		AVDD = 3.3V FOSC_RM16M = 8 MHz
RC16M_3B			-5.4		5.4		AVDD = 3.3V FOSC_RM16M = 12 MHz
RC16M_3C			-5.6		5.6		AVDD = 3.3V FOSC_RM16M = 16 MHz
RC16M_5	RC16M_START	Start-up Time	—	1.16	—	µs	AVDD = 3.3V FOSC_RM16M = 4 MHz
RC16M_5A			—	1.29	—		AVDD = 3.3V FOSC_RM16M = 8 MHz
RC16M_5B			—	1.34	—		AVDD = 3.3V FOSC_RM16M = 12 MHz
RC16M_5C			—	1.39	—		AVDD = 3.3V FOSC_RM16M = 16 MHz
RC16M_7	RC16M_Duty	OSC16M OSC Duty Cycle	—	50	—	%	T_A = 25°C , AVDD = 3.3V

49.18 DFLL48M Electrical Specifications

Table 49-21. DFLL48M (Digital Frequency Locked Loop) Electrical Specifications

AC CHARACTERISTICS			Standard Operating Conditions: VDD = AVDD = 1.62V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ T_A ≤ +85°C for Industrial				
Param. No.	**Symbol**	**Characteristics**	**Min.**	**Typ.**	**Max.**	**Units**	**Conditions**
DFLL48M (Open Loop) (1,2)							
DFLL_1	DFLL48M_OL_F_{OUT}	DFLL48M Open Loop Clock Frequency	45.6	46.4	47	MHz	LDO Mode, Performance Level 0 Mode
DFLL_1A			47.4	47.9	48.5		LDO Mode, Performance Level 2 Mode
DFLL_1B			45.1	46.4	47.3		BUCK Mode, Performance Level 0 Mode
DFLL_1C			47.3	47.9	48.6		BUCK Mode, Performance Level 2 Mode
DFLL_9	DFLL48M_OL_SRT	Start-Up (Ready bit valid)	—	—	11	µs	Performance Level 0 Mode
DFLL_9A			—	—	10		Performance Level 2 Mode
DFLL48M (Closed Loop) (3,4)							
DFLL_11	DFLL48M_CL_F_{IN}	DFLL48M Closed loop Input Frequency Range	0.732	32.768	33	kHz	-
DFLL_13	DFLL48M_CL_F_{OUT}	DFLL48M Closed Loop Clock Frequency	47.7	48	48.3	MHz	LDO Mode, Performance Level 0 Mode, XOSC32 32.768 kHz PPM≤100
DFLL_13A			47.8	48	48.2		LDO Mode, Performance Level 2 Mode, XOSC32 32.768 kHz PPM≤100
DFLL_13B			47.2	48	48.9		BUCK Mode, Performance Level 0 Mode, XOSC32 32.768 kHz PPM≤100
DFLL_13C			47.7	48	48.3		BUCK Mode, Performance Level 2 Mode, XOSC32 32.768 kHz PPM≤100

.....continued

AC CHARACTERISTICS			Standard Operating Conditions: VDD = AVDD = 1.62V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ T_A ≤ +85°C for Industrial				
Param. No.	**Symbol**	**Characteristics**	**Min.**	**Typ.**	**Max.**	**Units**	**Conditions**
DFLL_15	DFLL48M_CL_Jitter	DFLL48M Period Jitter Pk-to-Pk	—	—	2	%	LDO Mode, Performance Level 0 Mode, $f_{EXTERNAL}$=32.768 kHz @ ≤100ppm
DFLL_15A			—	—	1		LDO Mode, Performance Level 2 Mode, $f_{EXTERNAL}$=32.768 kHz @ ≤100ppm
DFLL_15B			—	—	13		LDO Mode, Performance Level 0 Mode, $f_{EXTERNAL}$=32.768 kHz @ ≤100ppm
DFLL_15C			—	—	4		LDO Mode, Performance Level 2 Mode, $f_{EXTERNAL}$=32.768 kHz @ ≤100ppm
DFLL_21	DFLL48M_CL_SRT (5)	DFLL48M Closed Loop Mode Lock Time	—	—	2074	µs	Performance Level 0 Mode, $f_{EXTERNAL}$=32.768 kHz @ ≤100ppm
DFLL_21A			—	—	853		Performance Level 2 Mode, $f_{EXTERNAL}$=32.768 kHz @ ≤100ppm

Notes:

1. In Open Loop mode, the DFLL48M clock will be determined by the values written to the DFLL Coarse and the DFLL Fine bit fields (DFLLVAL.COARSE and DFLLVAL.FINE) in the DFLL Value register. Using the "DFLL48M COARSE" value from the NVM Software Calibration Area in DFLLVAL.COARSE helps to output a frequency close to 48MHz.
2. Not recommended for functional USB operation, SOF sync start-up only.
3. In Closed loop mode the DFLL48M can use a variety of clock sources. The DFLL48M can be trimmed using register DFLLMUL.
4. To ensure that the device stays within the maximum allowed clock frequency, any reference clock for DFLL48M in close loop must be within a 2% error accuracy.
5. DFLLMUL = 1464, DFLLCTRL.BPLCKC = 1: only fine value change, coarse value locked to the reset value.
   DFLLCTRL.QLDIS = 0: quick lock enable, DFLLCTRL.CCDIS = 1: Enabling chill cycles might double the lock time.

   DFLLMUL.FSTEP = 10 : Max fine step size, divided or dividing into two parts, search. 10 is a optimum value.

49.19 DFLLULP Electrical Specifications

Table 49-22. DFLLULP (Digital Frequency Locked Loop) Electrical Specifications

AC CHARACTERISTICS			Standard Operating Conditions: VDD = AVDD = 1.62V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ T_A ≤ +85°C for Industrial				
Param. No.	**Symbol**	**Characteristics**	**Min.**	**Typ.**	**Max.**	**Units**	**Conditions**
DFLLULP (Closed Loop Only)							
DFLL_31	DFLLULP_CL_F_{IN}	DFLLULP Closed loop Input Frequency Range	32	32.768	33	kHz	-
DFLL_33	DFLLULP_CL_F_{OUT}	DFLLULP Closed Loop Clock Frequency	7.7	8	8.3	MHz	LDO Mode, Performance Level 0 Mode, XOSC32 32.768 kHz PPM≤100
DFLL_33A			30.9	32	33.1		LDO Mode, Performance Level 2 Mode, XOSC32 32.768 kHz PPM≤100
DFLL_33B			7.7	8	8.3		BUCK Mode, Performance Level 0 Mode, XOSC32 32.768 kHz PPM≤100
DFLL_33C			30.8	32	33.2		BUCK Mode, Performance Level 2 Mode, XOSC32 32.768 kHz PPM≤100
DFLL_35	DFLLULP_CL_Jitter	DFLLULP Period Jitter Pk-to-Pk	—	—	5.5	%	LDO Mode, Performance Level 0 Mode, $f_{EXTERNAL}$=32.768 kHz @ ≤100ppm
DFLL_35A			—	—	5.7		LDO Mode, Performance Level 2 Mode, $f_{EXTERNAL}$=32.768 kHz @ ≤100ppm
DFLL_35B			—	—	7.7		LDO Mode, Performance Level 0 Mode, $f_{EXTERNAL}$=32.768 kHz @ ≤100ppm
DFLL_35C			—	—	6.8		LDO Mode, Performance Level 2 Mode, $f_{EXTERNAL}$=32.768 kHz @ ≤100ppm
DFLL_37	DFLLULP_CL_SRT	DFLLULP Closed Loop Mode Lock Time	—	—	362	µs	Closed Loop mode, $f_{EXTERNAL}$=32.768 kHz @ ≤100ppm Binary Search mode enabled

49.20 FDPLL96M Electrical Specifications

Table 49-23. FDPLL96M (Frequency Digital Phase Locked Loop) Electrical Specifications

AC CHARACTERISTICS			Standard Operating Conditions: VDD = AVDD = 1.62V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ T_A ≤ +85°C for Industrial				
Param. No.	**Symbol**	**Characteristics**	**Min.**	**Typ.**	**Max.**	**Units**	**Conditions**
FDPLL96M (Fractional Digital Phase Locked Loop)							
FDPLL_1	FDPLL_F_{IN}	FDPLL96M Input Frequency Range	32	—	2000	kHz	XOSC32 32.768 kHz PPM≤100.
FDPLL_3	FDPLL_F_{OUT}	FDPLL96M Output Clock Frequency	32	—	48	MHz	XOSC32 32.768 kHz PPM≤100. Performance Level 0
FDPLL_3A			32	—	96		XOSC32 32.768 kHz PPM≤100. Performance Level 2
FDPLL_5	FDPLL_Jitter (1)	FDPLL96M Period Jitter Pk-to-Pk	—	—	3.8	%	LDO Mode, $f_{EXTERNAL}$=32.768 kHz @ ≤100ppm, f_{OUT}= 32MHz Performance Level 0
FDPLL_5A			—	—	4		LDO Mode, $f_{EXTERNAL}$=32.768 kHz @ ≤100ppm, f_{OUT}= 32MHz Performance Level 2
FDPLL_5B			—	—	2.7		LDO Mode, $f_{EXTERNAL}$=32.768 kHz @ ≤100ppm, f_{OUT}=48MHz Performance Level 0
FDPLL_5C			—	—	2.6		LDO Mode, $f_{EXTERNAL}$=32.768 kHz @ ≤100ppm, f_{OUT}=48MHz Performance Level 2
FDPLL_5D			—	—	2		LDO Mode, $f_{EXTERNAL}$=32.768 kHz @ ≤100ppm, f_{OUT}=96MHz Performance Level 2

.....continued

AC CHARACTERISTICS			Standard Operating Conditions: VDD = AVDD = 1.62V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ T_A ≤ +85°C for Industrial				
Param. No.	Symbol	Characteristics	Min.	Typ.	Max.	Units	Conditions
FDPLL_5E	FDPLL_Jitter (1)	FDPLL96M Period Jitter Pk-to-Pk	—	—	7.5	%	LDO Mode, $f_{EXTERNAL}$=2MHz @ ≤100ppm, f_{OUT}= 32MHz Performance Level 0
FDPLL_5F			—	—	6.5		LDO Mode, $f_{EXTERNAL}$=2MHz @ ≤100ppm, f_{OUT}= 32MHz Performance Level 2
FDPLL_5G			—	—	5.7		LDO Mode, $f_{EXTERNAL}$=2MHz @ ≤100ppm, f_{OUT}=48MHz Performance Level 0
FDPLL_5H			—	—	2.9		LDO Mode, $f_{EXTERNAL}$=2MHz @ ≤100ppm, f_{OUT}=48MHz Performance Level 2
FDPLL_5I			—	—	2.7		LDO Mode, $f_{EXTERNAL}$=2MHz @ ≤100ppm, f_{OUT}=96MHz Performance Level 2

.....continued

AC CHARACTERISTICS			Standard Operating Conditions: VDD = AVDD = 1.62V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ T_A ≤ +85°C for Industrial				
Param. No.	Symbol	Characteristics	Min.	Typ.	Max.	Units	Conditions
FDPLL_5J	FDPLL_Jitter (1)	FDPLL96M Period Jitter Pk-to-Pk	—	—	6.7	%	BUCK Mode, $f_{EXTERNAL}$=32.768 kHz @ ≤100ppm, f_{OUT}= 32MHz Performance Level 0
FDPLL_5K			—	—	4.8		BUCK Mode, $f_{EXTERNAL}$=32.768 kHz @ ≤100ppm, f_{OUT}= 32MHz Performance Level 2
FDPLL_5L			—	—	9.1		BUCK Mode, $f_{EXTERNAL}$=32.768 kHz @ ≤100ppm, f_{OUT}=48MHz Performance Level 0
FDPLL_5M			—	—	3.4		BUCK Mode, $f_{EXTERNAL}$=32.768 kHz @ ≤100ppm, f_{OUT}=48MHz Performance Level 2
FDPLL_5N			—	—	5.3		BUCK Mode, $f_{EXTERNAL}$=32.768 kHz @ ≤100ppm, f_{OUT}=96MHz Performance Level 2

.....continued

AC CHARACTERISTICS			Standard Operating Conditions: VDD = AVDD = 1.62V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ T_A ≤ +85°C for Industrial				
Param. No.	**Symbol**	**Characteristics**	**Min.**	**Typ.**	**Max.**	**Units**	**Conditions**
FDPLL_5O	FDPLL_Jitter (1)	FDPLL96M Period Jitter Pk-to-Pk	—	—	10.5	%	BUCK Mode, $f_{EXTERNAL}$=2MHz @ ≤100ppm, f_{OUT}= 32MHz Performance Level 0
FDPLL_5P			—	—	6.7		BUCK Mode, $f_{EXTERNAL}$=2MHz @ ≤100ppm, f_{OUT}= 32MHz Performance Level 2
FDPLL_5Q			—	—	10.0		BUCK Mode, $f_{EXTERNAL}$=2MHz @ ≤100ppm, f_{OUT}=48MHz Performance Level 0
FDPLL_5R			—	—	3.5		BUCK Mode, $f_{EXTERNAL}$=2MHz @ ≤100ppm, f_{OUT}=48MHz Performance Level 2
FDPLL_5S			—	—	5.3		BUCK Mode, $f_{EXTERNAL}$=2MHz @ ≤100ppm, f_{OUT}=96MHz Performance Level 2
FDPLL_11	FDPLL_SRT (1)	FDPLL96M Start-Up / Lock Time	—	1.2	—	ms	XOSC32 32.768 kHz PPM≤100.
FDPLL_11A			—	23	—		XOSC 2MHz PPM≤100.

Note:

1. REFCLK for FDPLL96M is XOSC or XOSC32K.

49.21 DAC Module Electrical Specifications

Table 49-24. DAC Module Electrical Specifications

AC CHARACTERISTICS				Standard Operating Conditions: VDD = AVDD = 1.62V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ T_A ≤ +85°C for Industrial				
Param. No.	**Symbol**	**Characteristics**		**Min.**	**Typ.**	**Max.**	**Units**	**Conditions**
DAC_1	D_{RES}	DAC Resolution		—	—	12	Bits	—
DAC_3	D_{CLK}	Internal DAC Clock Frequency (f_{GCLK_DAC})		—	—	FCLK_53	MHz	FCLK_53: Refer to Maximum Clock Frequencies Electrical Specifications
DAC_5	D_{SAMP}	DAC Sampling Rate	Low Power (DACCTRLx.CCTRL=0x0)	0.05	—	0.1	MSPS	+/-4 LSB of final value for step size≤100 LSB @ C_{LOAD} & R_{LOAD} w/AVDD=3.3V
			High Power (DACCTRLx.CCTRL=0x2)	0.1	—	1		
DAC_7	V_{OUT}	Output Voltage Linear Range		AVSS+0.15	—	AVDD-0.15	V	Ext Pin (Buffered) VREF=AVDD @ C_{LOAD} & R_{LOAD}
				AVSS+0.15	—	VREF		Ext Pin (Buffered) @ C_{LOAD} & R_{LOAD} (VREF < (AVDD-150mV))
				AVSS	—	VREF		Internal (no buffer)
DAC_9	V_{REF} (6)	DAC Reference Input Option	REFSEL = VREFAB CTRLB.REFSEL = 0x2	2.4V	—	AVDD-0.15	V	External Reference VREFAB (buffered) VREF bypass Cap = 10 nF
			REFSEL = VREFAU CTRLB.REFSEL = 0x0		—	AVDD		External Reference VREFAU (unbuffered) VREF bypass Cap = 10 nF
			REFSEL = INTREF CTRLB.REFSEL = 0x3		VR_1 VR_3	AVDD-0.15		Internal Reference VR_1, VR_3: Refer to Internal Voltage Reference Electrical Specifications
			REFSEL = AVDD CTRLB.REFSEL = 0x1	AVDD				—
DAC_11	C_{LOAD}	DAC Out max load to meet V_{OUT} & T_{SET}		—	—	50	pF	—
DAC_13	R_{LOAD}	DAC Out max load to meet V_{OUT} & T_{SET}		5	—	—	kΩ	—
DAC_15	T_{SET} (4)	DAC Settling Time	Low Power Mode (DACCTRLx.CCTRL = 0)	—	—	$1/D_{SAMPmin}+1$	µs	+/-4LSB of final value for step size≤100 LSB @ C_{LOAD} & R_{LOAD} w/ AVDD=3.3V
			High Performance Mode (DACCTRLx.CCTRL = 2)			$1/D_{SAMPmin}+1$		
DAC_17	T_{SET_FS} (4)	DAC Full Scale Settling Time	Low Power Mode (DACCTRLx.CCTRL = 0)	—	—	$7/D_{SAMPmin}+1$	µs	+/-4 LSB of final value for step size from 10% to 90% @ C_{LOAD} & R_{LOAD} w/ AVDD=3.3V
			High Performance Mode (DACCTRLx.CCTRL = 2)			$7/D_{SAMPmin}+1$		
DIFFERENTIAL MODE (2,3,6)								

.....continued

AC CHARACTERISTICS				Standard Operating Conditions: VDD = AVDD = 1.62V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ T_A ≤ +85°C for Industrial				
Param. No.	**Symbol**	**Characteristics**		**Min.**	**Typ.**	**Max.**	**Units**	**Conditions**
DDAC_19	INL	Integral Non Linearity	REFSEL = AVDD CTRLB.REFSEL = 0x1	-2.5	-	2.5	LSB	AVDD=3.3V w/ C_{LOAD} & R_{LOAD} High Power Mode (DACCTRLx.CCTRL=2) (Max sampling rate 1MSPS) A clipping value of 150mV is applied to both RampStart and RampEnd in the DAC Linearity calculation according to DAC_7 parameter.
			REFSEL = VREFAU (2.5V) CTRLB.REFSEL = 0x0	-2.9	-	2.5		
			REFSEL = VREFAB (2.5V) CTRLB.REFSEL = 0x2	-2.1	-	2.1		
			REFSEL = INTREF (2.4V) CTRLB.REFSEL = 0x3	-5.1	-	5.0		
DDAC_21	DNL	Differential Non Linearity	REFSEL = AVDD CTRLB.REFSEL = 0x1	-2.3	-	1.0	LSB	
			REFSEL = VREFAU (2.5V) CTRLB.REFSEL = 0x0	-2.6	-	2.7		
			REFSEL = VREFAB (2.5V) CTRLB.REFSEL = 0x2	-2.1	-	1.2		
			REFSEL = INTREF (2.4V) CTRLB.REFSEL = 0x3	-6.4	-	5.9		
DDAC_23	G_{ERR}	Gain Error	REFSEL = AVDD CTRLB.REFSEL = 0x1	-31.8	-	-18.2	LSB	
			REFSEL = VREFAU (2.5V) CTRLB.REFSEL = 0x0	-31.9	-	-16.4		
			REFSEL = VREFAB (2.5V) CTRLB.REFSEL = 0x2	-37.8	-	-17.6		
			REFSEL = INTREF (2.4V) CTRLB.REFSEL = 0x3	-105.8	-	59.2		
DDAC_25	E_{OFF}	Offset Error	REFSEL = AVDD CTRLB.REFSEL = 0x1	4.9	-	15.2	LSB	
			REFSEL = VREFAU (2.5V) CTRLB.REFSEL = 0x0	-1.6	-	16.8		
			REFSEL = VREFAB (2.5V) CTRLB.REFSEL = 0x2	4.3	-	19.1		
			REFSEL = INTREF (2.4V) CTRLB.REFSEL = 0x3	-34.0	-	51.9		

.....continued

AC CHARACTERISTICS				Standard Operating Conditions: VDD = AVDD = 1.62V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ T_A ≤ +85°C for Industrial				
Param. No.	**Symbol**	**Characteristics**		**Min.**	**Typ.**	**Max.**	**Units**	**Conditions**
DDAC_27	ENOB (5)	Effective Number of Bit	REFSEL = AVDD CTRLB.REFSEL = 0x1	10.0	-	10.9	Bits	AVDD=3.3V w/ C_{LOAD} & R_{LOAD}. Digital sine magnitude: 0x080 to 0xF7F High Power Mode (DACCTRLx.CCTRL=2) (Max sampling rate 1MSPS)
			REFSEL = VREFAU (2.5V) CTRLB.REFSEL = 0x0	10.7	-	12.5		
			REFSEL = VREFAB (2.5V) CTRLB.REFSEL = 0x2	10.6	-	12.3		
			REFSEL = INTREF (2.4V) CTRLB.REFSEL = 0x3	9.6	-	11.2		
DDAC_29	SNR (5)	Signal To Noise Ratio	REFSEL = AVDD CTRLB.REFSEL = 0x1	72.1	-	79.0	dB	
			REFSEL = VREFAU (2.5V) CTRLB.REFSEL = 0x0	72.5	-	78.1		
			REFSEL = VREFAB (2.5V) CTRLB.REFSEL = 0x2	71.6	-	78.7		
			REFSEL = INTREF (2.4V) CTRLB.REFSEL = 0x3	59.9	-	69.8		
DDAC_31	THD (1,5)	Total Harmonic Distortion	REFSEL = AVDD CTRLB.REFSEL = 0x1	-68.0	-	-62.4	dB	
			REFSEL = VREFAU (2.5V) CTRLB.REFSEL = 0x0	-85.2	-	-67.6		
			REFSEL = VREFAB (2.5V) CTRLB.REFSEL = 0x2	-80.3	-	-66.8		
			REFSEL = INTREF (2.4V) CTRLB.REFSEL = 0x3	-82.5	-	-67.3		
SINGLE ENDED MODE (2,3,6)								

.....continued

AC CHARACTERISTICS				Standard Operating Conditions: VDD = AVDD = 1.62V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ T_A ≤ +85°C for Industrial				
Param. No.	**Symbol**	**Characteristics**		**Min.**	**Typ.**	**Max.**	**Units**	**Conditions**
SDAC_19	INL	Integral Non Linearity	REFSEL = AVDD CTRLB.REFSEL = 0x1	-3.4	-	3.2	LSB	AVDD=3.3V w/ C_{LOAD} & R_{LOAD} High Power Mode (DACCTRLx.CCTRL=2) (Max sampling rate 1MSPS) A clipping value of 150mV is applied to both RampStart and RampEnd in the DAC Linearity calculation according to DAC_7 parameter.
			REFSEL = VREFAU (2.5V) CTRLB.REFSEL = 0x0	-5.1	-	4.7		
			REFSEL = VREFAB (2.5V) CTRLB.REFSEL = 0x2	-4.4	-	4.2		
			REFSEL = INTREF (2.4V) CTRLB.REFSEL = 0x3	-12.3	-	8.2		
SDAC_21	DNL	Differential Non Linearity	REFSEL = AVDD CTRLB.REFSEL = 0x1	-3.7	-	2.7	LSB	
			REFSEL = VREFAU (2.5V) CTRLB.REFSEL = 0x0	-4.8	-	5.1		
			REFSEL = VREFAB (2.5V) CTRLB.REFSEL = 0x2	-3.6	-	3.3		
			REFSEL = INTREF (2.4V) CTRLB.REFSEL = 0x3	-13.0	-	11.3		
SDAC_23	G_{ERR}	Gain Error	REFSEL = AVDD CTRLB.REFSEL = 0x1	-38.4	-	-19.1	LSB	
			REFSEL = VREFAU (2.5V) CTRLB.REFSEL = 0x0	-39.8	-	-19.8		
			REFSEL = VREFAB (2.5V) CTRLB.REFSEL = 0x2	-45.3	-	-20.6		
			REFSEL = INTREF (2.4V) CTRLB.REFSEL = 0x3	-111.9	-	53.8		
SDAC_25	E_{OFF}	Offset Error	REFSEL = AVDD CTRLB.REFSEL = 0x1	3.5	-	16.8	LSB	
			REFSEL = VREFAU (2.5V) CTRLB.REFSEL = 0x0	4.9	-	22.2		
			REFSEL = VREFAB (2.5V) CTRLB.REFSEL = 0x2	8.7	-	26.4		
			REFSEL = INTREF (2.4V) CTRLB.REFSEL = 0x3	8.7	-	27.4		

.....continued

AC CHARACTERISTICS				Standard Operating Conditions: VDD = AVDD = 1.62V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ T_A ≤ +85°C for Industrial				
Param. No.	**Symbol**	**Characteristics**		**Min.**	**Typ.**	**Max.**	**Units**	**Conditions**
SDAC_27	ENOB (5)	Effective Number of Bit	REFSEL = AVDD CTRLB.REFSEL = 0x1	9.9	-	11.0	Bits	AVDD=3.3V w/ C_{LOAD} & R_{LOAD}. Digital sine magnitude: 0x080 to 0xF7F High Power Mode (DACCTRLx.CCTRL=2) (Max sampling rate 1MSPS)
			REFSEL = VREFAU (2.5V) CTRLB.REFSEL = 0x0	9.8	-	11.3		
			REFSEL = VREFAB (2.5V) CTRLB.REFSEL = 0x2	9.7	-	11.2		
			REFSEL = INTREF (2.4V) CTRLB.REFSEL = 0x3	8.7	-	10.2		
SDAC_29	SNR (5)	Signal To Noise Ratio	REFSEL = AVDD CTRLB.REFSEL = 0x1	65.7	-	76.0	dB	
			REFSEL = VREFAU (2.5V) CTRLB.REFSEL = 0x0	63.6	-	75.5		
			REFSEL = VREFAB (2.5V) CTRLB.REFSEL = 0x2	63.6	-	75.5		
			REFSEL = INTREF (2.4V) CTRLB.REFSEL = 0x3	55.0	-	65.0		
SDAC_31	THD (1,5)	Total Harmonic Distortion	REFSEL = AVDD CTRLB.REFSEL = 0x1	-68.9	-	-62.4	dB	
			REFSEL = VREFAU (2.5V) CTRLB.REFSEL = 0x0	-71.2	-	-63.3		
			REFSEL = VREFAB (2.5V) CTRLB.REFSEL = 0x2	-70.5	-	-62.8		
			REFSEL = INTREF (2.4V) CTRLB.REFSEL = 0x3	-70.6	-	-62.6		

Notes:

1. Value taken over 7 harmonics.
2. 12-bit mode.
3. Over V_{OUT} range defined by DAC_7 parameter.
4. Assuming DAC is configured and that first conversion is done.
5. Characterized with 1kHz sine wave.
6. DAC functional device operation with external VREF < 2.4V is guaranteed, but not characterized. DAC will function, but with degraded performance. DAC accuracy is limited by users application noise/accuracy on AVDD, AVSS and VREF accuracy/drift.

49.22 ADC Electrical Specifications

Table 49-25. ADC Electrical Specifications

AC CHARACTERISTICS			Standard Operating Conditions: VDD = AVDD = 1.62V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ T_A ≤ +85°C for Industrial				
Param. No.	**Symbol**	**Characteristics**	**Min.**	**Typ.**	**Max.**	**Units**	**Conditions**
Device Supply							
ADC_1	AVDD	ADC Module Supply	AVDD(min)	—	AVDD(max)	V	—
Reference Inputs							
ADC_3	VREF (1)	ADC Reference Voltage	2.4	—	AVDD	V	VREF = AVDD (REFCTRL.REFSEL = 0x5)
				—	AVDD-0.6		VREF != AVDD (REFCTRL.REFSEL != 0x5)
Analog Input Range							
ADC_7	A_{FS}	Full-Scale Analog Input Signal Range	AVSS	—	VREF	V	Single-Ended Mode
			-VREF	—	VREF		Differential Mode
ADC_9	V_{CMIN}	Input Common Mode Voltage	0.7	—	VREF-0.7	V	—
ADC_11	$T_{SETTLING}$	ADC Stabilization Time	—	10	—	µs	CTRLA.ENABLE=1 or CTRLA.ONDEMAND=1

Note:

1. ADC functional device operation with external VREF<2.4V is functional, but not characterized. ADC will function, but with degraded accuracy of approximately ~((0.06 * 2n) / VREF) LSB's over full scale range, where "n"=#bits. ADC accuracy is limited by internal VREF accuracy + drift, MCU generated noise plus users application noise/accuracy on AVDD, AVSS.

Table 49-26. ADC Single Ended Mode Electrical Specifications

AC CHARACTERISTICS			Standard Operating Conditions: VDD = AVDD = 1.62V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ T_A ≤ +85°C for Industrial				
Param. No.	**Symbol**	**Characteristics**	**Min.**	**Typ.**	**Max.**	**Units**	**Conditions**
SINGLE ENDED MODE ADC Accuracy (3)							
SADC_11	Res	Resolution	8	—	12	Bits	Selectable 8, 10, 12 bit Resolution Ranges
SADC_13	ENOB (1,2)	Effective Number of bits	8.6	—	—	Bits	AVDD = 3.3V VREF=AVDD (REFCTRL.REFSEL = 0x5) SAMPCTRL.SAMPLEN=3
SADC_15			8.5	—	—	Bits	AVDD = 3.3V VREF=VREFA/B=2.7V (REFCTRL.REFSEL= 0x3 or 0x4) SAMPCTRL.SAMPLEN=3

.....continued

AC CHARACTERISTICS			Standard Operating Conditions: VDD = AVDD = 1.62V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ T_A ≤ +85°C for Industrial				
Param. No.	Symbol	Characteristics	Min.	Typ.	Max.	Units	Conditions
SADC_19	INL	Integral Nonlinearity	-7.4	—	7.3	LSB	AVDD = 3.3V VREF=AVDD (REFCTRL.REFSEL = 0x5) SAMPCTRL.SAMPLEN=3
SADC_21			-8.7	—	8.9	LSB	AVDD = 3.3V VREF=VREFA/B=2.7V (REFCTRL.REFSEL= 0x3 or 0x4) SAMPCTRL.SAMPLEN=3
SADC_25	DNL	Differential Nonlinearity	-0.99	—	1.4	LSB	AVDD = 3.3V VREF=AVDD (REFCTRL.REFSEL = 0x5) SAMPCTRL.SAMPLEN=3
SADC_27			-0.99	—	2.1	LSB	AVDD = 3.3V VREF=VREFA/B=2.7V (REFCTRL.REFSEL= 0x3 or 0x4) SAMPCTRL.SAMPLEN=3
SADC_31	GERR	Gain Error	-24	—	14	%	AVDD = 3.3V VREF=AVDD (REFCTRL.REFSEL = 0x5) SAMPCTRL.SAMPLEN=3
SADC_33			-37	—	18	%	AVDD = 3.3V VREF=VREFA/B=2.7V (REFCTRL.REFSEL= 0x3 or 0x4) SAMPCTRL.SAMPLEN=3
SADC_37	EOFF	Offset Error	-42	—	43	mV	AVDD = 3.3V VREF=AVDD (REFCTRL.REFSEL = 0x5) SAMPCTRL.SAMPLEN=3
SADC_39			-41	—	60	mV	AVDD = 3.3V VREF=VREFA/B=2.7V (REFCTRL.REFSEL= 0x3 or 0x4) SAMPCTRL.SAMPLEN=3

.....continued

AC CHARACTERISTICS			Standard Operating Conditions: VDD = AVDD = 1.62V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ T_A ≤ +85°C for Industrial				
Param. No.	**Symbol**	**Characteristics**	**Min.**	**Typ.**	**Max.**	**Units**	**Conditions**
SADC_43	TUE	Total Unadjusted Error	2.9	—	27	LSB	AVDD = 3.3V VREF=AVDD (REFCTRL.REFSEL = 0x5) SAMPCTRL.SAMPLEN=3
SADC_45			4.3	—	47	LSB	AVDD = 3.3V VREF=VREFA/B=2.7V (REFCTRL.REFSEL= 0x3 or 0x4) SAMPCTRL.SAMPLEN=3
SINGLE ENDED MODE ADC Dynamic Performance (1,2,3)							

.....continued

AC CHARACTERISTICS			Standard Operating Conditions: VDD = AVDD = 1.62V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ T_A ≤ +85°C for Industrial				
Param. No.	Symbol	Characteristics	Min.	Typ.	Max.	Units	Conditions
SADC_49	SINAD	Signal to Noise and Distortion	53.6	—	—	dB	AVDD = 3.3V VREF=AVDD (REFCTRL.REFSEL = 0x5) SAMPCTRL.SAMPLEN=3
SADC_50			52.8	—	—		AVDD = 3.3V VREF=VREFA/B=2.7V (REFCTRL.REFSEL= 0x3 or 0x4) SAMPCTRL.SAMPLEN=3
SADC_51	SNR	Signal to Noise ratio	55.3	—	—		AVDD = 3.3V VREF=AVDD (REFCTRL.REFSEL = 0x5) SAMPCTRL.SAMPLEN=3
SADC_52			54.2	—	—		AVDD = 3.3V VREF=VREFA/B=2.7V (REFCTRL.REFSEL= 0x3 or 0x4) SAMPCTRL.SAMPLEN=3
SADC_53	SFDR	Spurious Free Dynamic Range	57.6	—	—		AVDD = 3.3V VREF=AVDD (REFCTRL.REFSEL = 0x5) SAMPCTRL.SAMPLEN=3
SADC_54			57.3	—	—		AVDD = 3.3V VREF=VREFA/B=2.7V (REFCTRL.REFSEL= 0x3 or 0x4) SAMPCTRL.SAMPLEN=3
SADC_55	THD (4)	Total Harmonic Distortion	—	—	-56.1		AVDD = 3.3V VREF=AVDD (REFCTRL.REFSEL = 0x5) SAMPCTRL.SAMPLEN=3
SADC_56			—	—	-55.7		AVDD = 3.3V VREF=VREFA/B=2.7V (REFCTRL.REFSEL= 0x3 or 0x4) SAMPCTRL.SAMPLEN=3

Notes:

1. Characterized with an analog input sine wave = (F_{TP}(max) / 100). Example: F_{TP}(max)=1MSPS/100 = 10kHz sine wave.
2. Sine wave peak amplitude = 96% ADC Full Scale amplitude input with 12bit resolution.
3. Spec values collected under the following additional conditions:
   - All registers at reset default value unless otherwise stated.
   - 12bit resolution mode
4. Value taken over 7 harmonics.

Table 49-27. ADC Differential Mode Electrical Specifications

AC CHARACTERISTICS			Standard Operating Conditions: VDD = AVDD = 1.62V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ T_A ≤ +85°C for Industrial				
Param. No.	**Symbol**	**Characteristics**	**Min.**	**Typ.**	**Max.**	**Units**	**Conditions**
DIFFERENTIAL MODE ADC Accuracy (3)							
DADC_11	Res	Resolution	8	—	12	bits	Selectable 8, 10, 12 bit Resolution Ranges
DADC_13	ENOB (1,2)	Effective Number of bits	9.7	—	—	bits	AVDD = 3.3V VREF=AVDD (REFCTRL.REFSEL = 0x5) SAMPCTRL.OFFCOMP=1 SAMPCTRL.REFCOMP=0
DADC_15			9.5	—	—	bits	AVDD = 3.3V VREF=VREFA/B=2.7V (REFCTRL.REFSEL = 0x3 or 0x4) SAMPCTRL.OFFCOMP=1 SAMPCTRL.REFCOMP=1
DADC_19	INL	Integral Nonlinearity	-3.5	—	4.1	LSB	AVDD = 3.3V VREF=AVDD (REFCTRL.REFSEL = 0x5) SAMPCTRL.OFFCOMP=1 SAMPCTRL.REFCOMP=0
DADC_21			-5.4	—	4.8	LSB	AVDD = 3.3V VREF=VREFA/B=2.7V (REFCTRL.REFSEL = 0x3 or 0x4) SAMPCTRL.OFFCOMP=1 SAMPCTRL.REFCOMP=1
DADC_25	DNL	Differential Nonlinearity	-0.99	—	1.0	LSB	AVDD = 3.3V VREF=AVDD (REFCTRL.REFSEL = 0x5) SAMPCTRL.OFFCOMP=1 SAMPCTRL.REFCOMP=0
DADC_27			-0.99	—	1.8	LSB	AVDD = 3.3V VREF=VREFA/B=2.7V (REFCTRL.REFSEL = 0x3 or 0x4) SAMPCTRL.OFFCOMP=1 SAMPCTRL.REFCOMP=1

.....continued

AC CHARACTERISTICS			Standard Operating Conditions: VDD = AVDD = 1.62V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ T_A ≤ +85°C for Industrial				
Param. No.	**Symbol**	**Characteristics**	**Min.**	**Typ.**	**Max.**	**Units**	**Conditions**
DADC_31	GERR	Gain Error	-19	—	7	%	AVDD = 3.3V VREF=AVDD (REFCTRL.REFSEL = 0x5) SAMPCTRL.OFFCOMP=1 SAMPCTRL.REFCOMP=0
DADC_33			-14	—	22	%	AVDD = 3.3V VREF=VREFA/B=2.7V (REFCTRL.REFSEL = 0x3 or 0x4) SAMPCTRL.OFFCOMP=1 SAMPCTRL.REFCOMP=1
DADC_37	EOFF	Offset Error	-5.6	—	2.1	mV	AVDD = 3.3V VREF=AVDD (REFCTRL.REFSEL = 0x5) SAMPCTRL.OFFCOMP=1 SAMPCTRL.REFCOMP=0
DADC_39			-4.9	—	2.3	mV	AVDD = 3.3V VREF=VREFA/B=2.7V (REFCTRL.REFSEL = 0x3 or 0x4) SAMPCTRL.OFFCOMP=1 SAMPCTRL.REFCOMP=1
DADC_43	TUE	Total Unadjusted Error	2.3	—	8.0	LSB	AVDD = 3.3V VREF=AVDD (REFCTRL.REFSEL = 0x5) SAMPCTRL.OFFCOMP=1 SAMPCTRL.REFCOMP=0
DADC_45			3.5	—	11	LSB	AVDD = 3.3V VREF=VREFA/B=2.7V (REFCTRL.REFSEL = 0x3 or 0x4) SAMPCTRL.OFFCOMP=1 SAMPCTRL.REFCOMP=1
DIFFERENTIAL MODE ADC Dynamic Performance (1,2,3)							

.....continued

AC CHARACTERISTICS			Standard Operating Conditions: VDD = AVDD = 1.62V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ T_A ≤ +85°C for Industrial				
Param. No.	Symbol	Characteristics	Min.	Typ.	Max.	Units	Conditions
DADC_49	SINAD	Signal to Noise and Distortion	59.9	—	—	dB	AVDD = 3.3V VREF=AVDD (REFCTRL.REFSEL =0x5) SAMPCTRL.OFFCOMP=1 SAMPCTRL.REFCOMP=0
DADC_50			59.2	—	—		AVDD = 3.3V VREF=VREFA/B=2.7V (REFCTRL.REFSEL = 0x3 or 0x4) SAMPCTRL.OFFCOMP=1 SAMPCTRL.REFCOMP=1
DADC_51	SNR	Signal to Noise ratio	62.4	—	—		AVDD = 3.3V VREF=AVDD (REFCTRL.REFSEL =0x5) SAMPCTRL.OFFCOMP=1 SAMPCTRL.REFCOMP=0
DADC_52			61.4	—	—		AVDD = 3.3V VREF=VREFA/B=2.7V (REFCTRL.REFSEL = 0x3 or 0x4) SAMPCTRL.OFFCOMP=1 SAMPCTRL.REFCOMP=1
DADC_53	SFDR	Spurious Free Dynamic Range	61.9	—	—		AVDD = 3.3V VREF=AVDD (REFCTRL.REFSEL =0x5) SAMPCTRL.OFFCOMP=1 SAMPCTRL.REFCOMP=0
DADC_54			63.6	—	—		AVDD = 3.3V VREF=VREFA/B=2.7V (REFCTRL.REFSEL = 0x3 or 0x4) SAMPCTRL.OFFCOMP=1 SAMPCTRL.REFCOMP=1
DADC_55	THD (4)	Total Harmonic Distortion	—	—	-61.4		AVDD = 3.3V VREF=AVDD (REFCTRL.REFSEL =0x5) SAMPCTRL.OFFCOMP=1 SAMPCTRL.REFCOMP=0
DADC_56			—	—	-63.5		AVDD = 3.3V VREF=VREFA/B=2.7V (REFCTRL.REFSEL = 0x3 or 0x4) SAMPCTRL.OFFCOMP=1 SAMPCTRL.REFCOMP=1

.....continued

AC CHARACTERISTICS			Standard Operating Conditions: VDD = AVDD = 1.62V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ T_A ≤ +85°C for Industrial				
Param. No.	Symbol	Characteristics	Min.	Typ.	Max.	Units	Conditions

Notes:

1. Characterized with an analog input sine wave = (F_{TP}(max) / 100). Example: F_{TP}(max)=1MSPS/100 = 10kHz sine wave.
2. Sine wave peak amplitude = 96% ADC Full Scale amplitude input with 12bit resolution.
3. Spec values collected under the following additional conditions:
   - All registers at reset default value unless otherwise stated
   - 12bit resolution mode
4. Value taken over 7 harmonics.

Table 49-28. ADC Conversion Electrical Requirements

AC CHARACTERISTICS			Standard Operating Conditions: VDD = AVDD = 1.62V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ T_A ≤ +85°C for Industrial				
Param. No.	**Symbol**	**Characteristics**	**Min.**	**Typ.**	**Max.**	**Units**	**Conditions**
ADC Clock Requirements							
ADC_57	TAD	ADC Clock Period	62.5	—	6250	ns	—
ADC_58	f_{GCLK_ADC}	ADC Module GCLK max input freq	—	—	FCLK_51	MHz	FCLK_51: Refer to Maximum Clock Frequencies Electrical Specifications
ADC Single-Ended Throughput Rates							
ADC_59	FTP (Single-Ended Mode) (1)	Throughput Rate (Single-Ended)	—	—	1.231	MSPS	12-bit resolution, Rsource ≤298 Ω, SAMPCTRL.OFFCOMP=0 SAMPCTRL.SAMPLEN=0
			—	—	1.333		10-bit resolution, Rsource ≤633 Ω, SAMPCTRL.OFFCOMP=0 SAMPCTRL.SAMPLEN=0
			—	—	1.6		8-bit resolution, Rsource ≤1,103 Ω, SAMPCTRL.OFFCOMP=0 SAMPCTRL.SAMPLEN=0
			—	—	1	MSPS	12-bit resolution, Rsource ≤6,335Ω SAMPCTRL.OFFCOMP=1 SAMPCTRL.SAMPLEN=n/a
			—	—	1.067		10-bit resolution, Rsource ≤7,677 Ω SAMPCTRL.OFFCOMP=1 SAMPCTRL.SAMPLEN=n/a
			—	—	1.231		8-bit resolution, Rsource ≤9,556 Ω SAMPCTRL.OFFCOMP=1 SAMPCTRL.SAMPLEN=n/a
ADC Differential Mode Throughput Rates							

.....continued

AC CHARACTERISTICS			Standard Operating Conditions: VDD = AVDD = 1.62V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ T_A ≤ +85°C for Industrial				
Param. No.	**Symbol**	**Characteristics**	**Min.**	**Typ.**	**Max.**	**Units**	**Conditions**
ADC_61	F_{TP} (Differential Mode)	Throughput Rate (Differential Mode)	—	—	1.231	MSPS	12-bit resolution, Rsource ≤298 Ω, SAMPCTRL.OFFCOMP=0 SAMPCTRL.SAMPLEN=0
			—	—	1.455		10-bit resolution, Rsource ≤633 Ω, SAMPCTRL.OFFCOMP=0 SAMPCTRL.SAMPLEN=0
			—	—	1.778		8-bit resolution, Rsource ≤1,103 Ω, SAMPCTRL.OFFCOMP=0 SAMPCTRL.SAMPLEN=0
			—	—	1	MSPS	12-bit resolution, Rsource ≤6,335Ω SAMPCTRL.OFFCOMP=1 SAMPCTRL.SAMPLEN=n/a
			—	—	1.143		10-bit resolution, Rsource ≤7,667 Ω SAMPCTRL.OFFCOMP=1 SAMPCTRL.SAMPLEN=n/a
			—	—	1.333		8-bit resolution, Rsource ≤9,556 Ω SAMPCTRL.OFFCOMP=1 SAMPCTRL.SAMPLEN=n/a

Note:

1. ADC Throughput Rate F_{TP} = ((1 / ((TSAMP + TCNV) * TAD)) / (# of user analog inputs in use on specific target ADC module)).
   – Specification values assume only one AINx channel in use

Table 49-29. ADC Conversion Electrical Requirements (continued)

AC CHARACTERISTICS			Standard Operating Conditions: VDD = AVDD = 1.62V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ T_A ≤ +85°C for Industrial				
Param. No.	**Symbol**	**Characteristics**	**Min.**	**Typ.**	**Max.**	**Units**	**Conditions**
ADC_63	T_{SAMP} (1,2)	ADC Sample Time	1	—	—	TAD	12-bit resolution, TAD(min), Ext Analog Input Rsource ≤ 298 Ω
							10-bit resolution, TAD(min), Ext Analog Input Rsource ≤ 633 Ω
							8-bit resolution, TAD(min), Ext Analog Input Rsource ≤ 1,103 Ω
			2	—	—		12-bit resolution, TAD(min), Ext Analog Input Rsource ≤ 2,310 Ω
							10-bit resolution, TAD(min), Ext Analog Input Rsource ≤ 2,981 Ω
							8-bit resolution, TAD(min), Ext Analog Input Rsource ≤ 3,921 Ω
			3	—	—		12-bit resolution, TAD(min), Ext Analog Input Rsource ≤ 4,323 Ω
							10-bit resolution, TAD(min), Ext Analog Input Rsource ≤ 5,329 Ω
							8-bit resolution, TAD(min), Ext Analog Input Rsource ≤ 6,738 Ω
			4	—	—		12-bit resolution, TAD(min), Ext Analog Input Rsource ≤ 6,335 Ω
							10-bit resolution, TAD(min), Ext Analog Input Rsource ≤ 7,678 Ω
							8-bit resolution, TAD(min), Ext Analog Input Rsource ≤ 9,556 Ω
			5	—	—		12-bit resolution, TAD(min), Ext Analog Input Rsource ≤ 8,348 Ω
							10-bit resolution, TAD(min), Ext Analog Input Rsource ≤ 10,026 Ω
							8-bit resolution, TAD(min), Ext Analog Input Rsource ≤ 12,374 Ω

.....continued

AC CHARACTERISTICS			Standard Operating Conditions: VDD = AVDD = 1.62V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ T_A ≤ +85°C for Industrial				
Param. No.	**Symbol**	**Characteristics**	**Min.**	**Typ.**	**Max.**	**Units**	**Conditions**
ADC_63	T_{SAMP} (1,2)	ADC Sample Time	6	—	—	TAD	12-bit resolution, TAD(min), Ext Analog Input Rsource ≤ 10,361 Ω
							10-bit resolution, TAD(min), Ext Analog Input Rsource ≤ 12,374 Ω
							8-bit resolution, TAD(min), Ext Analog Input Rsource ≤ 15,192 Ω
			3000	—	—	ns	With DAC as input
			10000	—	—	ns	With Internal BANDGAP as input
ADC_65	T_{CNV}	Conversion Time (Single-Ended Mode)	12			TAD	12-bit resolution
			11				10-bit resolution
			9				8-bit resolution
ADC_67		Conversion Time (Differential Mode)	12			TAD	12-bit resolution
			10				10-bit resolution
			8				8-bit resolution
ADC_69	C_{SAMPLE}	ADC Internal Sample Cap	—	—	3.2	pF	—
ADC_71	R_{SAMPLE}	ADC Internal impedance	—	—	1715	Ω	—

Notes:

1. When SAMPCTRL.OFFCOMP = 0:
   - T_{SAMP} = ((((R_{SAMPLE} + R_{SOURCE}) * C_{SAMPLE} * (#Bits Resolution+2) * ln(2)) / TAD) +1 rounded down to nearest whole integer
   - Use the above formula for R_{SOURCE} values exceeding TSAMP = 6.
   - SAMPCTRL.SAMPLEN = (T_{SAMP} / TAD - 1)
2. When SAMPCTRL.OFFCOMP=1:
   - T_{SAMP} = 4 (Forced by ADC Hardware)
   - SAMPCTRL.SAMPLEN = (n/a, Ignored by ADC Hardware)

49.23 OPAMP Electrical Specifications

Table 49-30. OPAMP Electrical Specifications (1,2)

AC CHARACTERISTICS			Standard Operating Conditions: VDD = AVDD = 1.62V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ T_A ≤ +85°C for Industrial				
Param. No.	**Symbol**	**Characteristics**	**Min.**	**Typ.**	**Max.**	**Units**	**Conditions**
OA_1	VCMR	Common Mode Input Voltage Range	0.012	—	AVDD - 0.15	V	—
OA_2	CMRR	Common Mode Rejection Ratio	-65	—	—	dB	VCM = AVDD/2
OA_3	VOFFSET	Op amp Offset Voltage	-6	—	6	mV	—
OA_4	VIN	Input Voltage Range	AVSS	—	AVDD	V	—
OA_5	ILKG	Input leakage current	—	—	DI_13	µA	DI_13: Refer to I/O Pin Electrical Specifications
OA_6	PSRR	Power Supply Rejection Ratio	—	64.8	—	dB	@ 10 kHz, OPAMPCTRLx.BIAS = 0x3
				44.8			@ 100 kHz, OPAMPCTRLx.BIAS = 0x3
				25.1			@ 1 MHz, OPAMPCTRLx.BIAS = 0x3
OA_8	VOH	Amplifier Output Voltage High	AVDD - 0.3	—	—	V	ISOURCE @ 5mA
			AVDD - 0.2				ISOURCE @ 3mA
			AVDD - 0.07				ISOURCE @ 1mA
OA_9	VOL	Amplifier Output Voltage Low	—	—	AVSS + 0.4	V	ISINK @ 5mA
					AVSS + 0.25		ISINK @ 3mA
					AVSS + 0.09		ISINK @ 1mA
OA_10	TON	Enable to Valid Output	—	2.7	—	µs	OPAMPCTRLx.BIAS = 0x3
				6.4			OPAMPCTRLx.BIAS = 0x2
				21.5			OPAMPCTRLx.BIAS = 0x1
				88.5			OPAMPCTRLx.BIAS = 0x0
OA_12	IOS	Input Offset Current	—	—	DI_13	µA	DI_13: Refer to I/O Pin Electrical Specifications
OA_13	IB	Input Bias Current	—	—	DI_13	µA	DI_13: Refer to I/O Pin Electrical Specifications
OA_14	SR	Slew Rate	—	2.1	—	V/µs	OPAMPCTRLx.BIAS = 0x3, OPAMP Vout 10-90%, unity gain, 32 pF load
				0.9			OPAMPCTRLx.BIAS = 0x2, OPAMP Vout 10-90%, unity gain, 32 pF load
				0.25			OPAMPCTRLx.BIAS = 0x1, OPAMP Vout 10-90%, unity gain, 32 pF load
				0.06			OPAMPCTRLx.BIAS = 0x0, OPAMP Vout 10-90%, unity gain, 32 pF load

.....continued

AC CHARACTERISTICS			Standard Operating Conditions: VDD = AVDD = 1.62V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ T_A ≤ +85°C for Industrial				
Param. No.	Symbol	Characteristics	Min.	Typ.	Max.	Units	Conditions
OA_15	GBW1	Non Unity Gain Bandwidth	3.57	—	—	MHz	Non-inverting gain of 16 Config: OPAMPCTRLx.BIAS=3, POTMUX=0x7, Cload=47pF, Rload=1kΩ. Vin=25mVpp and Vout DC at half AVDD. GBW1 is measured at Gain=0dB.
			1.70				Non-inverting gain of 16 Config: OPAMPCTRLx.BIAS=2, POTMUX=0x7, Cload=47pF, Rload=1kΩ. Vin=25mVpp and Vout DC at half AVDD. GBW1 is measured at Gain=0dB.
			0.84				Non-inverting gain of 2 Config: OPAMPCTRLx.BIAS=1, POTMUX=0x2, Cload=47pF, Rload=1kΩ. Vin=25mVpp and Vout DC at half AVDD. GBW1 is measured at Gain=0dB.
			0.23				Non-inverting gain of 2 Config: OPAMPCTRLx.BIAS=0, POTMUX=0x2, Cload=47pF, Rload=1kΩ. Vin=25mVpp and Vout DC at half AVDD. GBW1 is measured at Gain=0dB.

.....continued

AC CHARACTERISTICS			Standard Operating Conditions: VDD = AVDD = 1.62V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ T_A ≤ +85°C for Industrial				
Param. No.	**Symbol**	**Characteristics**	**Min.**	**Typ.**	**Max.**	**Units**	**Conditions**
OA_16	AV1	Non Unity Gain	—	16	—	V/V	Non-inverting gain of 16 Config: OPAMPCTRLx.BIAS=3, POTMUX=0x7, Cload=47pF, Rload=1kΩ. Fin=50kHz, Vin=25mVpp and Vout DC at half AVDD.
				16			Non-inverting gain of 16 Config: OPAMPCTRLx.BIAS=2, POTMUX=0x7, Cload=47pF, Rload=1kΩ. Fin=50kHz, Vin=25mVpp and Vout DC at half AVDD.
				2			Non-inverting gain of 2 Config: OPAMPCTRLx.BIAS=1, POTMUX=0x2, Cload=47pF, Rload=1kΩ. Fin=50kHz, Vin=25mVpp and Vout DC at half AVDD.
				2			Non-inverting gain of 2 Config: OPAMPCTRLx.BIAS=0, POTMUX=0x2, Cload=47pF, Rload=1kΩ. Fin=50kHz, Vin=25mVpp and Vout DC at half AVDD.
OA_17	PM1	Non Unity Phase Margin	49.1	—	—	Degrees	Non-inverting gain of 16 Config: OPAMPCTRLx.BIAS=3, POTMUX=0x7, Cload=47pF, Rload=1kΩ. Vin=25mVpp and Vout DC at half AVDD.
			57.7				Non-inverting gain of 16 Config: OPAMPCTRLx.BIAS=2, POTMUX=0x7, Cload=47pF, Rload=1kΩ. Vin=25mVpp and Vout DC at half AVDD.
			88.6				Non-inverting gain of 2 Config: OPAMPCTRLx.BIAS=1, POTMUX=0x2, Cload=47pF, Rload=1kΩ. Vin=25mVpp and Vout DC at half AVDD.
			87.2				Non-inverting gain of 2 Config: OPAMPCTRLx.BIAS=0, POTMUX=0x2, Cload=47pF, Rload=1kΩ. Vin=25mVpp and Vout DC at half AVDD.

.....continued

AC CHARACTERISTICS			Standard Operating Conditions: VDD = AVDD = 1.62V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ T_A ≤ +85°C for Industrial				
Param. No.	Symbol	Characteristics	Min.	Typ.	Max.	Units	Conditions
OA_19	GBW2	Unity Gain Bandwidth	4.29	—	—	MHz	Unity gain follower Config: OPAMPCTRLx.BIAS=3, POTMUX=0x0, Cload=47pF, Rload=1kΩ. Vin=25mVpp and Vout DC at half AVDD. GBW2 is measured at Phase=-45deg.
			2.02				Unity gain follower Config: OPAMPCTRLx.BIAS=2, POTMUX=0x0, Cload=47pF, Rload=1kΩ. Vin=25mVpp and Vout DC at half AVDD. GBW2 is measured at Phase=-45deg.
			0.75				Unity gain follower Config: OPAMPCTRLx.BIAS=1, POTMUX=0x0, Cload=47pF, Rload=1kΩ. Vin=25mVpp and Vout DC at half AVDD. GBW2 is measured at Phase=-45deg.
			0.21				Unity gain follower Config: OPAMPCTRLx.BIAS=0, POTMUX=0x0, Cload=47pF, Rload=1kΩ. Vin=25mVpp and Vout DC at half AVDD. GBW2 is measured at Phase=-45deg.

.....continued

AC CHARACTERISTICS			Standard Operating Conditions: VDD = AVDD = 1.62V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ T_A ≤ +85°C for Industrial				
Param. No.	**Symbol**	**Characteristics**	**Min.**	**Typ.**	**Max.**	**Units**	**Conditions**
OA_21	AV2	Unity Gain	—	1	—	V/V	Unity gain follower Config: OPAMPCTRLx.BIAS=3, POTMUX=0x0, Cload=47pF, Rload=1kΩ. Fin=50kHz, Vin=25mVpp and Vout DC at half AVDD.
				1			Unity gain follower Config: OPAMPCTRLx.BIAS=2, POTMUX=0x0, Cload=47pF, Rload=1kΩ. Fin=50kHz, Vin=25mVpp and Vout DC at half AVDD.
				1			Unity gain follower Config: OPAMPCTRLx.BIAS=1, POTMUX=0x0, Cload=47pF, Rload=1kΩ. Fin=50kHz, Vin=25mVpp and Vout DC at half AVDD.
				1			Unity gain follower Config: OPAMPCTRLx.BIAS=0, POTMUX=0x0, Cload=47pF, Rload=1kΩ. Fin=50kHz, Vin=25mVpp and Vout DC at half AVDD.

Notes:

1. AVDD = 3.3V.
2. Parameters depend on the opamp configuration and must fulfill OA_8, OA_9, OA_15, OA_16, OA_17, OA_19 and OA_21 parameters specification.

49.24 AC Electrical Specifications

Table 49-31. Analog Comparator Electrical Specifications (1)

AC CHARACTERISTICS			Standard Operating Conditions: VDD = AVDD = 1.62V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ T_A ≤ +85°C for Industrial				
Param. No.	Symbol	Characteristics	Min.	Typ.	Max.	Units	Conditions
CMP_1	VIOFF	Input Offset Voltage	-70	—	70	mV	COMPCTRLn.SPEED = 0x0
			-55	—	55		COMPCTRLn.SPEED = 0x1
			-48	—	48		COMPCTRLn.SPEED = 0x2
			-42	—	42		COMPCTRLn.SPEED = 0x3
CMP_4	VIN	Input Voltage Range	AVSS	—	AVDD	V	With respect to AVSS and AVDD
CMP_5	VHYST	Input Hysteresis Voltage	10	—	74	mV	COMPCTRLn.HYSTEN = 0x1 COMPCTRLn.HYST = 0x0
			22	—	106		COMPCTRLn.HYSTEN = 0x1 COMPCTRLn.HYST = 0x1
			37	—	129		COMPCTRLn.HYSTEN = 0x1 COMPCTRLn.HYST = 0x2
			49	—	150		COMPCTRLn.HYSTEN = 0x1 COMPCTRLn.HYST = 0x3
CMP_15	$TRESP_{SS}$ (3)	Small Signal Response Time	—	—	12.3	µs	Comparator ref voltage=AVDD/2, COMPCTRLn.HYSTEN = 0x0, COMPCTRLn.SPEED = 0x0, Input overdrive = ± 100mV
				—	2.6		Comparator ref voltage=AVDD/2, COMPCTRLn.HYSTEN = 0x0, COMPCTRLn.SPEED = 0x1, Input overdrive = ± 100mV
				—	1.4		Comparator ref voltage=AVDD/2, COMPCTRLn.HYSTEN = 0x0, COMPCTRLn.SPEED = 0x2, Input overdrive = ± 100mV
				—	0.77		Comparator ref voltage=AVDD/2, COMPCTRLn.HYSTEN = 0x0, COMPCTRLn.SPEED = 0x3, Input overdrive = ± 100mV
CMP_19	COUTVAL	Comparator Enabled to Output Valid	—	—	71	µs	COMPCTRLn.SPEED = 0x0
				—	4.5		COMPCTRLn.SPEED = 0x1
				—	3.2		COMPCTRLn.SPEED = 0x2
				—	2.7		COMPCTRLn.SPEED = 0x3
CMP_21	AC_{IREF}	Comparator Internal Band Gap Voltage Reference	VR_9, VR_25, VR_27			V	VR_9, VR_25, VR_27: Refer to Internal Voltage Reference Specifications.

.....continued

AC CHARACTERISTICS			Standard Operating Conditions: VDD = AVDD = 1.62V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ T_A ≤ +85°C for Industrial				
Param. No.	**Symbol**	**Characteristics**	**Min.**	**Typ.**	**Max.**	**Units**	**Conditions**
CMP_23	$C_{VREFRNG}$	Comparator Voltage Reference Input Range	Note (2)	—	AVDD-0.1 or Note (2) whichever is lower	V	—
CMP_25	f_{GCLK_AC}	Comp digital input Clk Freq	—	—	FCLK_49	MHz	FCLK_49: Refer to Maximum Clock Frequencies Electrical Specifications

Notes:

1. Typical values at 25°C only.
2. Comparator Ref voltage cannot exceed (VIN(max) - VIOFF(max) - CMP_5(max) - 50mV) ≥ $C_{VREFRNG}$ ≥ (VIN(min) + abs(VIOFF(min)) + (CMP_5(max) + 50mV)) .
3. $TRESP_{SS\_XX}$ is measured from VIN transition to CMPx output toggle. It takes into account only analog propagation delay.

49.25 PTC Electrical Specifications

Table 49-32. Peripheral Touch Controller Electrical Specifications

DC CHARACTERISTICS				Standard Operating Conditions: VDD = AVDD = 1.62V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ T_A ≤ +85°C for Industrial				
Param. No.	**Symbol**	**Characteristics**		**Min.**	**Typ.**	**Max.**	**Units**	**Conditions**
PTC_1	$C_{LOAD\ SC}$ (1)	Self Capacitance Mode (PTC Channel XY0 - XY31)		—	—	45	pF	Maximum sensor load capacitance
PTC_3	$C_{LOAD\ MC}$ (1)	Mutual Capacitance Mode (PTC Channel XY0 - XY31)		—	—	31	pF	
PTC ANALOG GAIN SETTINGS (2)								
PTC_5	PTC_{GAIN}	Analog Gain Settings	Gain_1	—	1.0	—	—	—
			Gain_2	—	2.0	—		
			Gain_4	—	3.9	—		
			Gain_8	—	8.1	—		

Notes:

1. Maximum capacitive load that the PTC circuitry can compensate for each channel.
2. Analog Gain is a parameter of the Touch Library. Refer to the “Touch Library Peripheral Touch Controller User Guide”.

49.26 SPI Electrical Specifications (PL0)

Figure 49-5. SERCOMx SPI Host Module CPHA = 0 Timing Diagrams

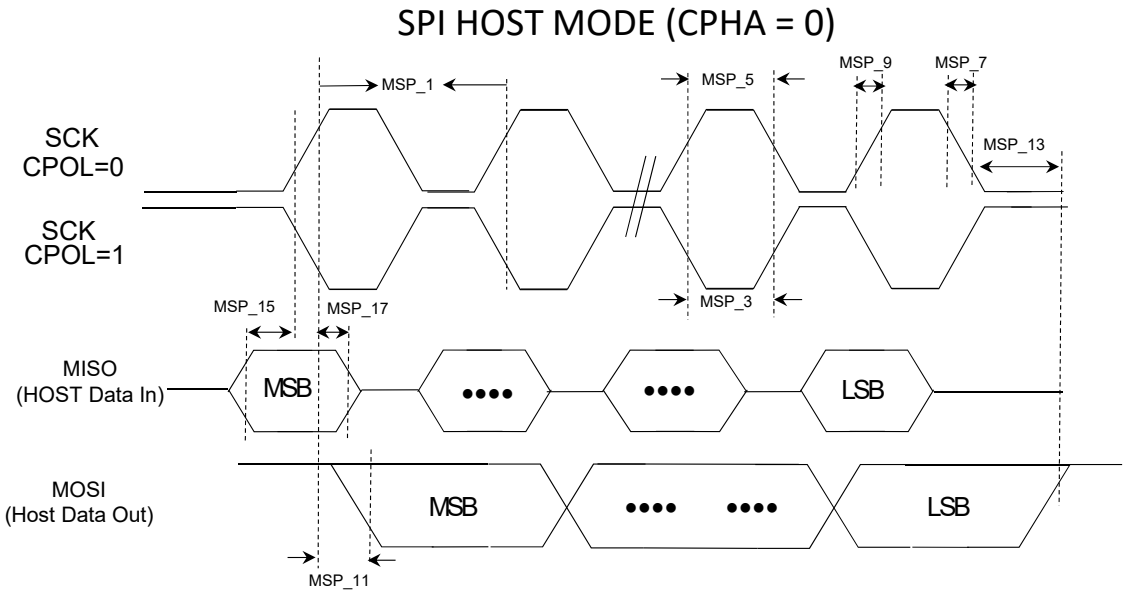


Figure 49-6. SERCOMx SPI Host Module CPHA = 1 Timing Diagrams

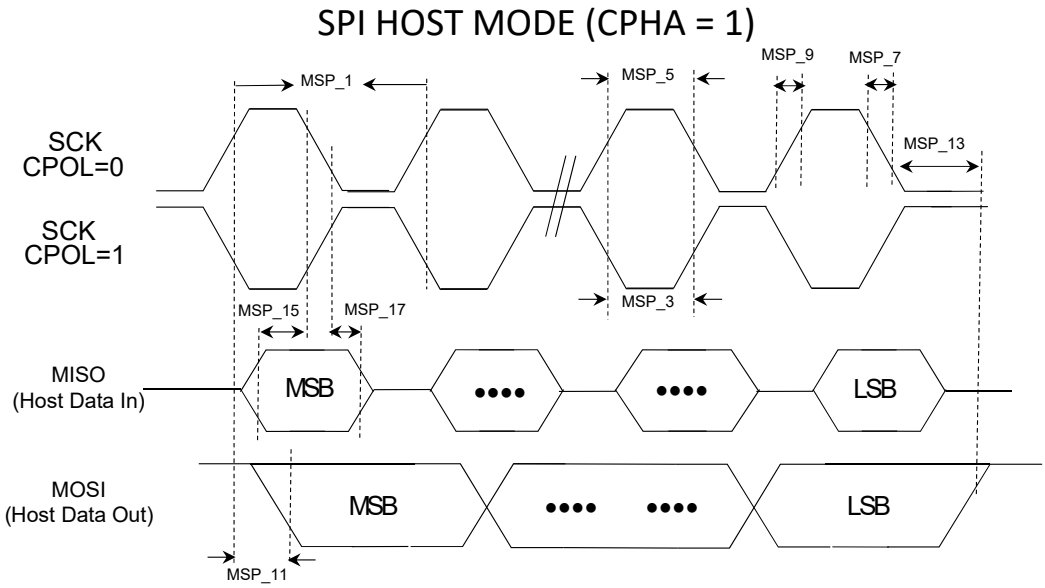


Table 49-33. SERCOMx SPI Module Host Mode Electrical Specifications (Performance Level 0 Mode) (1)

AC CHARACTERISTICS			Standard Operating Conditions: VDD = AVDD = 1.62V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ T_A ≤ +85°C for Industrial				
Param. No.	**Symbol**	**Characteristics**	**Min.**	**Typ.**	**Max.**	**Units**	**Conditions**
MSP_1	FSCK	SCK Frequency	—	—	3	MHz	VDD=1.8V, C_{LOAD}=30pF(MAX)
			—	—	3		VDD=3.3V, C_{LOAD}=30pF(MAX)
MSP_3	TSCL	SCK Output Low Time	1/(2*FSCK)	—	—	ns	—

.....continued

AC CHARACTERISTICS			Standard Operating Conditions: VDD = AVDD = 1.62V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ T_A ≤ +85°C for Industrial				
Param. No.	Symbol	Characteristics	Min.	Typ.	Max.	Units	Conditions
MSP_5	TSCH	SCK Output High Time	1/(2*FSCK)	—	—	ns	—
MSP_7	TSCF	SCK & MOSI Output Fall Time	—	—	DI_27	ns	DI_27: Refer to I/O Pin Electrical Specifications
MSP_9	TSCR	SCK & MOSI Output Rise Time	—	—	DI_25	ns	DI_25: Refer to I/O Pin Electrical Specifications
MSP_11	TMOV	MOSI Data Output Valid after SCK	—	—	39.3	ns	VDD=3.3V, C_{LOAD}=30pF(MAX)
MSP_13	TMOH	MOSI hold after SCK	0	—	—	ns	
MSP_15	TMIS	MISO Setup Time of Data Input to SCK	0	—	—	ns	
MSP_17	TMIH	MISO Hold Time of Data Input to SCK	153.2	—	—	ns	
MSP_19	SPI_GCLK	SERCOM SPI input clk freq, GCLK_SERCOMx_CORE	—	—	FCLK_23	MHz	FCLK_23: Refer to Maximum Clock Frequencies Electrical Specifications

Note:

1. SPI I/O pins configured with drive strength enabled (PORT.PINCFGn.DRVSTR=1).

Figure 49-7. SERCOMx SPI Client Module (CPHA = 0) Timing Diagram

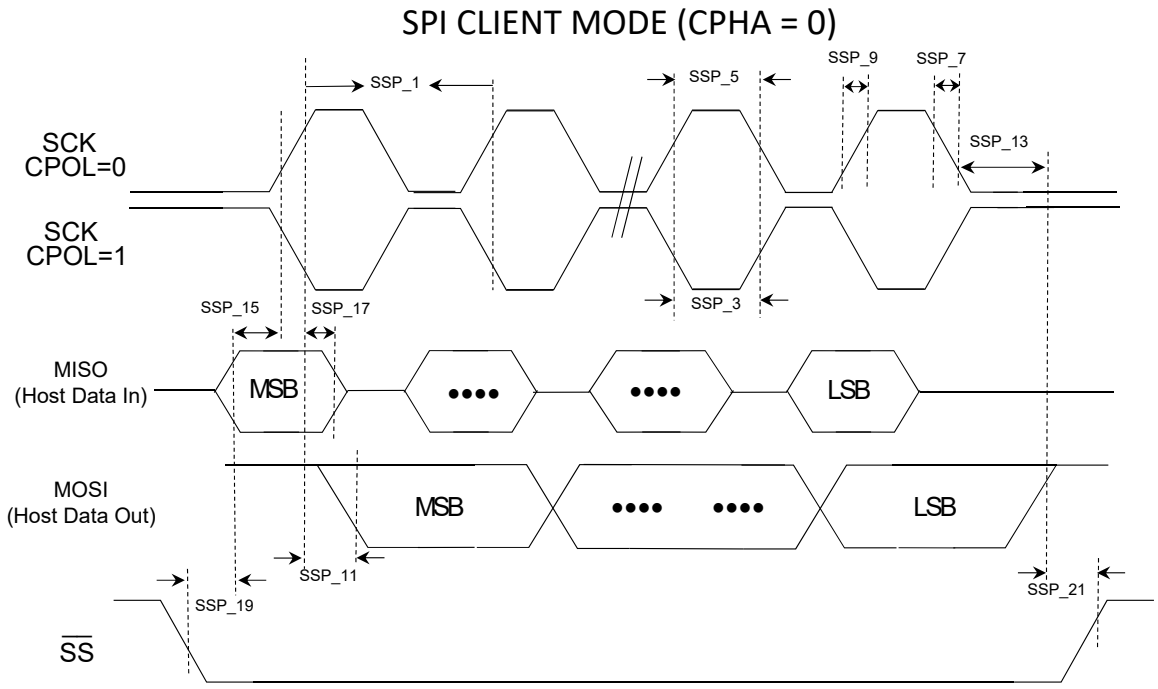


Figure 49-8. SERCOMx SPI Client Module (CPHA = 1) Timing Diagram

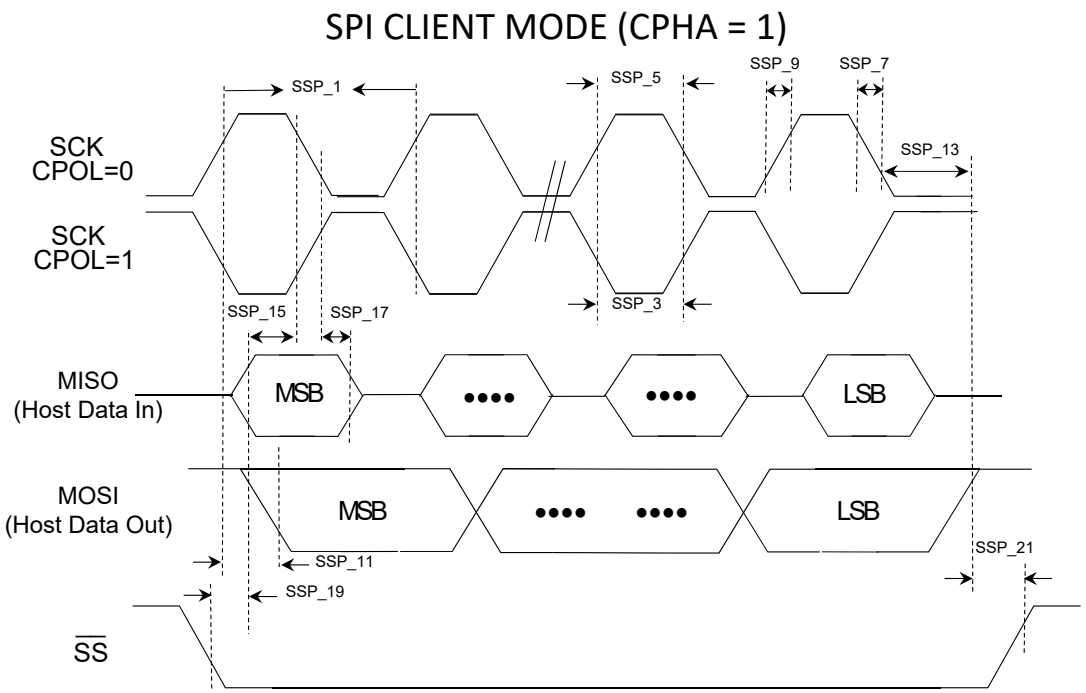


Table 49-34. SERCOMx SPI Module Client Mode Electrical Specifications (Performance Level 0 Mode) (1)

AC CHARACTERISTICS			Standard Operating Conditions: VDD = AVDD = 1.62V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ T_A ≤ +85°C for Industrial				
Param. No.	**Symbol**	**Characteristics**	**Min.**	**Typ.**	**Max.**	**Units**	**Conditions**
SSP_1	FSCK	SCK Frequency	—	—	3	MHz	VDD=1.8V, C_{LOAD}=30pF(MAX)
			—	—	3		VDD=3.3V, C_{LOAD}=30pF(MAX)
SSP_3	TSCL	SCK Output Low Time	1/(2*FSCK)	—	—	ns	—
SSP_5	TSCH	SCK Output High Time	1/(2*FSCK)	—	—	ns	—
SSP_7	TSCF	SCK & MOSI Output Fall Time	—	—	DI_27	ns	DI_27: Refer to I/O Pin Electrical Specifications
SSP_9	TSCR	SCK & MOSI Output Rise Time	—	—	DI_25	ns	DI_25: Refer to I/O Pin Electrical Specifications
SSP_11	TSOV	MOSI Data Output Valid after SCK	—	—	80.7	ns	VDD=3.3V, C_{LOAD}=30pF(MAX)
SSP_13	TSOH	MOSI hold after SCK	15.5	—	—	ns	
SSP_15	TSIS	MISO Setup Time of Data Input to SCK	40.5	—	—	ns	
SSP_17	TSIH	MISO Hold Time of Data Input to SCK	5.4	—	—	ns	
SSP_19	TSSS	$\overline{SS}$ setup to SCK (CTRLB.PLOADEN=1)	2/fAPBx+32	—	—	ns	
		$\overline{SS}$ setup to SCK (CTRLB.PLOADEN=0)	32	—	—		
SSP_21	TSSH	$\overline{SS}$ hold after SCK Client	166.7	—	—	ns	
SSP_23	$f_{GCLK\_SERCOMx\_CORE}$	SERCOM SPI input clk freq, GCLK_SPI	—	—	FCLK_23	MHz	FCLK_23: Refer to Maximum Clock Frequencies Electrical Specifications

.....continued

AC CHARACTERISTICS			Standard Operating Conditions: VDD = AVDD = 1.62V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ T_A ≤ +85°C for Industrial				
Param. No.	Symbol	Characteristics	Min.	Typ.	Max.	Units	Conditions

Note:

1. The SPI I/O pins configured with drive strength enabled (PORT.PINCFGn.DRVSTR = 1).

49.27 SPI Electrical Specifications (PL2)

Figure 49-9. SERCOMx SPI Host Module CPHA = 0 Timing Diagrams

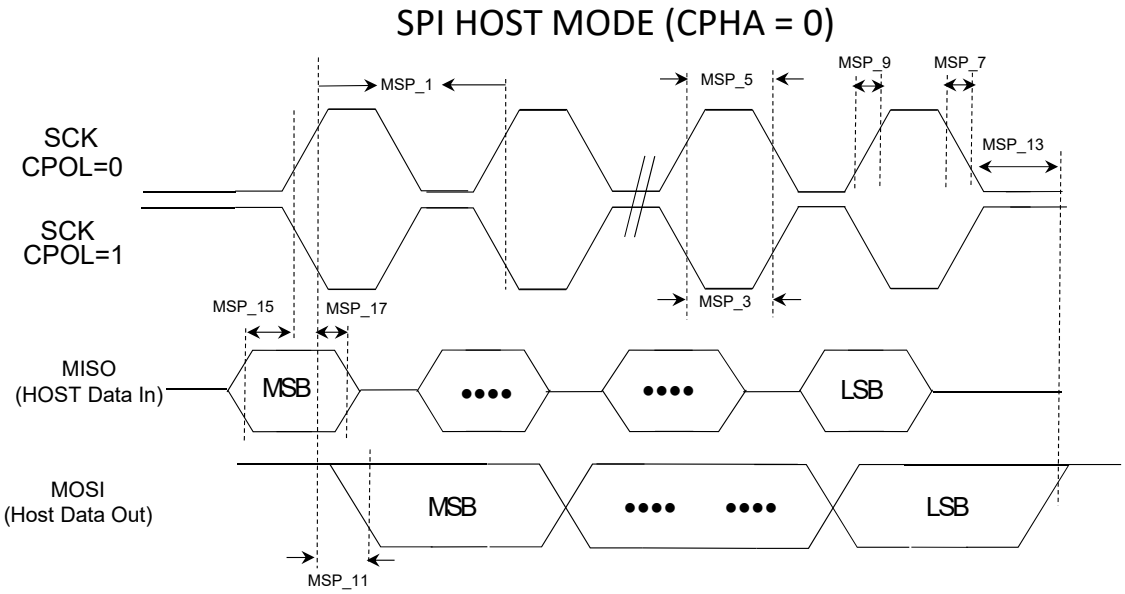


Figure 49-10. SERCOMx SPI Host Module CPHA = 1 Timing Diagrams

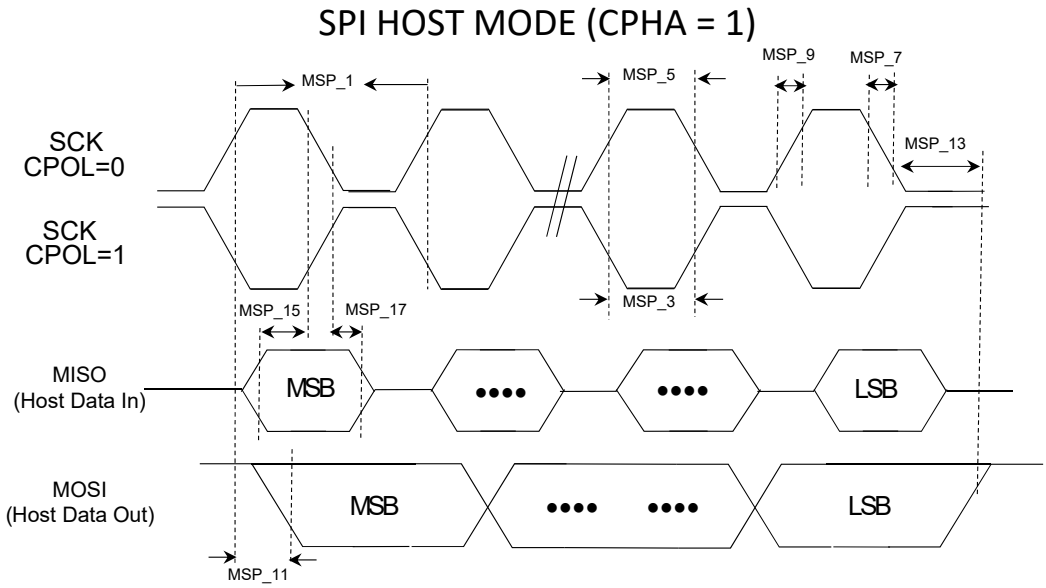


Table 49-35. SERCOMx SPI Module Host Mode Electrical Specifications (Performance Level 2 Mode) (1)

AC CHARACTERISTICS			Standard Operating Conditions: VDD = AVDD = 1.62V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ T_A ≤ +85°C for Industrial				
Param. No.	**Symbol**	**Characteristics**	**Min.**	**Typ.**	**Max.**	**Units**	**Conditions**
MSP_1	FSCK	SCK Frequency	—	—	12	MHz	VDD=1.8V, C_{LOAD}=30pF(MAX)
			—	—	12		VDD=3.3V, C_{LOAD}=30pF(MAX)
MSP_3	TSCL	SCK Output Low Time	1/(2*FSCK)	—	—	ns	—

.....continued

AC CHARACTERISTICS			Standard Operating Conditions: VDD = AVDD = 1.62V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ T_A ≤ +85°C for Industrial				
Param. No.	Symbol	Characteristics	Min.	Typ.	Max.	Units	Conditions
MSP_5	TSCH	SCK Output High Time	1/(2*FSCK)	—	—	ns	—
MSP_7	TSCF	SCK & MOSI Output Fall Time	—	—	DI_27	ns	DI_27: Refer to I/O Pin Electrical Specifications
MSP_9	TSCR	SCK & MOSI Output Rise Time	—	—	DI_25	ns	DI_25: Refer to I/O Pin Electrical Specifications
MSP_11	TMOV	MOSI Data Output Valid after SCK	—	—	17.2	ns	VDD=3.3V, C_{LOAD}=30pF(MAX)
MSP_13	TMOH	MOSI hold after SCK	0	—	—	ns	
MSP_15	TMIS	MISO Setup Time of Data Input to SCK	1.2	—	—	ns	
MSP_17	TMIH	MISO Hold Time of Data Input to SCK	32.6	—	—	ns	
MSP_19	SPI_GCLK	SERCOM SPI input clk freq, GCLK_SERCOMx_CORE	—	—	FCLK_23	MHz	FCLK_23: Refer to Maximum Clock Frequencies Electrical Specifications

Note:

1. The SPI I/O pins configured with drive strength enabled (PORT.PINCFGn.DRVSTR = 1).

Figure 49-11. SERCOMx SPI Client Module (CPHA = 0) Timing Diagrams

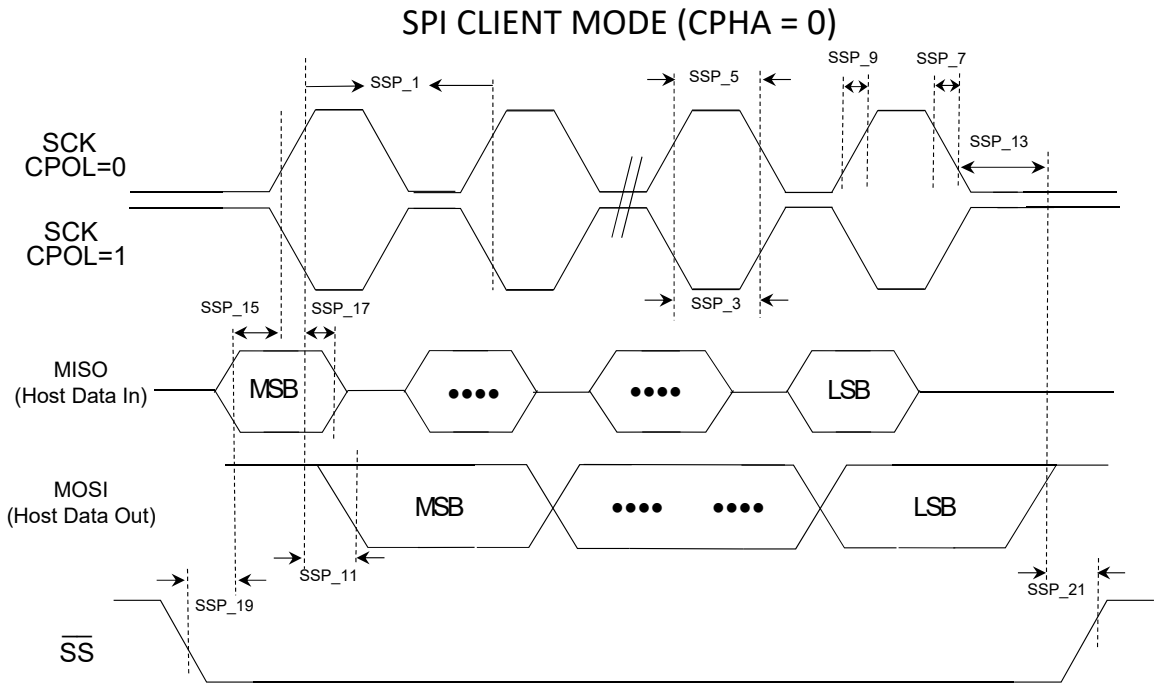


Figure 49-12. SERCOMx SPI Client Module (CPHA = 1) Timing Diagrams

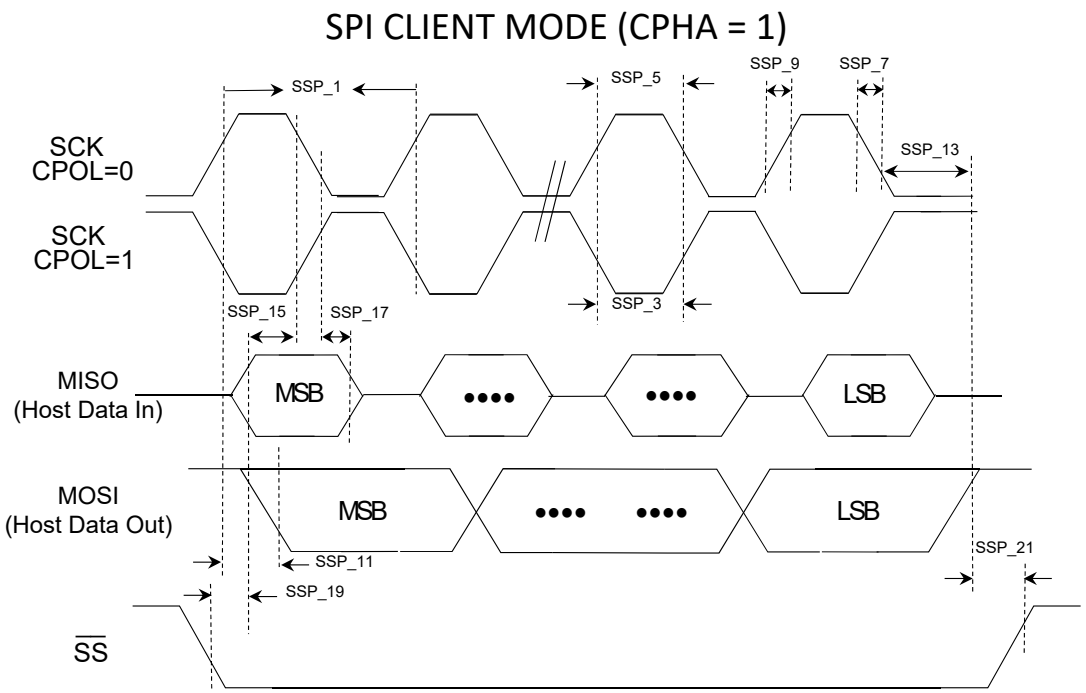


Table 49-36. SERCOMx SPI Module Client Mode Electrical Specifications (Performance Level 2 Mode) (1)

AC CHARACTERISTICS			Standard Operating Conditions: VDD = AVDD = 1.62V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ T_A ≤ +85°C for Industrial				
Param. No.	**Symbol**	**Characteristics**	**Min.**	**Typ.**	**Max.**	**Units**	**Conditions**
SSP_1	FSCK	SCK Frequency	—	—	12	MHz	VDD=1.8V, C_{LOAD}=30pF(MAX)
			—	—	12		VDD=3.3V, C_{LOAD}=30pF(MAX)
SSP_3	TSCL	SCK Output Low Time	1/(2*FSCK)	—	—	ns	—
SSP_5	TSCH	SCK Output High Time	1/(2*FSCK)	—	—	ns	—
SSP_7	TSCF	SCK & MOSI Output Fall Time	—	—	DI_27	ns	DI_27: Refer to I/O Pin Electrical Specifications
SSP_9	TSCR	SCK & MOSI Output Rise Time	—	—	DI_25	ns	DI_25: Refer to I/O Pin Electrical Specifications
SSP_11	TSOV	MOSI Data Output Valid after SCK	—	—	40.1	ns	VDD=3.3V, C_{LOAD}=30pF(MAX)
SSP_13	TSOH	MOSI hold after SCK	10.4	—	—	ns	
SSP_15	TSIS	MISO Setup Time of Data Input to SCK	16.5	—	—	ns	
SSP_17	TSIH	MISO Hold Time of Data Input to SCK	3	—	—	ns	
SSP_19	TSSS	$\overline{SS}$ setup to SCK (CTRLB.PLOADEN=1)	2/fAPBx+13,9	—	—	ns	
		$\overline{SS}$ setup to SCK (CTRLB.PLOADEN=0)	13.9	—	—		
SSP_21	TSSH	$\overline{SS}$ hold after SCK Client	41.7	—	—	ns	
SSP_23	$f_{GCLK\_SERCOMx\_CORE}$	SERCOM SPI input clk freq, GCLK_SPI	—	—	FCLK_23	MHz	FCLK_23: Refer to Maximum Clock Frequencies Electrical Specifications

.....continued

AC CHARACTERISTICS			Standard Operating Conditions: VDD = AVDD = 1.62V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ T_A ≤ +85°C for Industrial				
Param. No.	Symbol	Characteristics	Min.	Typ.	Max.	Units	Conditions

Note:

1. The SPI I/O pins configured with drive strength enabled (PORT.PINCFGn.DRVSTR = 1).

49.28 USART Electrical Specifications (PL0)

Table 49-37. USART Electrical Specifications (Performance Level 0 Mode)

AC CHARACTERISTICS				Standard Operating Conditions: VDD = AVDD = 1.62V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ T_A ≤ +85°C for Industrial				
Param. No.	Symbol	Characteristics		Min.	Typ.	Max.	Units	Conditions
UT_1	F_{BRATE}	Baud Rate	Asynchronous SAMPR = 16x mode	—	—	0.375	Mbps	—
UT_3			Asynchronous SAMPR = 8x mode	—	—	0.75	Mbps	—
UT_5			Asynchronous SAMPR = 3x mode	—	—	2	Mbps	—
UT_7			Synchronous Mode	—	—	1.5	Mbps	—
UT_9	F_{USART}	USART max GCLK_SERCOMx_CORE		—	—	FCLK_23	MHz	FCLK_23: Refer to Maximum Clock Frequencies Electrical Specifications
UT_11	F_{XCK}	USART External Clock Input		—	—	FCLK_23	MHz	—

Note:

1. These parameters are characterized, but not tested in manufacturing.

49.29 USART Electrical Specifications (PL2)

Table 49-38. USART Electrical Specifications (Performance Level 2 Mode)

AC CHARACTERISTICS				Standard Operating Conditions: VDD = AVDD = 1.62V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ T_A ≤ +85°C for Industrial				
Param. No.	Symbol	Characteristics		Min.	Typ.	Max.	Units	Conditions
UT_21	F_{BRATE}	Baud Rate	Asynchronous SAMPR = 16x mode	—	—	1.5	Mbps	—
UT_23			Asynchronous SAMPR = 8x mode	—	—	3	Mbps	—
UT_25			Asynchronous SAMPR = 3x mode	—	—	8	Mbps	—
UT_27			Synchronous Mode	—	—	6	Mbps	—
UT_29	F_{USART}	USART max GCLK_SERCOMx_CORE		—	—	FCLK_23	MHz	FCLK_23: Refer to Maximum Clock Frequencies Electrical Specifications

.....continued

AC CHARACTERISTICS			Standard Operating Conditions: VDD = AVDD = 1.62V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ T_A ≤ +85°C for Industrial				
Param. No.	Symbol	Characteristics	Min.	Typ.	Max.	Units	Conditions
UT_31	F_{XCK}	USART External Clock Input	—	—	FCLK_23	MHz	—

Note:

1. These parameters are characterized, but not tested in manufacturing.

49.30 I²S Electrical Specifications (PL2)

Figure 49-13. I²S Host Mode Timing Diagram

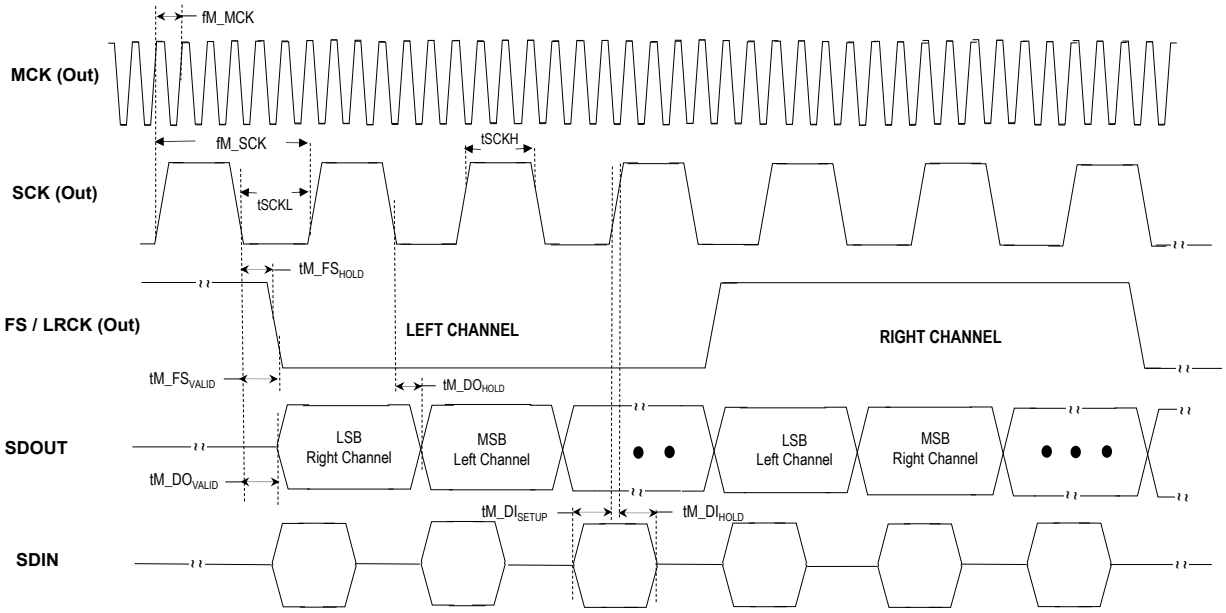


Table 49-39. I²S Host Mode Electrical Specifications (Performance Level 2 Mode) (1)

AC CHARACTERISTICS			Standard Operating Conditions: VDD = AVDD = 1.62V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ T_A ≤ +85°C for Industrial				
Param. No.	Symbol	Characteristics	Min.	Typ.	Max.	Units	Conditions
I2S_1	fM_MCK	Host MCK Frequency	—	—	19.4	MHz	VDD=1.8V, C_{LOAD}=30pF(MAX)
			—	—	27		VDD=3.3V, C_{LOAD}=30pF(MAX)
I2S_3	tMCKL	MCK Output Low Time	1/(2*fM_SCK)	—	—	ns	—
I2S_5	tMCKH	MCK Output High Time	1/(2*fM_SCK)	—	—	ns	—
I2S_7	tMCKR	MCK Rise Time	DI_25			ns	DI_25: Refer to I/O Pin Electrical Specifications
I2S_9	tMCKF	MCK Fall Time	DI_27			ns	DI_27: Refer to I/O Pin Electrical Specifications
I2S_11	fM_SCK	Host SCK Frequency	—	—	7	MHz	VDD=1.8V, C_{LOAD}=30pF(MAX)
			—	—	7		VDD=3.3V, C_{LOAD}=30pF(MAX)
I2S_13	tSCKL	SCK Output Low Time	1/(2*fM_SCK)	—	—	ns	—
I2S_15	tSCKH	SCK Output High Time	1/(2*fM_SCK)	—	—	ns	—
I2S_17	tSCKR	SCK Rise Time	DI_25			ns	DI_25: Refer to I/O Pin Electrical Specifications

.....continued

AC CHARACTERISTICS			Standard Operating Conditions: VDD = AVDD = 1.62V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ T_A ≤ +85°C for Industrial				
Param. No.	Symbol	Characteristics	Min.	Typ.	Max.	Units	Conditions
I2S_19	tSCKF	SCK Fall Time	DI_27			ns	DI_27: Refer to I/O Pin Electrical Specifications
I2S_21	tM_FS$_{VALID}$	Host Frame Sync Valid	—	—	11.1	ns	VDD=1.8V, C_{LOAD}=30pF(MAX)
			—	—	11.2		VDD=3.3V, C_{LOAD}=30pF(MAX)
I2S_23	tM_FS$_{HOLD}$	Host Frame Sync Hold	—	—	0	ns	VDD=1.8V, C_{LOAD}=30pF(MAX)
			—	—	0		VDD=3.3V, C_{LOAD}=30pF(MAX)
I2S_25	tM_DI$_{SETUP}$	Host Data Input Setup	38.1	—	—	ns	VDD=1.8V, C_{LOAD}=30pF(MAX)
			27.5	—	—		VDD=3.3V, C_{LOAD}=30pF(MAX)
I2S_27	tM_DI$_{HOLD}$	Host Data Input Hold	0	—	—	ns	VDD=1.8V, C_{LOAD}=30pF(MAX)
			0	—	—		VDD=3.3V, C_{LOAD}=30pF(MAX)
I2S_29	tM_DO$_{VALID}$	Host Data Output Valid	—	—	7.1	ns	VDD=1.8V, C_{LOAD}=30pF(MAX)
			—	—	6.8		VDD=3.3V, C_{LOAD}=30pF(MAX)
I2S_31	tM_DO$_{HOLD}$	Host Data Output Hold	0.7	—	—	ns	VDD=1.8V, C_{LOAD}=30pF(MAX)
			0	—	—		VDD=3.3V, C_{LOAD}=30pF(MAX)
I2S_33	fGCLK_I2S	I^2S Max GLK Input Clock Freq	—	—	FCLK_29	MHz	FCLK_29: Refer to Maximum Clock Frequencies Electrical Specifications

Note:

1. I²S I/O pins configured with drive strength enabled (PORT.PINCFGn.DRVSTR=1).

Figure 49-14. I^2S Client Mode Timing Diagram

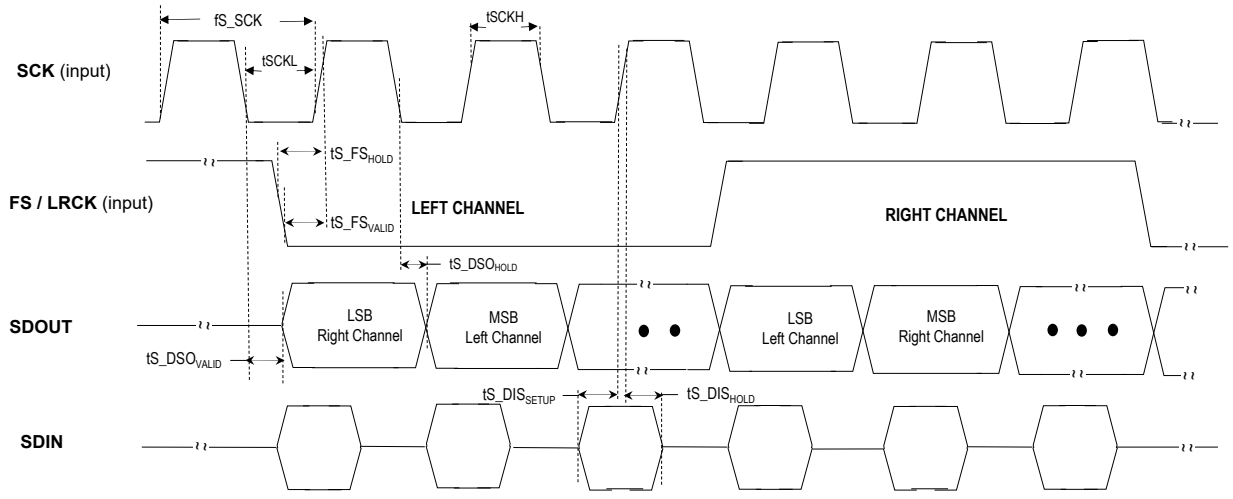


Table 49-40. I²S Client Mode Electrical Specifications (Performance Level 2 Mode) (1)

AC CHARACTERISTICS			Standard Operating Conditions: VDD = AVDD = 1.62V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ T_A ≤ +85°C for Industrial				
Param. No.	Symbol	Characteristics	Min.	Typ.	Max.	Units	Conditions
I2S_41	fS_SCK	SCK Client Frequency	—	—	12.5	MHz	VDD=1.8V, C_{LOAD}=30pF(MAX)
			—	—	12.5		VDD=3.3V, C_{LOAD}=30pF(MAX)
I2S_43	tS_FS_{VALID}	Frame Sync Valid	7.5	—	—	ns	VDD=1.8V, C_{LOAD}=30pF(MAX)
			7	—	—		VDD=3.3V, C_{LOAD}=30pF(MAX)
I2S_45	tS_FS_{HOLD}	Frame Sync Hold	0	—	—	ns	VDD=1.8V, C_{LOAD}=30pF(MAX)
			0.4	—	—		VDD=3.3V, C_{LOAD}=30pF(MAX)
I2S_47	tS_DIS_{SETUP}	Data Input Client Setup	6.3	—	—	ns	VDD=1.8V, C_{LOAD}=30pF(MAX)
			5.7	—	—		VDD=3.3V, C_{LOAD}=30pF(MAX)
I2S_49	tS_DIS_{HOLD}	Data Input Client Hold	0.6	—	—	ns	VDD=1.8V, C_{LOAD}=30pF(MAX)
			1.2	—	—		VDD=3.3V, C_{LOAD}=30pF(MAX)
I2S_51	tS_DSO_{VALID}	Data Output Client Valid	—	—	38.6	ns	VDD=1.8V, C_{LOAD}=30pF(MAX)
			—	—	27.3		VDD=3.3V, C_{LOAD}=30pF(MAX)
I2S_53	tS_DSO_{HOLD}	Data Output Client Hold	33.6	—	—	ns	VDD=1.8V, C_{LOAD}=30pF(MAX)
			10.1	—	—		VDD=3.3V, C_{LOAD}=30pF(MAX)

Note:

1. I²S I/O pins configured with drive strength enabled (PORT.PINCFGn.DRVSTR=1).

Figure 49-15. I²S PDM2 Mode Timing Diagram

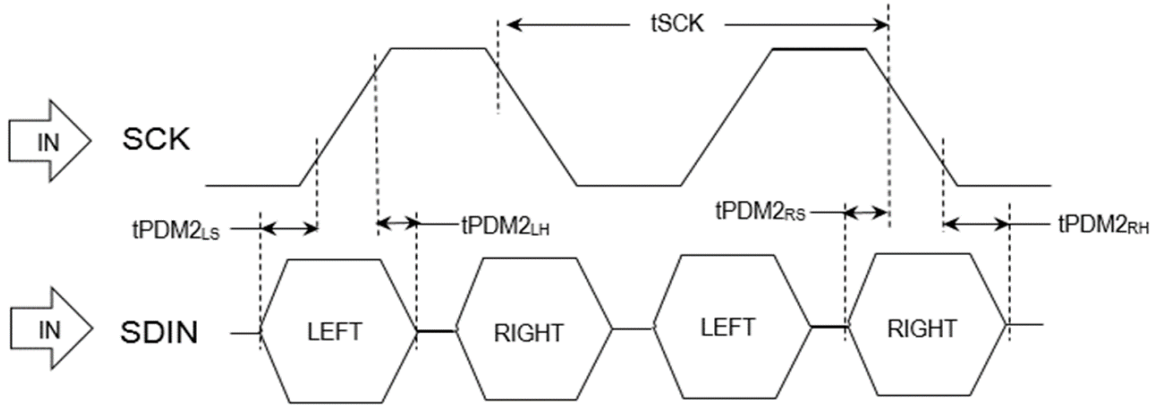


Table 49-41. I²S PDM2 Mode Electrical Specifications (Performance Level 2 Mode) (1)

AC CHARACTERISTICS			Standard Operating Conditions: VDD = AVDD = 1.62V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ T_A ≤ +85°C for Industrial				
Param. No.	Symbol	Characteristics	Min.	Typ.	Max.	Units	Conditions
I2S_55	tSCK	SCK Frequency	—	—	7	MHz	VDD=1.8V, C_{LOAD}=30pF(MAX)
			—	—	7		VDD=3.3V, C_{LOAD}=30pF(MAX)
I2S_57	$tPDM2_{LS}$	PDM Left Channel Setup	39.1	—	—	ns	VDD=1.8V, C_{LOAD}=30pF(MAX)
			28	—	—		VDD=3.3V, C_{LOAD}=30pF(MAX)

.....continued

AC CHARACTERISTICS			Standard Operating Conditions: VDD = AVDD = 1.62V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ T_A ≤ +85°C for Industrial				
Param. No.	Symbol	Characteristics	Min.	Typ.	Max.	Units	Conditions
I2S_59	$tPDM2_{LH}$	PDM Left Channel Hold	0	—	—	ns	VDD=1.8V, C_{LOAD}=30pF(MAX)
			0	—	—		VDD=3.3V, C_{LOAD}=30pF(MAX)
I2S_61	$tPDM2_{RS}$	PDM Right Channel Setup	42.5	—	—	ns	VDD=1.8V, C_{LOAD}=30pF(MAX)
			32.1	—	—		VDD=3.3V, C_{LOAD}=30pF(MAX)
I2S_63	$tPDM2_{RH}$	PDM Right Channel Hold	0	—	—	ns	VDD=1.8V, C_{LOAD}=30pF(MAX)
			0	—	—		VDD=3.3V, C_{LOAD}=30pF(MAX)

Note:

1. I²S I/O pins configured with drive strength enabled (PORT.PINCFGn.DRVSTR=1).

49.31 I²C Electrical Specifications

Figure 49-16. I²C Start/Stop Bits Host Mode Timing Diagrams

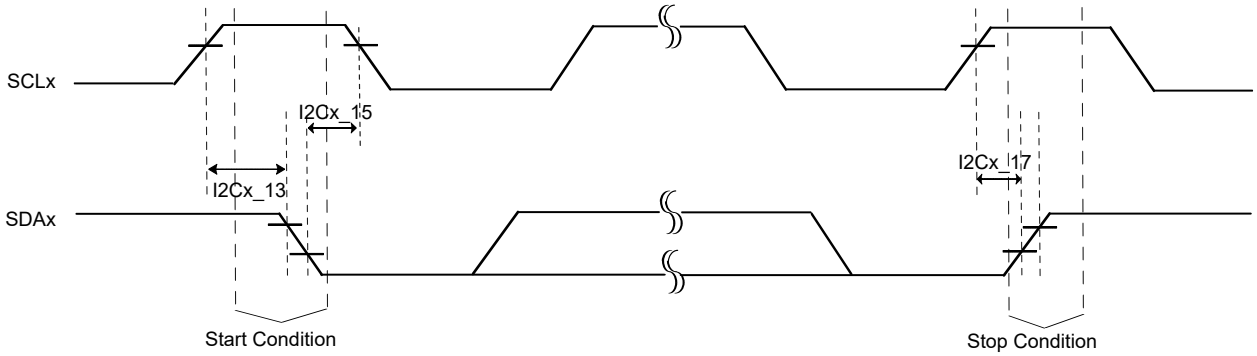


Figure 49-17. I²C Bus Data Host Mode Timing Diagrams

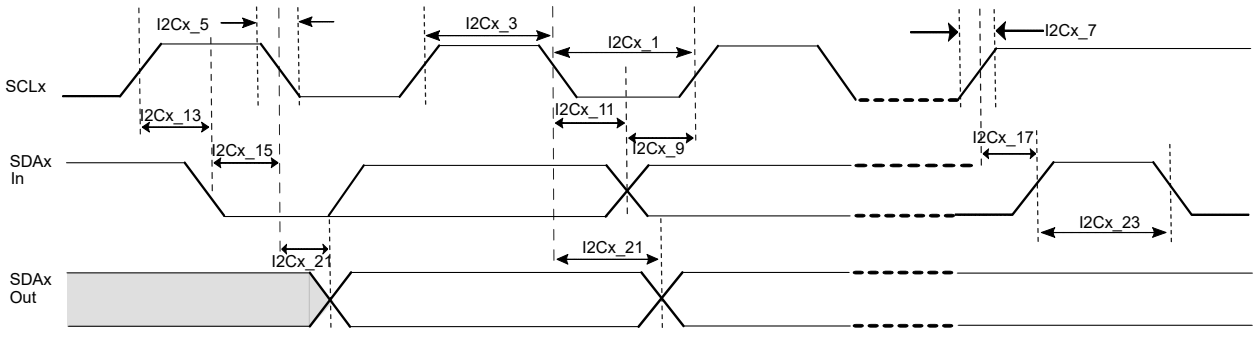


Table 49-42. I²C Host Mode Electrical Specifications

AC CHARACTERISTICS				Standard Operating Conditions: VDD = AVDD = 1.62V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ T_A ≤ +85°C for Industrial			
Param. No.	**Symbol**	**Characteristics**		**Min.**	**Max.**	**Units**	**Conditions**
I2CM_1	TLO:SCL	Host Clock Low Time	100 kHz mode	4.7	—	µs	VDD=3.3V, I_{PULLUP} = 3mA, C_{LOAD}=400pF
			400 kHz mode	1.3	—	µs	
			1 MHz mode	0.5	—	µs	VDD=3.3V, I_{PULLUP} = 20mA, C_{LOAD}=550pF
			3.4 MHz mode	160	—	ns	VDD=3.3V, I_{PULLUP} = 20mA, C_{LOAD}=100pF
I2CM_3	THI:SCL	Host Clock High Time	100 kHz mode	4	—	µs	VDD=3.3V, I_{PULLUP} = 3mA, C_{LOAD}=400pF
			400 kHz mode	0.6	—	µs	
			1 MHz mode	0.26	—	µs	VDD=3.3V, I_{PULLUP} = 20mA, C_{LOAD}=550pF
			3.4 MHz mode	60	—	ns	VDD=3.3V, I_{PULLUP} = 20mA, C_{LOAD}=100pF
I2CM_5	TF:SCL	SDAx and SCLx Fall Time	100 kHz mode	—	300	ns	VDD=3.3V, I_{PULLUP} = 3mA, C_{LOAD}=400pF
			400 kHz mode	—	300	ns	
			1 MHz mode	—	120	ns	VDD=3.3V, I_{PULLUP} = 20mA, C_{LOAD}=550pF
			3.4 MHz mode	—	40	ns	VDD=3.3V, I_{PULLUP} = 20mA, C_{LOAD}=100pF
I2CM_7	TR:SCL	SDAx and SCLx Rise Time	100 kHz mode	—	1000	ns	VDD=3.3V, I_{PULLUP} = 3mA, C_{LOAD}=400pF
			400 kHz mode	—	300	ns	
			1 MHz mode	—	120	ns	VDD=3.3V, I_{PULLUP} = 20mA, C_{LOAD}=550pF
			3.4 MHz mode	—	40	ns	VDD=3.3V, I_{PULLUP} = 20mA, C_{LOAD}=100pF
I2CM_9	TSU:DAT	Data Input Setup Time (1)	100 kHz mode	250	—	ns	VDD=3.3V, I_{PULLUP} = 3mA, C_{LOAD}=400pF
			400 kHz mode	100	—	ns	
			1 MHz mode	50	—	ns	VDD=3.3V, I_{PULLUP} = 20mA, C_{LOAD}=550pF
			3.4 MHz mode	10	—	ns	VDD=3.3V, I_{PULLUP} = 20mA, C_{LOAD}=100pF
I2CM_11	THD:DAT	Data Input Hold Time	100 kHz mode	300	—	µs	VDD=3.3V, I_{PULLUP} = 3mA, C_{LOAD}=400pF
			400 kHz mode	300	—	µs	
			1 MHz mode	300	—	µs	VDD=3.3V, I_{PULLUP} = 20mA, C_{LOAD}=550pF
			3.4 MHz mode	5	—	ns	VDD=3.3V, I_{PULLUP} = 20mA, C_{LOAD}=100pF

.....continued

AC CHARACTERISTICS				Standard Operating Conditions: VDD = AVDD = 1.62V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ T_A ≤ +85°C for Industrial			
Param. No.	**Symbol**	**Characteristics**		**Min.**	**Max.**	**Units**	**Conditions**
I2CM_13	TSU:STA	Start Condition Setup Time	100 kHz mode	4.7	—	µs	VDD=3.3V, I_{PULLUP} = 3mA, C_{LOAD}=400pF
			400 kHz mode	0.6	—	µs	
			1 MHz mode	0.26	—	µs	VDD=3.3V, I_{PULLUP} = 20mA, C_{LOAD}=550pF
			3.4 MHz mode	160	—	ns	VDD=3.3V, I_{PULLUP} = 20mA, C_{LOAD}=100pF
I2CM_15	THD:STA	Start Condition Hold Time	100 kHz mode	4.0	—	µs	VDD=3.3V, I_{PULLUP} = 3mA, C_{LOAD}=400pF
			400 kHz mode	0.6	—	µs	
			1 MHz mode	0.26	—	µs	VDD=3.3V, I_{PULLUP} = 20mA, C_{LOAD}=550pF
			3.4 MHz mode	160	—	ns	VDD=3.3V, I_{PULLUP} = 20mA, C_{LOAD}=100pF
I2CM_17	TSU:STO	Stop Condition Setup Time	100 kHz mode	4.0	—	µs	VDD=3.3V, I_{PULLUP} = 3mA, C_{LOAD}=400pF
			400 kHz mode	0.6	—	µs	
			1 MHz mode	0.26	—	µs	VDD=3.3V, I_{PULLUP} = 20mA, C_{LOAD}=550pF
			3.4 MHz mode	160	—	ns	VDD=3.3V, I_{PULLUP} = 20mA, C_{LOAD}=100pF
I2CM_21	TAA:SCL	Output Valid from Clock	100 kHz mode	—	3.45	µs	VDD=3.3V, I_{PULLUP} = 3mA, C_{LOAD}=400pF
			400 kHz mode	—	0.9	µs	
			1 MHz mode	—	0.45	µs	VDD=3.3V, I_{PULLUP} = 20mA, C_{LOAD}=550pF
			3.4 MHz mode	—	0.1	µs	VDD=3.3V, I_{PULLUP} = 20mA, C_{LOAD}=100pF
I2CM_23	TBF:SDA	Bus Free Time (2)	100 kHz mode	4.7	—	µs	VDD=3.3V, I_{PULLUP} = 3mA, C_{LOAD}=400pF
			400 kHz mode	1.3	—	µs	
			1 MHz mode	0.5	—	µs	VDD=3.3V, I_{PULLUP} = 20mA, C_{LOAD}=550pF
			3.4 MHz mode	160	—	ns	VDD=3.3V, I_{PULLUP} = 20mA, C_{LOAD}=100pF

Notes:

1. Longest delay between data hold timing based on bitfield SDAHOLD of register CTRLA from SERCOM Module and timing based on 4 period of GCLK_SERCOM for 100kHz/400kHz/1MHz mode.
2. The amount of time the bus must be free before a new transmission can start (STOP condition to START condition).

Figure 49-18. I²C Start/Stop Bits Client Mode Timing Diagram

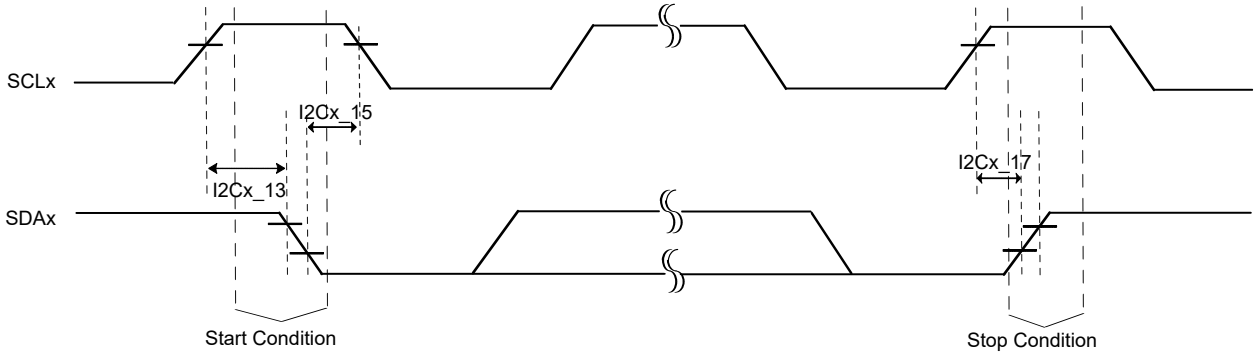


Figure 49-19. I²C Bus Data Client Mode Timing Diagrams

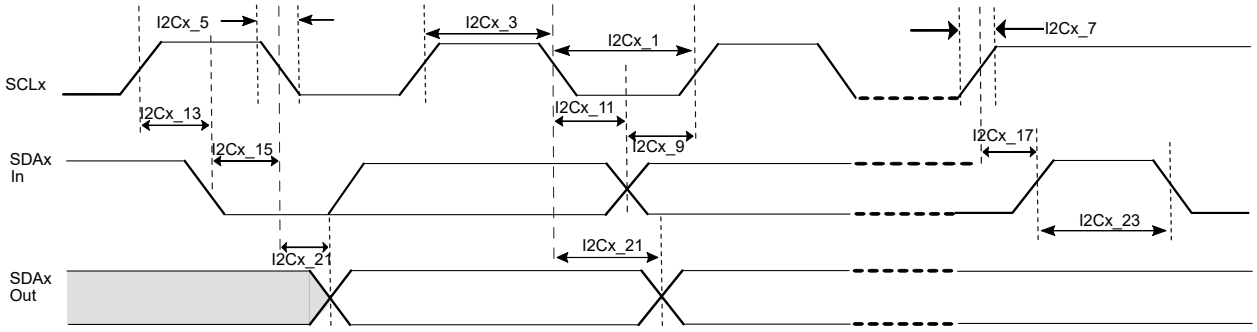


Table 49-43. I²C Client Mode Electrical Specifications

AC CHARACTERISTICS				Standard Operating Conditions: VDD = AVDD = 1.62V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ T_A ≤ +85°C for Industrial			
Param. No.	**Symbol**	**Characteristics**		**Min.**	**Max.**	**Units**	**Conditions**
I2CS_1	TLO:SCL	Client Clock Low Time	100 kHz mode	4.7	—	µs	VDD=3.3V, I_{PULLUP} = 3mA, C_{LOAD}=400pF
			400 kHz mode	1.3	—	µs	
			1 MHz mode	0.5	—	µs	VDD=3.3V, I_{PULLUP} = 20mA, C_{LOAD}=550pF
			3.4 MHz mode	160	—	ns	VDD=3.3V, I_{PULLUP} = 20mA, C_{LOAD}=100pF
I2CS_3	THI:SCL	Client Clock High Time	100 kHz mode	4.0	—	µs	VDD=3.3V, I_{PULLUP} = 3mA, C_{LOAD}=400pF
			400 kHz mode	0.6	—	µs	
			1 MHz mode	0.26	—	µs	VDD=3.3V, I_{PULLUP} = 20mA, C_{LOAD}=550pF
			3.4 MHz mode	60	—	ns	VDD=3.3V, I_{PULLUP} = 20mA, C_{LOAD}=100pF
I2CS_5	TF:SCL	SDAx and SCLx Fall Time	100 kHz mode	—	300	ns	VDD=3.3V, I_{PULLUP} = 3mA, C_{LOAD}=400pF
			400 kHz mode	—	300	ns	
			1 MHz mode	—	120	ns	VDD=3.3V, I_{PULLUP} = 20mA, C_{LOAD}=550pF
			3.4 MHz mode	—	40	ns	VDD=3.3V, I_{PULLUP} = 20mA, C_{LOAD}=100pF

.....continued

AC CHARACTERISTICS				Standard Operating Conditions: VDD = AVDD = 1.62V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ T_A ≤ +85°C for Industrial			
Param. No.	**Symbol**	**Characteristics**		**Min.**	**Max.**	**Units**	**Conditions**
I2CS_7	TR:SCL	SDAx and SCLx Rise Time	100 kHz mode	—	1000	ns	VDD=3.3V, I_{PULLUP} = 3mA, C_{LOAD}=400pF
			400 kHz mode	—	300	ns	
			1 MHz mode	—	120	ns	VDD=3.3V, I_{PULLUP} = 20mA, C_{LOAD}=550pF
			3.4 MHz mode	—	40	ns	VDD=3.3V, I_{PULLUP} = 20mA, C_{LOAD}=100pF
I2CS_9	TSU:DAT	Data Input Setup Time	100 kHz mode	250	—	ns	VDD=3.3V, I_{PULLUP} = 3mA, C_{LOAD}=400pF
			400 kHz mode	100	—	ns	
			1 MHz mode	50	—	ns	VDD=3.3V, I_{PULLUP} = 20mA, C_{LOAD}=550pF
			3.4 MHz mode	10	—	ns	VDD=3.3V, I_{PULLUP} = 20mA, C_{LOAD}=100pF
I2CS_11	THD:DAT	Data Input Hold Time (1)	100 kHz mode	300	—	µs	VDD=3.3V, I_{PULLUP} = 3mA, C_{LOAD}=400pF
			400 kHz mode	300	—	µs	
			1 MHz mode	300	—	µs	VDD=3.3V, I_{PULLUP} = 20mA, C_{LOAD}=550pF
			3.4 MHz mode	5	—	ns	VDD=3.3V, I_{PULLUP} = 20mA, C_{LOAD}=100pF
I2CS_13	TSU:STA	Start Condition Setup Time	100 kHz mode	4.7	—	µs	VDD=3.3V, I_{PULLUP} = 3mA, C_{LOAD}=400pF
			400 kHz mode	0.6	—	µs	
			1 MHz mode	0.26	—	µs	VDD=3.3V, I_{PULLUP} = 20mA, C_{LOAD}=550pF
			3.4 MHz mode	160	—	ns	VDD=3.3V, I_{PULLUP} = 20mA, C_{LOAD}=100pF
I2CS_15	THD:STA	Start Condition Hold Time	100 kHz mode	4.0	—	µs	VDD=3.3V, I_{PULLUP} = 3mA, C_{LOAD}=400pF
			400 kHz mode	0.6	—	µs	
			1 MHz mode	0.26	—	µs	VDD=3.3V, I_{PULLUP} = 20mA, C_{LOAD}=550pF
			3.4 MHz mode	160	—	ns	VDD=3.3V, I_{PULLUP} = 20mA, C_{LOAD}=100pF
I2CS_17	TSU:STO	Stop Condition Setup Time	100 kHz mode	4.0	—	µs	VDD=3.3V, I_{PULLUP} = 3mA, C_{LOAD}=400pF
			400 kHz mode	0.6	—	µs	
			1 MHz mode	0.26	—	µs	VDD=3.3V, I_{PULLUP} = 20mA, C_{LOAD}=550pF
			3.4 MHz mode	160	—	ns	VDD=3.3V, I_{PULLUP} = 20mA, C_{LOAD}=100pF

.....continued

AC CHARACTERISTICS				Standard Operating Conditions: VDD = AVDD = 1.62V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ T_A ≤ +85°C for Industrial			
Param. No.	**Symbol**	**Characteristics**		**Min.**	**Max.**	**Units**	**Conditions**
I2CS_21	TAA:SCL	Output Valid from Clock	100 kHz mode	—	3.45	µs	VDD=3.3V, I_{PULLUP} = 3mA, C_{LOAD}=400pF
			400 kHz mode	—	0.9	µs	
			1 MHz mode	—	0.45	µs	VDD=3.3V, I_{PULLUP} = 20mA, C_{LOAD}=550pF
			3.4 MHz mode	—	100	ns	VDD=3.3V, I_{PULLUP} = 20mA, C_{LOAD}=100pF
I2CS_23	TBF:SDA	Bus Free Time (2)	100 kHz mode	4.7	—	µs	VDD=3.3V, I_{PULLUP} = 3mA, C_{LOAD}=400pF
			400 kHz mode	1.3	—	µs	
			1 MHz mode	0.5	—	µs	VDD=3.3V, I_{PULLUP} = 20mA, C_{LOAD}=550pF
			3.4 MHz mode	160	—	ns	VDD=3.3V, I_{PULLUP} = 20mA, C_{LOAD}=100pF

Notes:

1. Longest delay between data hold timing based on bitfield SDAHOLD of register CTRLA from SERCOM Module and timing based on 4 period of GCLK_SERCOM for 100kHz/400kHz/1MHz mode.
2. The amount of time the bus must be free before a new transmission can start (STOP condition to START condition).

49.32 TC Module Electrical Specifications

Figure 49-20. TCx Timer Capture Input Module Timing Diagrams

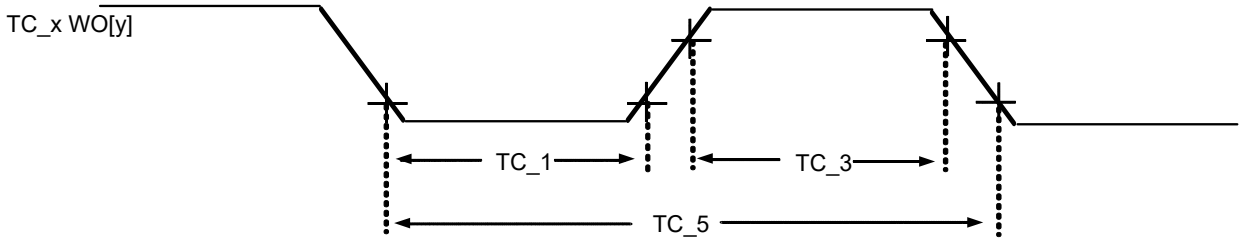


Figure 49-21. TCx Timer Compare Output Module Timing Diagrams

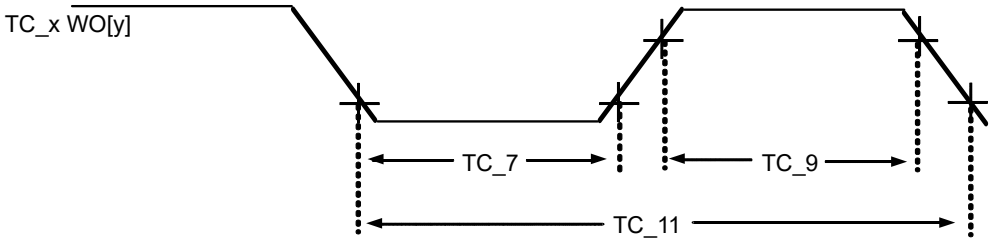


Table 49-44. TCx Timer Capture Module Electrical Specifications

AC CHARACTERISTICS			Standard Operating Conditions: VDD = AVDD = 1.62V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ T_A ≤ +85°C for Industrial				
Param. No.	**Symbol**	**Characteristics**	**Min.**	**Typ.**	**Max.**	**Units**	**Conditions**
TC_1	TC_{INLOW}	Capture TCx Input Low Time	$2/f_{GCLK\_TCx}$	—	—	ns	VDD(min)
TC_3	TC_{INHIGH}	Capture TCx Input High Time	$2/f_{GCLK\_TCx}$	—	—	ns	VDD(min)
TC_5	$TC_{INPERIOD}$	Capture Input Period	$4/f_{GCLK\_TCx}$	—	—	ns	VDD(min)
TC_7	TC_{OUTLOW}	Compare TCx Output Low Time	3 x DI_27	—	—	ns	VDD(min)
TC_9	$TC_{OUTHIGH}$	Compare TCx Output High Time	3 x DI_25	—	—	ns	VDD(min)
TC_11	$TC_{OUTPERIOD}$	Compare Output Period	TC_7 + TC_9	—	—	ns	VDD(min)
TC_13	f_{GCLK_TCx}	TC peripheral module clock frequency	—	—	FCLK_37	MHz	VDD(min), FCLK_37: Refer to Maximum Clock Frequencies Electrical Specifications

49.33 TCC Module Electrical Specifications

Figure 49-22. TCCx Timer Capture Input Module Timing Diagrams

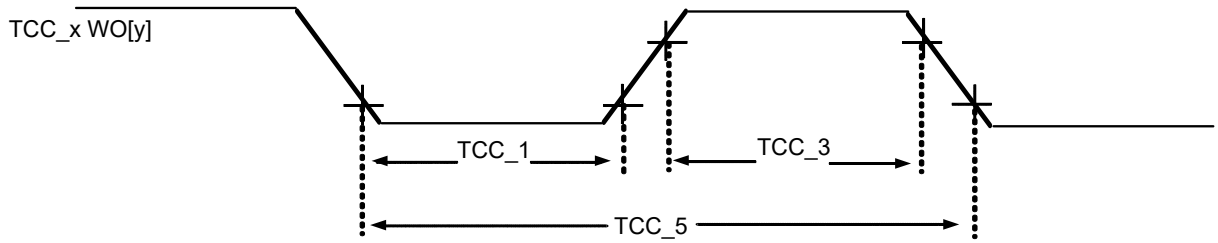


Figure 49-23. TCCx Timer Compare Output Module Timing Diagrams

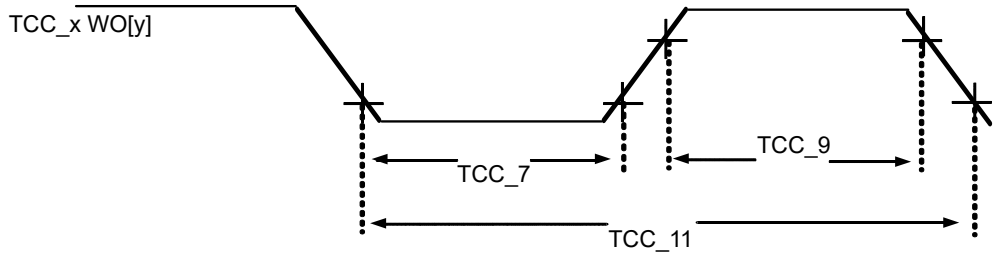


Figure 49-24. TCCx Timer Compare Fault Output Module Timing Diagrams

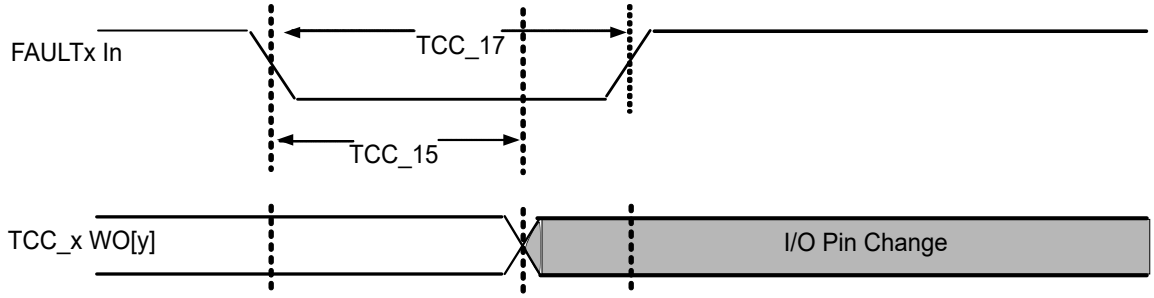


Table 49-45. TCCx Timer Capture Module Electrical Specifications

AC CHARACTERISTICS			Standard Operating Conditions: VDD = AVDD = 1.62V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ T_A ≤ +85°C for Industrial				
Param. No.	**Symbol**	**Characteristics**	**Min.**	**Typ.**	**Max.**	**Units**	**Conditions**
TCC_1	TCC_{INLOW}	Capture TCCx Input Low Time	$2/f_{GCLK\_TCCx}$	—	—	ns	VDD(min)
TCC_3	TCC_{INHIGH}	Capture TCCx Input High Time	$2/f_{GCLK\_TCCx}$	—	—	ns	VDD(min)
TCC_5	$TCC_{INPERIOD}$	Capture Input Period	$4/f_{GCLK\_TCCx}$	—	—	ns	VDD(min)
TCC_7	TCC_{OUTLOW}	Compare TCCx Output Low Time	3 x DI_27	—	—	ns	VDD(min) DI_27: Refer to I/O Pin Electrical Specifications

.....continued

AC CHARACTERISTICS			Standard Operating Conditions: VDD = AVDD = 1.62V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ T_A ≤ +85°C for Industrial				
Param. No.	Symbol	Characteristics	Min.	Typ.	Max.	Units	Conditions
TCC_9	$TCC_{OUTHIGH}$	Compare TCCx Output High Time	3 x DI_25	—	—	ns	VDD(min) DI_25: Refer to I/O Pin Electrical Specifications
TCC_11	$TCC_{OUTPERIOD}$	Compare Output Period	TCC_7 + TCC_9	—	—	ns	VDD(min)
TCC_13	f_{GCLK_TCCx}	TCC peripheral module clock frequency	—	—	FCLK_35	MHz	VDD(min), FCLK_35: Refer to Maximum Clock Frequencies Electrical Specifications
TCC_15	TCC_{FD}	Fault Input to I/O Pin Change	—	—	90.1	ns	VDD(min)

49.34 USB Electrical Specifications

Table 49-46. USB Electrical Specifications

AC CHARACTERISTICS			Standard Operating Conditions: VDD = AVDD = 1.62V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ T_A ≤ +85°C for Industrial				
Param. No.	**Symbol**	**Characteristics**	**Min.**	**Typ.**	**Max.**	**Units**	**Conditions**
USB_1	VUSB	USB Transceiver Voltage	3	—	3.6	V	—
VBUS Supply							
USB_3	VBUS	High-power Port	4.75	—	5.25	V	500 mA Load
USB_5		Low-power Port	4.4	—	5.25	V	100 mA Load
USB_7	VILUSB	Input Low Voltage for USB Buffer	—	—	0.8	V	—
USB_9	VIHUSB	Input High Voltage for USB Buffer	2	—	—	V	—
USB_11	VDIFS	Differential Input Sensitivity	0.2	—	—	V	The difference between D+ and D- must exceed this value while VCM is met
USB_13	VCM	Differential Common Mode Range	0.8	—	2.5	V	V_{USB} = 3.0V-to-3.6V
USB_15	ZOUT	Driver Output Impedance	28	—	44	Ω	—
USB_17	VOLUSB	Voltage Output Low	—	—	0.3	V	1.425 kΩ load connected to V_{USB} = 3.6V
USB_19	VOHUSB	Voltage Output High	2.8	—	—	V	14.25 kΩ load connected to ground w/V_{USB} = 3.0V
USB_23	USBCLKS	USB Clock Source (1) f_{GCLK_USB} of 48 MHz ± 0.25%	—	48	—	MHz	f_{GCLK_USB} = 48 MHz ± 0.25%
USB_25	USB_{AHB}	Min. AHB Clock for USB operations	12	—	—	MHz	—

Note:

1. External Crystal or clock oscillator ≤ 50 ppm with FDPLL96M only for f_{GCLK_USB} of 48 MHz ± 0.25%.

Table 49-47. USB Clocks Configuration

USB Clock configuration			Standard Operating Conditions: VDD = AVDD = 1.62V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ T_A ≤ +85°C for Industrial	
Param. No.	**Symbol**	**Characteristics**	**Device mode operation**	**Host mode operation**
USB_31	USB_GCLK_SRC	DFLL48M Open loop	No	No
		DFLL48M Close loop, Ref. internal OSC source	No	No
		DFLL48M Close loop, Ref. external XOSC source	Yes	No
		DFLL48M Close loop, Ref. SOF (USB recovery mode)	Yes (1)	N/A
		FDPLL96M internal OSC	No	No
		FDPLL96M external OSC (with FDPLL96M reference clock ≤ 1MHz)	Yes	No
		FDPLL96M external OSC (with FDPLL96M reference clock >1MHz)	Yes (2)	Yes

.....continued

USB Clock configuration			Standard Operating Conditions: VDD = AVDD = 1.62V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ T_A ≤ +85°C for Industrial	
Param. No.	Symbol	Characteristics	Device mode operation	Host mode operation

Notes:

1. When using DFLL48M in USB recovery mode, the Fine Step value must be 0x0A to guarantee a USB clock at +/-0.25% before 11ms after a resume. Only usable in LDO regulator mode.
2. FDPLL96M lock time is short when the clock frequency source is high (> 1 MHz). Thus, FDPLL96M and external OSC can be stopped during USB suspend mode to reduce consumption and guarantee a USB resume signal time, refer to “USB specification” for additional information.

49.35 NVM Block (Flash, Data Flash, NVM Configuration Rows) Electrical Specifications

Table 49-48. NVM Block Electrical Specifications

AC CHARACTERISTICS				Standard Operating Conditions: VDD = AVDD = 1.62V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ T_A ≤ +85°C for Industrial						
Param. No.	Symbol	Characteristics		Min.	Typ.		Max.		Units	Conditions
					PL0	PL2	PL0	PL2		
NVM_1	FRETEN (1)	Flash Data Retention		20	—		—		Yrs	Under all conditions less than Absolute Maximum Ratings specifications
NVM_3	EP (1)	Cell Endurance (Flash Erase or Write Operation)		10K	—		—		Cycles	
NVM_3A		Cell Endurance using Tamper Erase		50						
NVM_5	F_{READ} (2,4)	Flash Read	0 Wait State	—	—	—	8	11	MHz	VDD=3.3V
			1 Wait States	—	—	—	12	22		
			2 Wait States	—	—	—	12	35		
			3 Wait States	—	—	—	12	48		
			0 Wait State	—	—	—	6	11	MHz	VDD=1.8V
			1 Wait States	—	—	—	12	22		
			2 Wait States	—	—	—	12	35		
			3 Wait States	—	—	—	12	48		
NVM_7	T_{FPW} (3,4)	Program Cycle Time	Write Page	—	—		2.5		ms	VDD=3.3V
NVM_9	T_{CE} (4)		Erase Chip	—	2		12		sec	
NVM_11	T_{FEB} (3,4)		Erase Row	—	—		6		ms	
NVM_13	IDD_{PROG}	Supply Current during Programming		—	—		10		mA	VDD=3.3V

Notes:

1. Reliability characteristics are given when not using tamper erase operations except if noted.
2. 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.
3. For this Flash technology, a maximum number of eight consecutive writes is allowed per row. Once this number is reached, a row erase is mandatory.
4. T_A = 25°C.

49.36 FREQM Electrical Specifications

Table 49-49. Frequency Meter Electrical Specifications

AC CHARACTERISTICS			Standard Operating Conditions: VDD = AVDD = 1.62V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ T_A ≤ +85°C for Industrial				
Param. No.	Symbol	Characteristics	Min.	Typ.	Max.	Units	Conditions
FM_1	FM_{LOW}	GCLK_IO[7:0] Input Low Time	21.9	—	—	ns	VDD(min)
FM_3	FM_{HIGH}	GCLK_IO[7:0] Input High Time	18.5	—	—	ns	
FM_5	FM_{PERIOD}	GCLK_IO[7:0] Input Period	40.4	—	—	ns	VDD(min) FCLK_15, FCLK_17: Refer to Maximum Clock Frequencies Electrical Specifications
FM_7	fGCLK_FREQM_REF	FREQM Reference	—	—	FCLK_17	MHz	
FM_9	fGCLK_FREQM_MSR	FREQM Measure	—	—	FCLK_15	MHz	

49.37 SWD 2-Wire Electrical Specifications

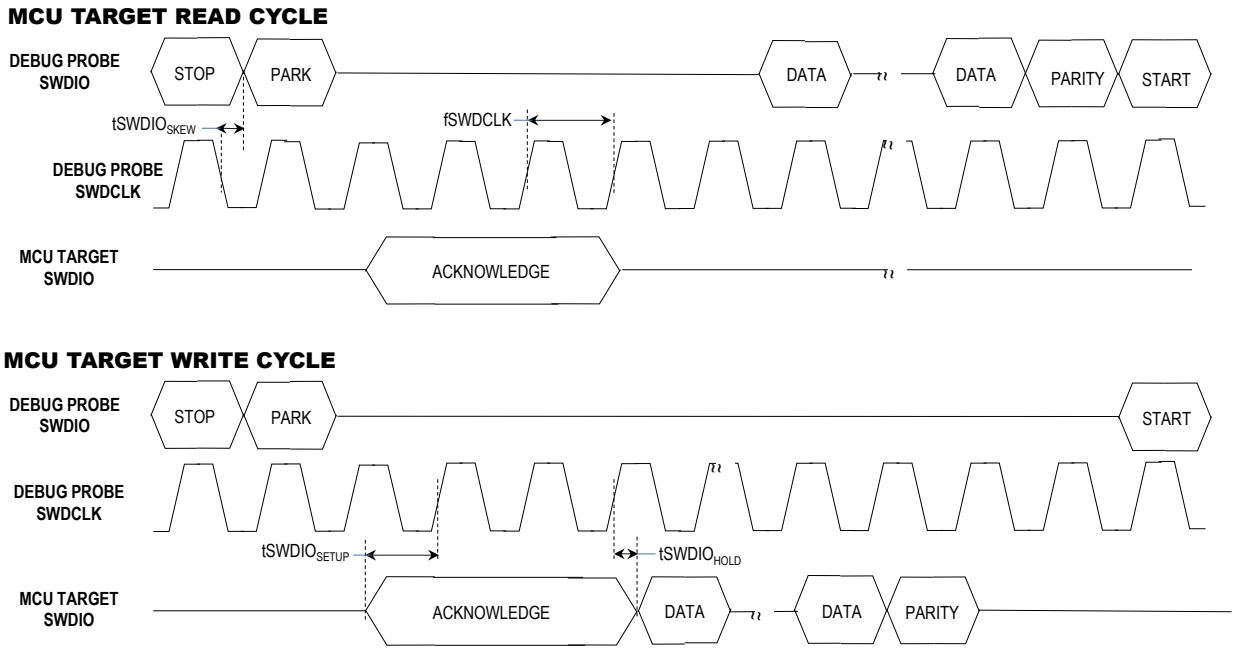


Table 49-50. SWD 2-Wire Electrical Specifications

AC CHARACTERISTICS			Standard Operating Conditions: VDD = AVDD = 1.62V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ T_A ≤ +85°C for Industrial				
Param. No.	**Symbol**	**Characteristics**	**Min.**	**Typ.**	**Max.**	**Units**	**Conditions**
SWD_1	fSWDCLK	SWDCLK Clock Frequency	—	—	10	MHz	Performance Level Mode = PL0
			—	—	20	MHz	Performance Level Mode = PL2
SWD_3	$tSWDCLK_{HIGH}$	SWDCLK Clock High Time	1 / (2 * fSWDCLK)	—	—	ns	—
SWD_5	$tSWDCLK_{LOW}$	SWDCLK Clock Low Time	1 / (2 * fSWDCLK)	—	—	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	—

49.38 TRNG Electrical Specifications

Table 49-51. True Random Number Generator Module Electrical Specifications

AC CHARACTERISTICS			Standard Operating Conditions: VDD = AVDD = 1.62V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ T_A ≤ +85°C for Industrial				
Param. No.	Symbol	Characteristics	Min.	Typ.	Max.	Units	Conditions
TRNG_1	$TRNG_{WKUP}$	TRNG Wake-up Time	100	—	—	ms	Delay between TRNG Enable (CTRLA.ENABLE=1) and first random number read

50. Packaging Information

50.1 Package Marking Information

All devices are marked with the Microchip logo and the ordering code.

Where:

- "XXXXXX": Part Number
- “YYWWNNN”: Trace Code
  - "YY": Manufacturing Year (two last digit(s))
  - "WW": Manufacturing Week
  - "NNN": Internal Code

Figure 50-1. 48-pin TQFP

Figure 50-2. 64, 100-pin TQFP

Figure 50-3. 48, 64-pin QFN

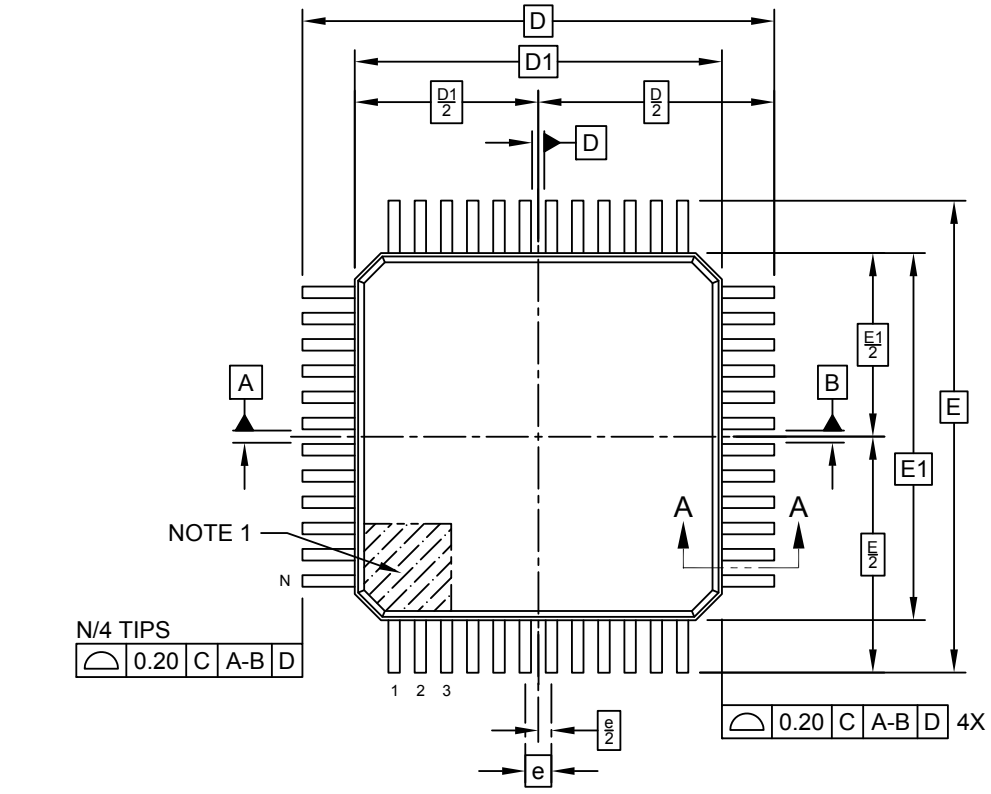
50.2 Package Drawings

For the most current package drawings, see the Microchip Packaging Specification located at http://www.microchip.com/packaging

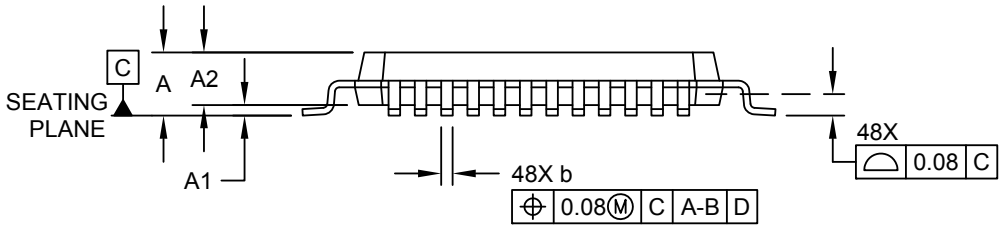
50.2.1 48-pin TQFP

48-Lead Plastic Thin Quad Flatpack (Y8X) - 7x7x1.0 mm Body [TQFP]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



TOP VIEW

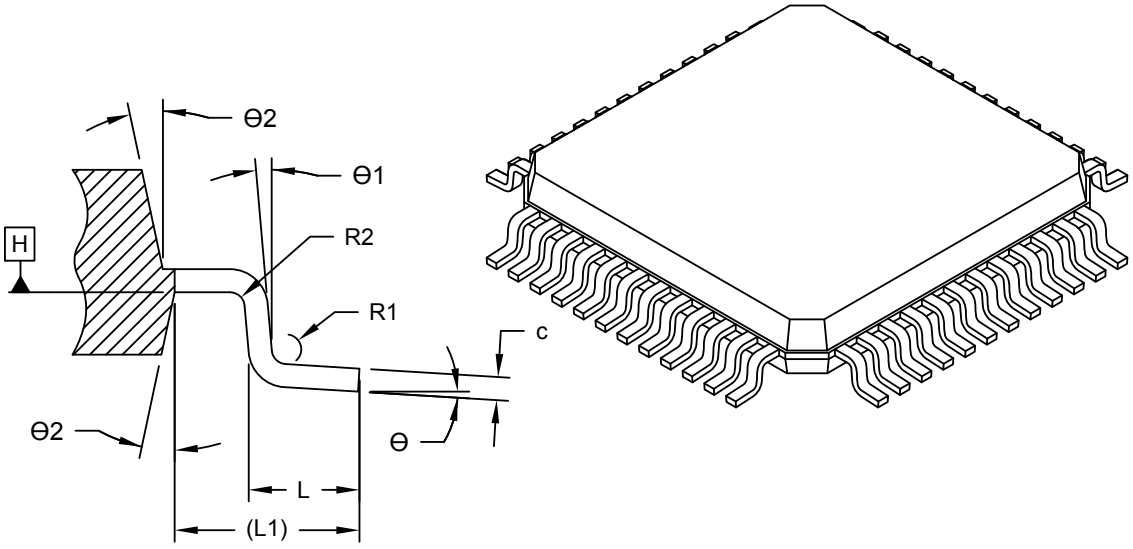


SIDE VIEW

Microchip Technology Drawing C04-300-Y8X Rev D Sheet 1 of 2

48-Lead Plastic Thin Quad Flatpack (Y8X) - 7x7x1.0 mm Body [TQFP]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



SECTION A-A

Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Terminals	N		48	
Pitch	e		0.50 BSC	
Overall Height	A	-	-	1.20
Standoff	A1	0.05	-	0.15
Molded Package Thickness	A2	0.95	1.00	1.05
Overall Length	D		9.00 BSC	
Molded Package Length	D1		7.00 BSC	
Overall Width	E		9.00 BSC	
Molded Package Width	E1		7.00 BSC	
Terminal Width	b	0.17	0.22	0.27
Terminal Thickness	c	0.09	-	0.16
Terminal Length	L	0.45	0.60	0.75
Footprint	L1		1.00 REF	
Lead Bend Radius	R1	0.08	-	-
Lead Bend Radius	R2	0.08	-	0.20
Foot Angle	Θ	0°	3.5°	7°
Lead Angle	Θ1	0°	-	-
Mold Draft Angle	Θ2	11°	12°	13°

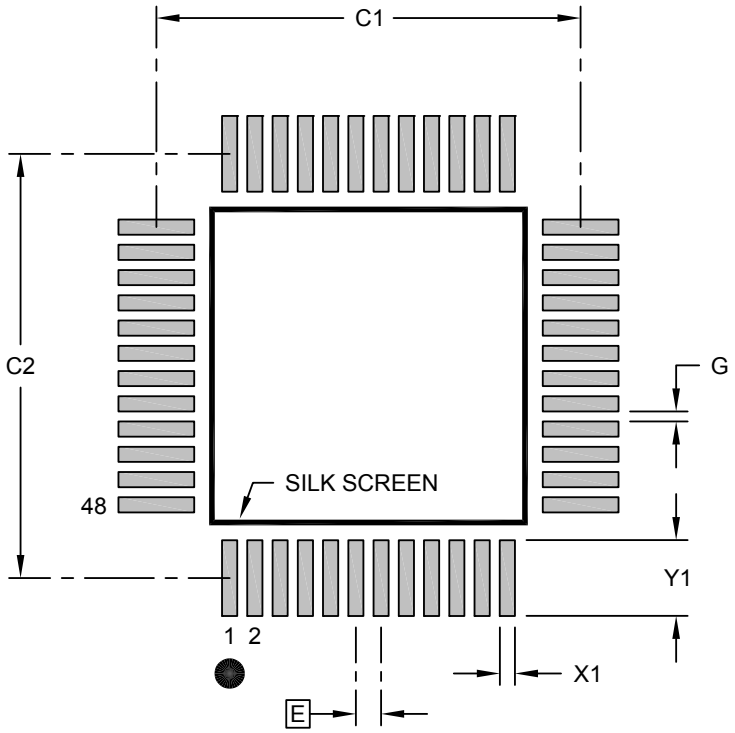
Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. 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 C04-300-Y8X Rev D Sheet 2 of 2

48-Lead Plastic Thin Quad Flatpack (Y8X) - 7x7x1.0 mm Body [TQFP]

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	
Contact Pad Spacing	C1		8.40	
Contact Pad Spacing	C2		8.40	
Contact Pad Width (X48)	X1			0.30
Contact Pad Length (X48)	Y1			1.50
Distance Between Pads	G	0.20		

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 C04-2300-Y8X Rev D

Table 50-1. Device and Package Maximum Weight

140	mg

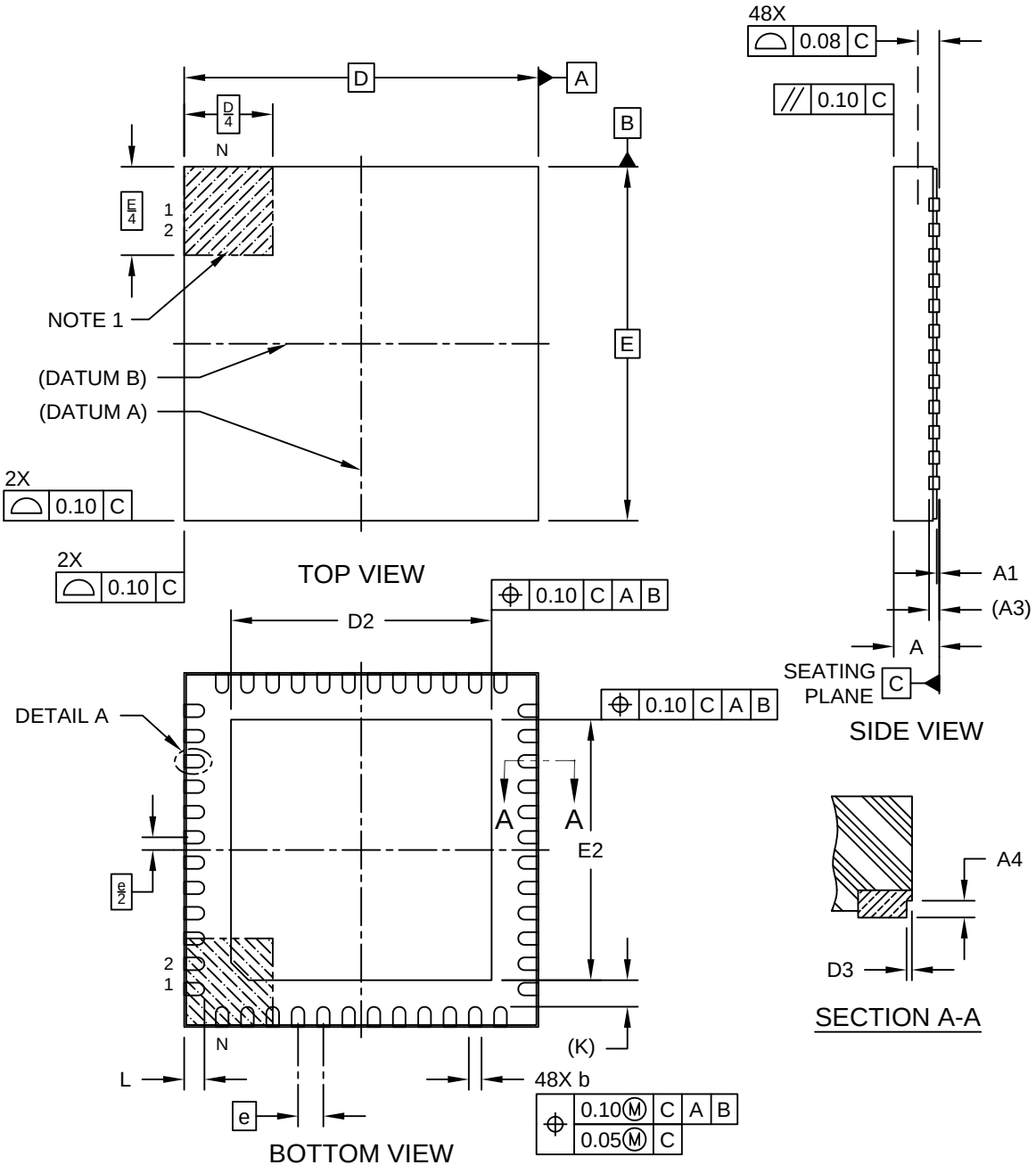
Table 50-2. Package Reference

Package Outline Drawing MCHP reference	C04-00300
JESD97 Classification	E3

50.2.2 48-Pin VQFN

48-Lead Very Thin Plastic Quad Flat, No Lead Package (U5B) - 7x7 mm Body [VQFN] With 5.15 mm Exposed Pad and Stepped Wettable Flanks; Atmel Legacy ZLH

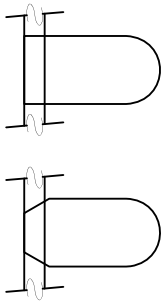
Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



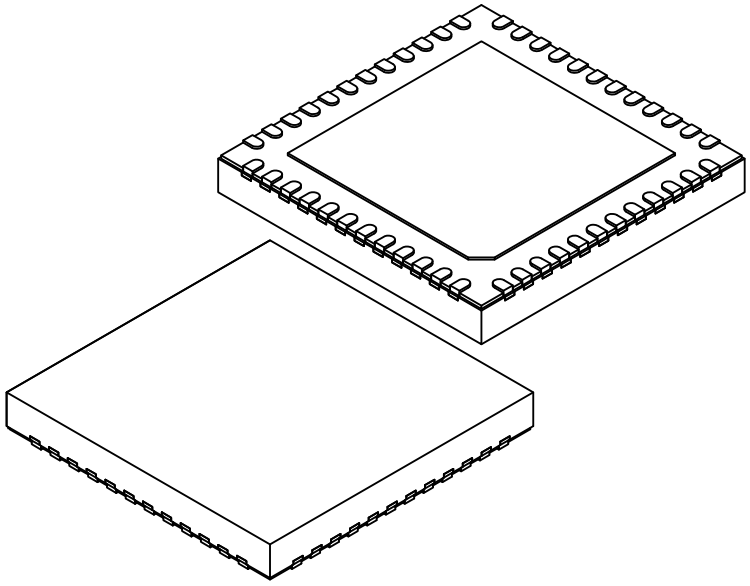
Microchip Technology Drawing C04-21493 Rev A Sheet 1 of 2

48-Lead Very Thin Plastic Quad Flat, No Lead Package (U5B) - 7x7 mm Body [VQFN] With 5.15 mm Exposed Pad and Stepped Wettable Flanks; Atmel Legacy ZLH

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		48	
Pitch	e		0.50 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		7.00 BSC	
Exposed Pad Length	D2	5.05	5.15	5.25
Overall Width	E		7.00 BSC	
Exposed Pad Width	E2	5.05	5.15	5.25
Terminal Width	b	0.20	0.25	0.30
Terminal Length	L	0.35	0.40	0.45
Terminal-to-Exposed-Pad	K		0.53 REF	
Wettable Flank Step Length	D3	-	-	0.085
Wettable Flank Step Height	A4	0.10	-	0.19

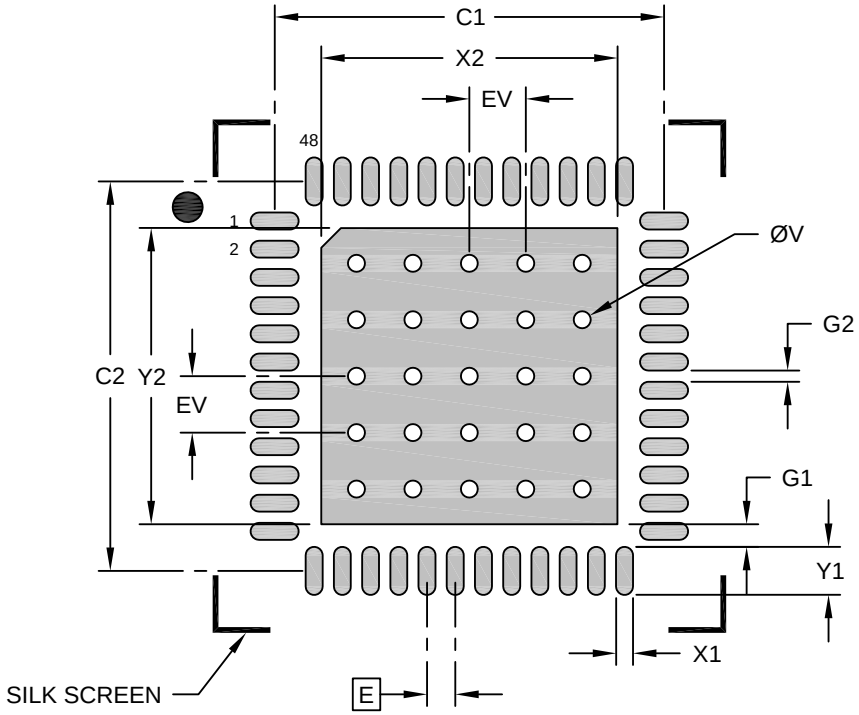
Notes:

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 C04-21493 Rev A Sheet 2 of 2

48-Lead Very Thin Plastic Quad Flat, No Lead Package (U5B) - 7x7 mm Body [VQFN] With 5.15 mm Exposed Pad and Stepped Wettable Flanks; Atmel Legacy ZLH

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			5.25
Optional Center Pad Length	Y2			5.25
Contact Pad Spacing	C1		6.90	
Contact Pad Spacing	C2		6.90	
Contact Pad Width (X48)	X1			0.30
Contact Pad Length (X48)	Y1			0.85
Contact Pad to Center Pad (X48)	G1	0.20		
Contact Pad to Center Pad (X44)	G2	0.40		
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 C04-23493 Rev A

Table 50-3. Device and Package Maximum Weight

140	mg

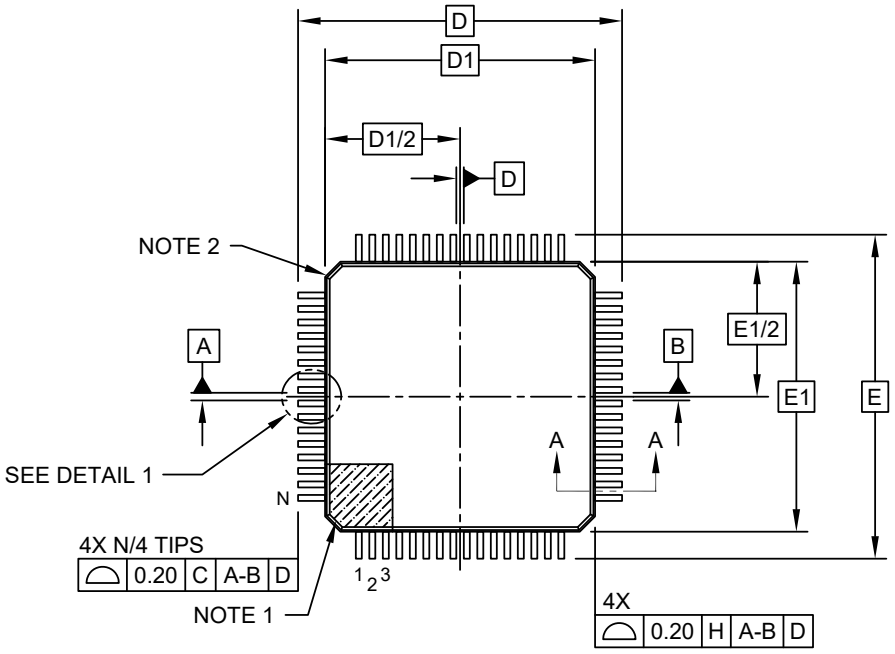
Table 50-4. Package Reference

Package Outline Drawing MCHP reference	C04-21493
JESD97 Classification	E3

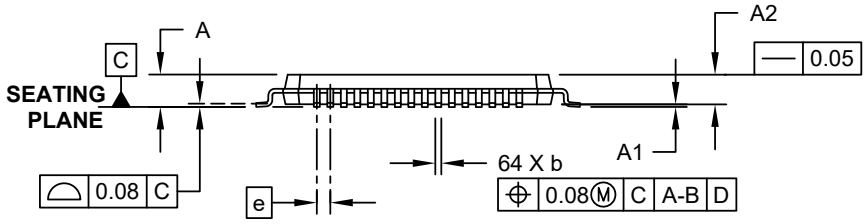
50.2.3 64-pin TQFP

64-Lead Plastic Thin Quad Flatpack (PT)-10x10x1 mm Body, 2.00 mm Footprint [TQFP]

Note:	For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



TOP VIEW

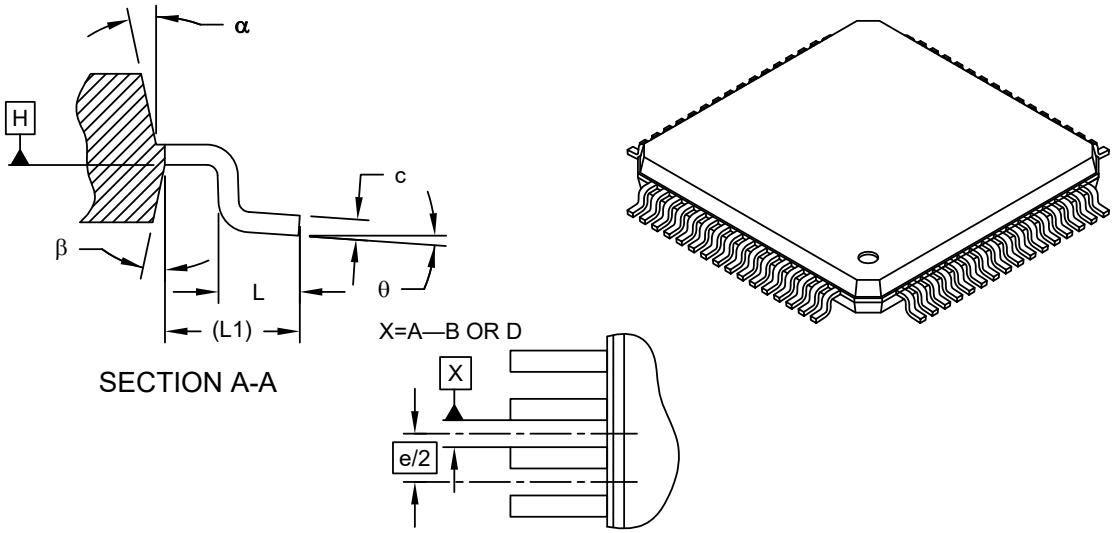


SIDE VIEW

Microchip Technology Drawing C04-085C Sheet 1 of 2

64-Lead Plastic Thin Quad Flatpack (PT)-10x10x1 mm Body, 2.00 mm Footprint [TQFP]

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 Leads	N		64	
Lead Pitch	e		0.50 BSC	
Overall Height	A	-	-	1.20
Molded Package Thickness	A2	0.95	1.00	1.05
Standoff	A1	0.05	-	0.15
Foot Length	L	0.45	0.60	0.75
Footprint	L1		1.00 REF	
Foot Angle	ϕ	0°	3.5°	7°
Overall Width	E		12.00 BSC	
Overall Length	D		12.00 BSC	
Molded Package Width	E1		10.00 BSC	
Molded Package Length	D1		10.00 BSC	
Lead Thickness	c	0.09	-	0.20
Lead Width	b	0.17	0.22	0.27
Mold Draft Angle Top	α	11°	12°	13°
Mold Draft Angle Bottom	β	11°	12°	13°

Notes:

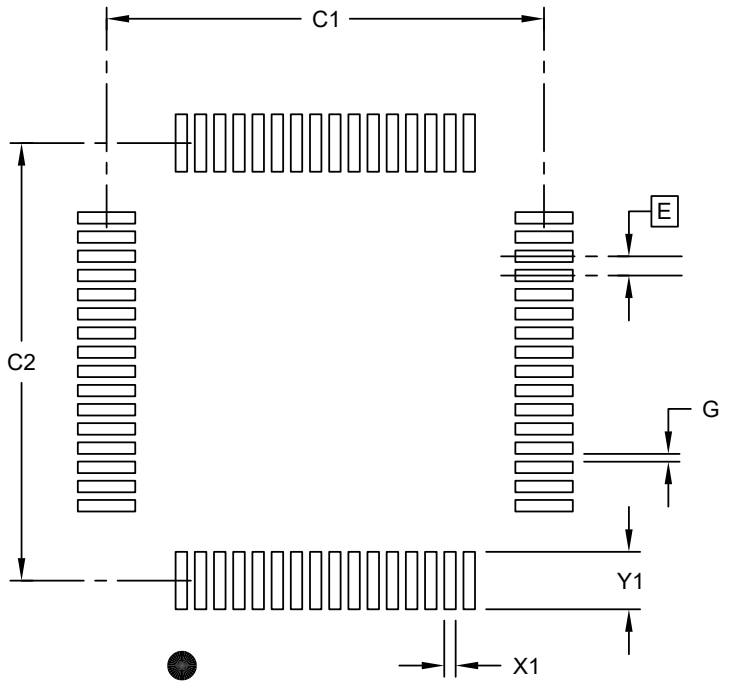
1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Chamfers at corners are optional; size may vary.
3. Dimensions D1 and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.25mm per side.
4. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.
REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-085C Sheet 2 of 2

64-Lead Plastic Thin Quad Flatpack (PT)-10x10x1 mm Body, 2.00 mm Footprint [TQFP]

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	
Contact Pad Spacing	C1		11.40	
Contact Pad Spacing	C2		11.40	
Contact Pad Width (X28)	X1			0.30
Contact Pad Length (X28)	Y1			1.50
Distance Between Pads	G	0.20		

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing C04-2085B Sheet 1 of 1

Table 50-5. Device and Package Maximum Weight

300	mg

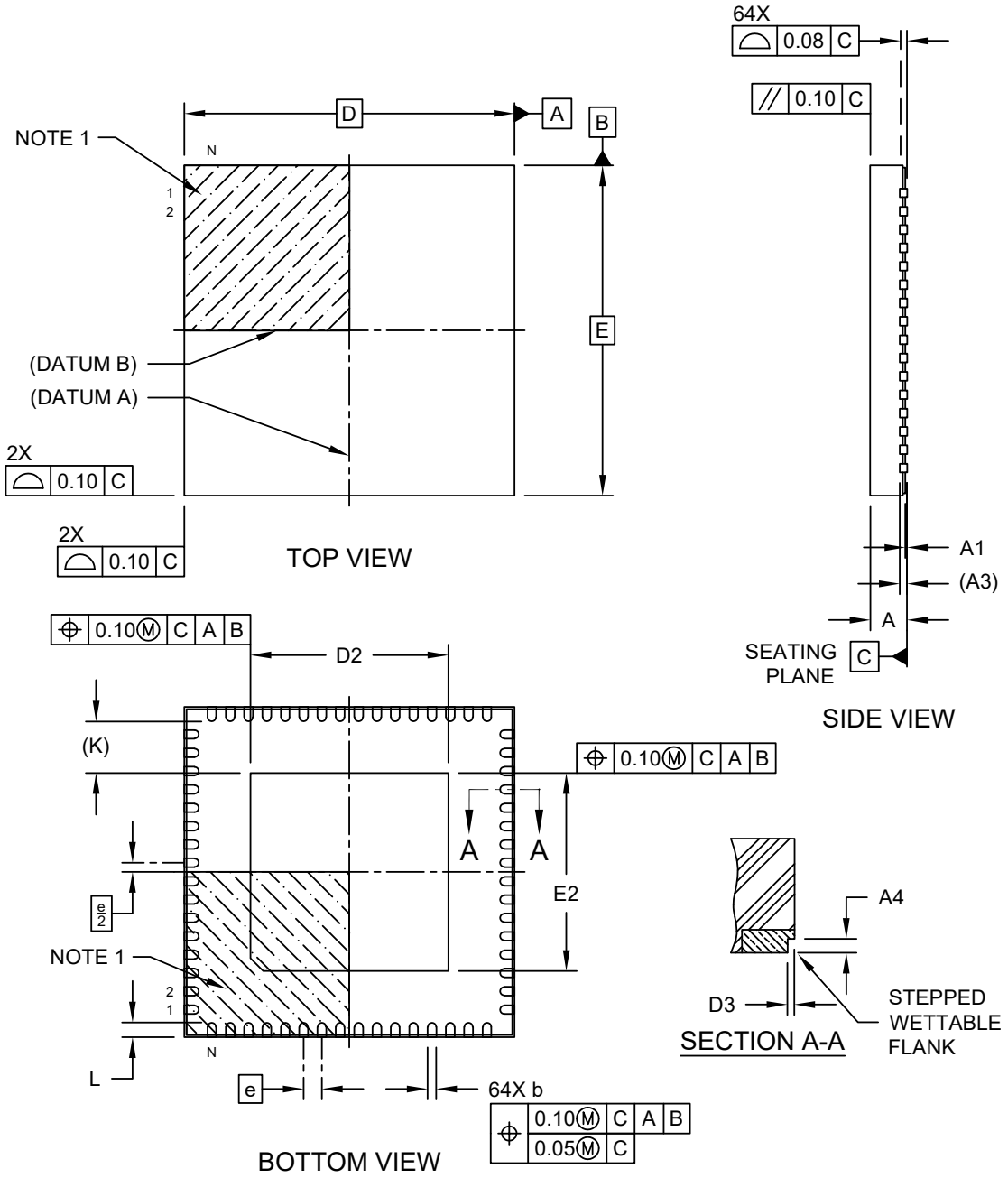
Table 50-6. Package Reference

Package Outline Drawing MCHP reference	C04-00085
JESD97 Classification	E3

50.2.4 64-pin VQFN

64-Lead Very Thin Plastic Quad Flat, No Lead Package (5LX) - 9x9x1.0 mm Body [VQFN] With 5.4 mm Exposed Pad and Stepped Wettable Flanks

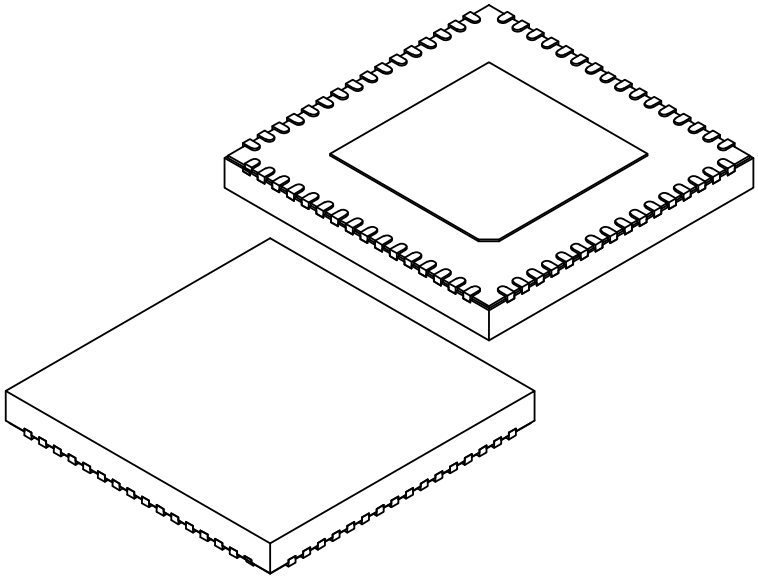
Note:	For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



Microchip Technology Drawing C04-483 Rev E Sheet 1 of 2

64-Lead Very Thin Plastic Quad Flat, No Lead Package (5LX) - 9x9x1.0 mm Body [VQFN] With 5.4 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



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Terminals	N		64	
Pitch	e		0.50 BSC	
Overall Height	A	0.80	0.90	1.00
Standoff	A1	0.00	0.02	0.05
Terminal Thickness	A3		0.203 REF	
Overall Length	D		9.00 BSC	
Exposed Pad Length	D2	5.30	5.40	5.50
Overall Width	E		9.00 BSC	
Exposed Pad Width	E2	5.30	5.40	5.50
Terminal Width	b	0.20	0.25	0.30
Terminal Length	L	0.30	0.40	0.50
Terminal-to-Exposed-Pad	K		1.40 REF	
Wettable Flank Step Length	D3	0.035	0.060	0.085
Wettable Flank Step Height	A4	0.10	-	0.19

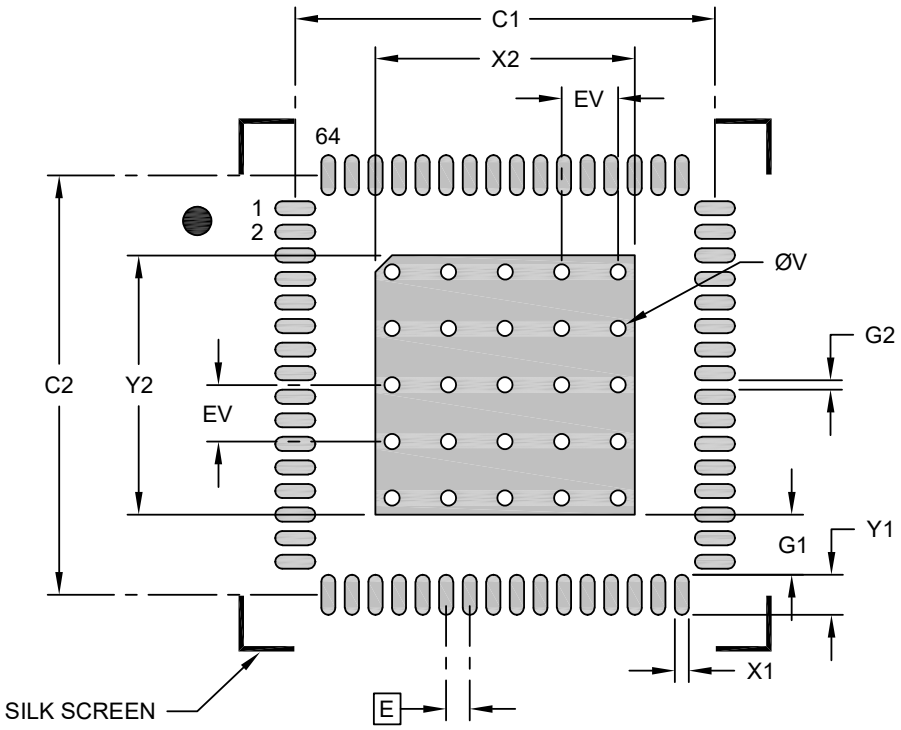
Notes:

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 C04-483 Rev E Sheet 2 of 2

64-Lead Very Thin Plastic Quad Flat, No Lead Package (5LX) - 9x9x1.0 mm Body [VQFN] With 5.4 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			5.50
Optional Center Pad Length	Y2			5.50
Contact Pad Spacing	C1		8.90	
Contact Pad Spacing	C2		8.90	
Contact Pad Width (X64)	X1			0.30
Contact Pad Length (X64)	Y1			0.85
Contact Pad to Center Pad (X64)	G1	1.28		
Contact Pad to Contact Pad (X60)	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 C04-2483 Rev E

Table 50-7. Device and Package Maximum Weight

200	mg

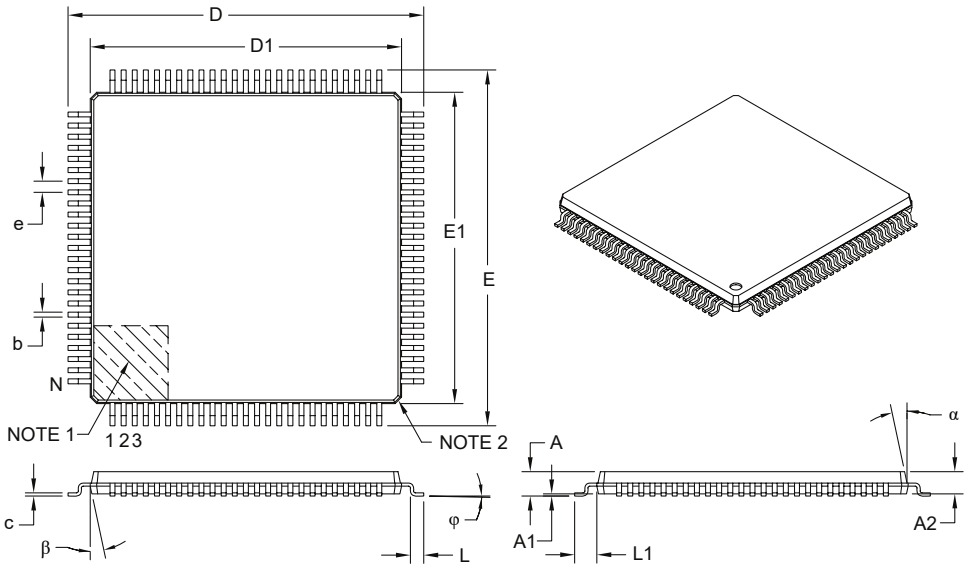
Table 50-8. Package Reference

Package Outline Drawing MCHP reference	C04-00483
JESD97 Classification	E3

50.2.5 100-pin TQFP

100-Lead Plastic Thin Quad Flatpack (PF) – 14x14x1 mm Body, 2.00 mm [TQFP]

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 Leads	N		100	
Lead Pitch	e		0.50 BSC	
Overall Height	A	–	–	1.20
Molded Package Thickness	A2	0.95	1.00	1.05
Standoff	A1	0.05	–	0.15
Foot Length	L	0.45	0.60	0.75
Footprint	L1		1.00 REF	
Foot Angle	φ	0°	3.5°	7°
Overall Width	E		16.00 BSC	
Overall Length	D		16.00 BSC	
Molded Package Width	E1		14.00 BSC	
Molded Package Length	D1		14.00 BSC	
Lead Thickness	c	0.09	–	0.20
Lead Width	b	0.17	0.22	0.27
Mold Draft Angle Top	α	11°	12°	13°
Mold Draft Angle Bottom	β	11°	12°	13°

Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Chamfers at corners are optional; size may vary.
3. Dimensions D1 and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.25 mm per side.
4. Dimensioning and tolerancing per ASME Y14.5M.

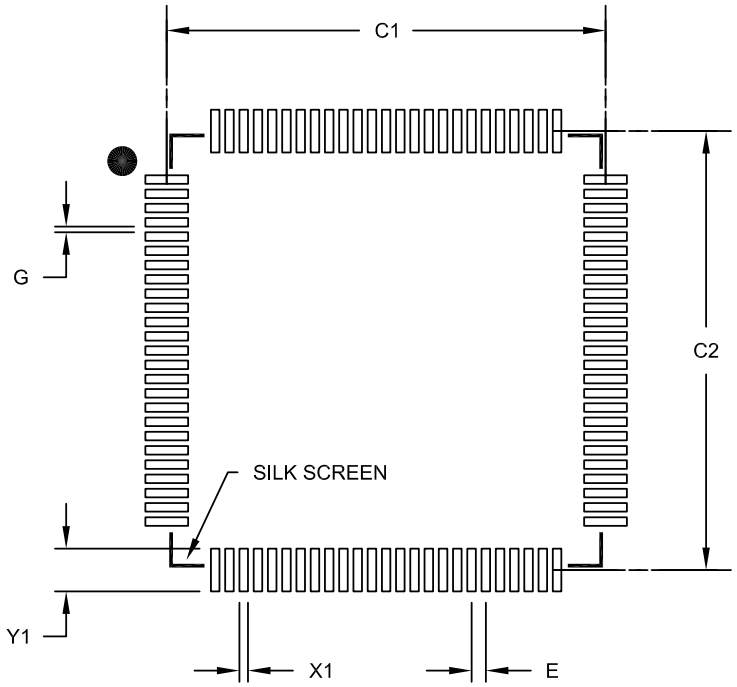
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.

   REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-110B

100-Lead Plastic Thin Quad Flatpack (PF) - 14x14x1 mm Body 2.00 mm Footprint [TQFP]

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	
Contact Pad Spacing	C1		15.40	
Contact Pad Spacing	C2		15.40	
Contact Pad Width (X100)	X1			0.30
Contact Pad Length (X100)	Y1			1.50
Distance Between Pads	G	0.20		

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2110B

Table 50-9. Device and Package Maximum Weight

500	mg

Table 50-10. Package Reference

Package Outline Drawing MCHP reference	C04-0110
JESD97 Classification	E3

50.3 Soldering Profile

The following table gives the recommended soldering profile from J-STD-20.

Table 50-11. Recommended Soldering Profile

Profile Feature	Green Package
Average Ramp-up Rate (217°C to peak)	3°C/s max.
Preheat Temperature 175°C ±25°C	150-200°C
Time Maintained Above 217°C	60-150s
Time within 5°C of Actual Peak Temperature	30s
Peak Temperature Range	260°C
Ramp-down Rate	6°C/s max.
Time 25°C to Peak Temperature	8 minutes max.

A maximum of three reflow passes is allowed per component.

51. Schematic Checklist

51.1 Introduction

The schematic checklist provides the user with the requirements regarding the different pin connections that must be considered before starting any new board design as well as information on the minimum hardware resources required to quickly develop an application with the PIC32CM LE00/LS00/LS60. It does not consider PCB layout constraints.

It also provides recommendations regarding low-power design constraints to minimize power consumption.

This information is not intended to be exhaustive. Its objective is to cover as many configurations of use as possible.

⚠ CAUTION

Operation in Noisy Environment

If the device is operating in an environment with much electromagnetic noise, it must be protected from this noise to ensure reliable operation. In addition to following best practice EMC design guidelines, the recommendations listed in the schematic checklist sections must be followed. In particular, placing decoupling capacitors very close to the power pins, an RC-filter on the RESET pin, and a pull-up resistor on the SWCLK pin is critical for reliable operations.

It is also relevant to eliminate or attenuate noise in order to avoid that it reaches supply pins, I/O pins and crystals.

51.2 Power Supply

The PIC32CM LE00/LS00/LS60 supports a single power supply.

The same voltage must be applied to both VDD and AVDD.

51.2.1 Decoupling Capacitors

The use of decoupling capacitors on power supply pins, such as VDD, and AVDD, is required.

Consider the following criteria when using decoupling capacitors:

- **Bulk capacitors:** Each primary power supply group VDD, AVDD, VDDCORE, and VBAT should have one bulk capacitor. The use of a bulk capacitor is recommended to improve power supply stability. Typical values range from 4.7 µF to 22 µF ceramic or tantalum capacitors with low ESR. This capacitor should be located as close to the device as possible.
- **Value and type of capacitor:** all power pins should have a 100nF bypass cap. A value of 0.1 µF (100 nF),10-20V is recommended. The capacitor should be a low Equivalent Series Resistance (low-ESR) capacitor and have resonance frequency in the range of 20 MHz and higher. It is further recommended that ceramic capacitors be used.
- **Placement on the printed circuit board:** The decoupling capacitors should be placed as close to the pins as possible. It is recommended that the capacitors be placed on the same side of the board as the device. If space is constricted, the capacitor can be placed on another layer on the PCB using a via; however, ensure that the trace length from the pin to the capacitor is within one-quarter inch (6 mm) in length.
- **Handling high frequency noise:** If the board is experiencing high frequency noise, upward of tens of MHz, add a second ceramic-type capacitor in parallel to the above described decoupling capacitor. The value of the second capacitor can be in the range of 0.01 µF to 0.001 µF. Place this second capacitor next to the primary decoupling capacitor. In high-speed circuit designs, consider implementing a decade pair of capacitances as close to the power and ground pins as possible. For example, 0.1 µF in parallel with 0.001 µF.
- **Maximizing performance:** On the board layout from the power supply circuit, run the power and return traces to the decoupling capacitors first, and then to the device pins. This ensures that the decoupling capacitors are first in the power chain. Equally important is to keep the trace length between the capacitor and the power pins to a minimum thereby reducing PCB track inductance.

51.2.2 Power Supply Connections

The following figures shows the recommended power supply connections for Switched/Linear mode and Linear mode only.

Figure 51-1. Power Supply Connection for Switching/Linear Mode

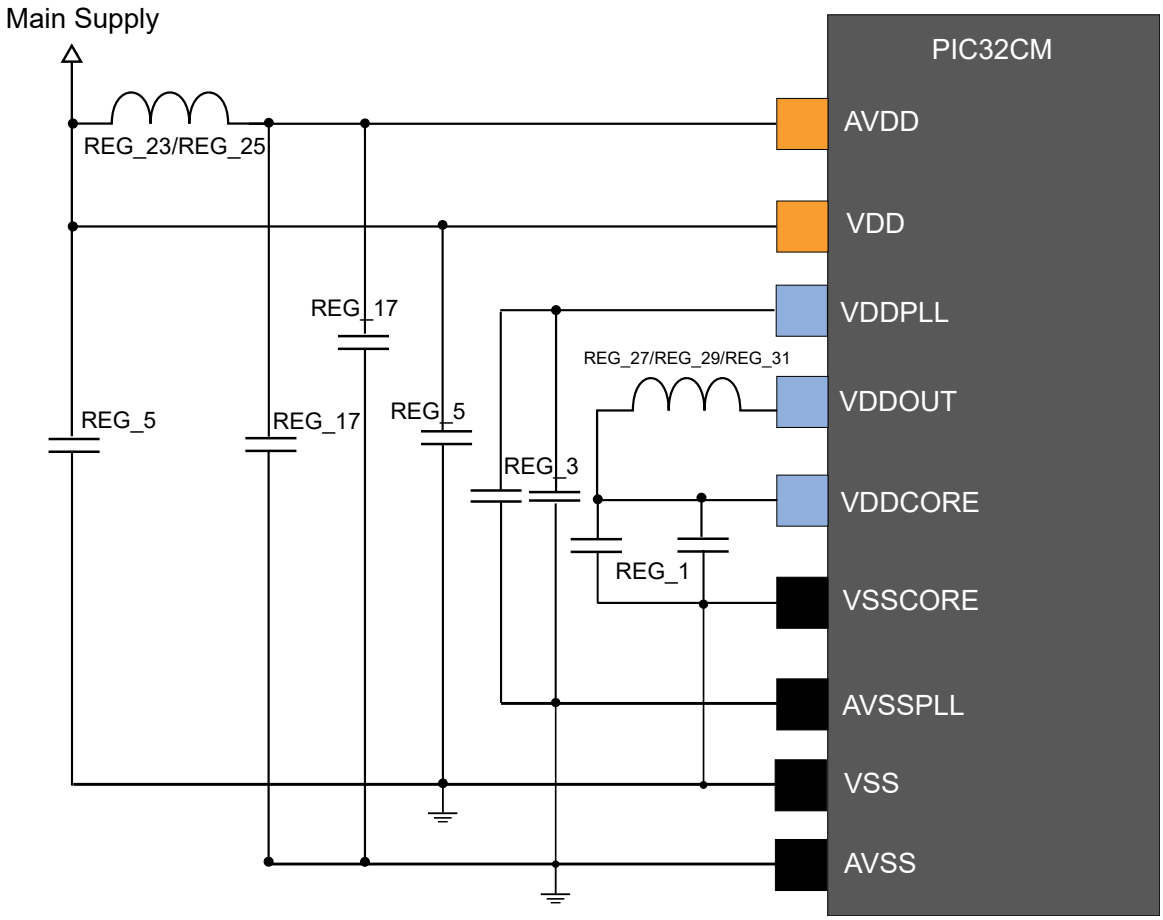


Figure 51-2. Power Supply Connection for Linear Mode Only

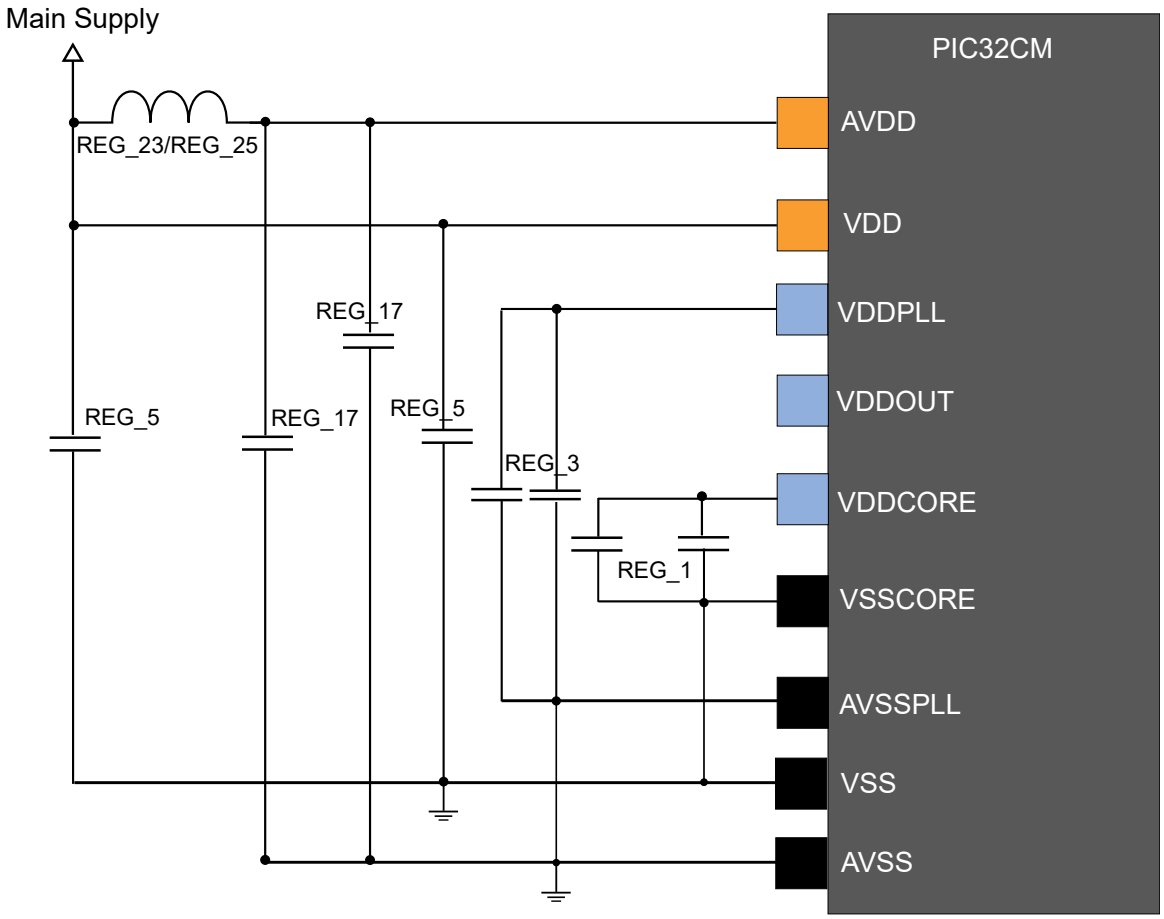


Table 51-1. Power Supply Connections (1,2)

Signal Name	Associated Ground	Recommended Pin Connection
V_{DD}	VSS	Capacitors: refer to REG_5
AVDD	AVSS	Capacitors: refer to REG_17 Ferrite bead(3): refer to REG_23/REG_25
VDDCORE	VSSCORE	Capacitors: refer to REG_1
VDDPLL	AVSSPLL	Capacitors: refer to REG_3
VDDOUT	-	BUCK mode: Inductor required to VDDCORE: refer to REG_27/REG_29/REG_31
		LDO mode: Not Connected

Notes:

1. REG_X values: Refer to Power Supply Electrical Specifications from the Electrical Characteristics chapter.
2. Decoupling capacitors must be placed close to the device, low ESR capacitors must be used for better decoupling.
3. A ferrite bead must be added between VDD and AVDD to attenuate high-frequency digital noise from entering the analog power domain. The bead must provide enough impedance to help isolate digital and analog power domains. Make sure to select a ferrite bead with a low DC resistance to avoid any relevant IR drop across the ferrite bead that could impact analog peripherals performance.

51.2.3 Special Considerations for Packages with an Exposed Pad

The QFN package has an exposed pad that must be connected to VSS.

51.3 External Analog Reference Connections

The following schematic checklist is only necessary if the application is using one or more of the external analog references. If the internal references are used instead, the circuits in the following two figures are not necessary.

Figure 51-3. External Analog Reference Schematic With Two References

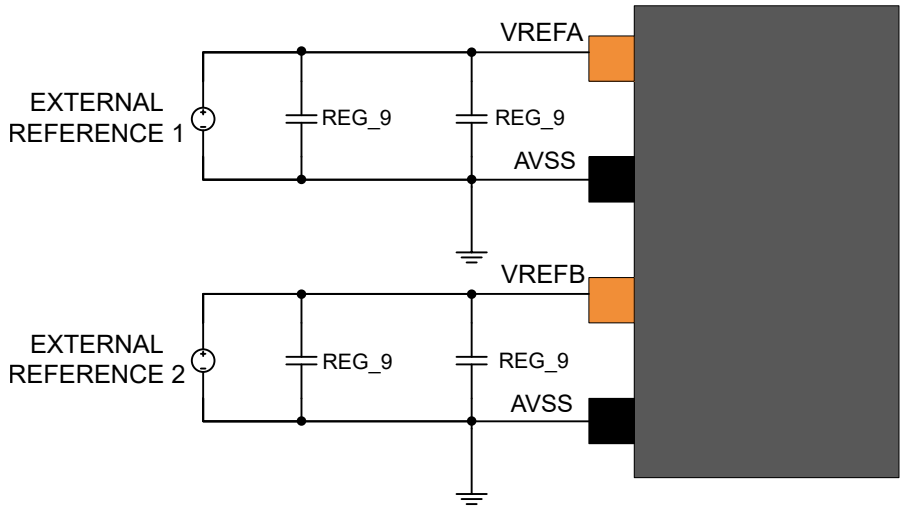


Figure 51-4. External Analog Reference Schematic With One Reference

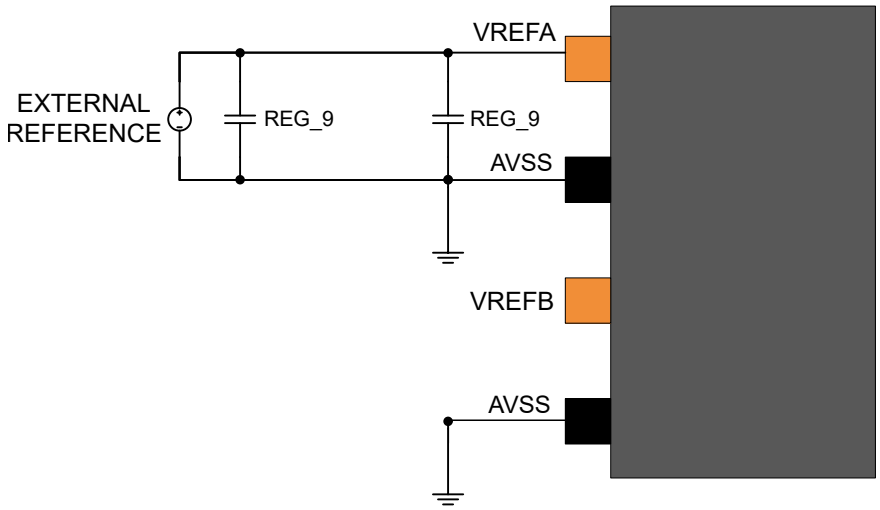


Table 51-2. External Analog Reference Connections

Signal Name	Description	Recommended Pin Connection
VREFx	External reference VREFA/VREFB	Capacitors(2): refer to REG_9(1)
AVSS	Ground	-

Notes:

1. Refer to Power Supply Electrical Specifications from the Electrical Characteristics chapter.
2. Decoupling capacitors should be placed close to the device, low ESR capacitors should be used for better decoupling.

51.4 External Reset Circuit

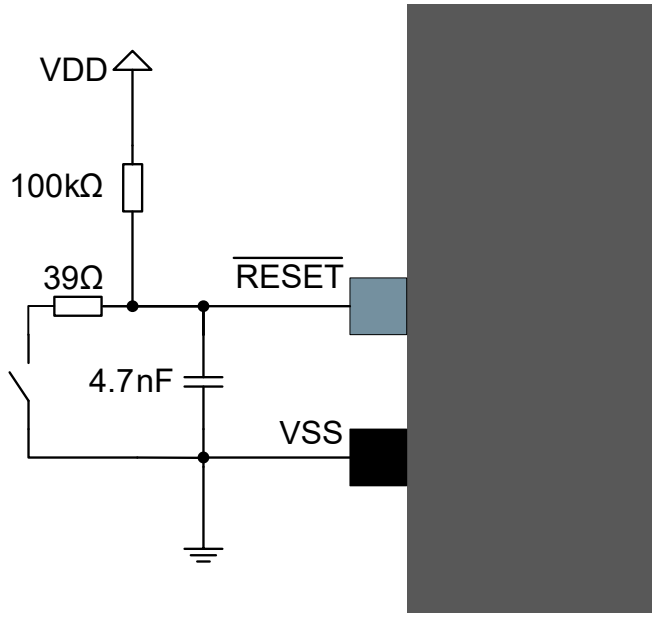
The external Reset circuit is connected to the $\overline{\text{RESET}}$ pin when the external Reset function is used. The circuit is not necessary when the $\overline{\text{RESET}}$ pin is not driven low externally by the application circuitry.

The reset switch can also be removed, if a manual reset is not desired.

A pull-up resistor makes sure that the reset does not go low and unintentionally causing a device reset. An additional resistor has been added in series with the switch to safely discharge the filtering capacitor, that is, preventing a current surge when shorting the filtering capacitor which again can cause a noise spike that can have a negative effect on the system.

Place the components illustrated within one-half inch (12 mm) from the RESET pin.

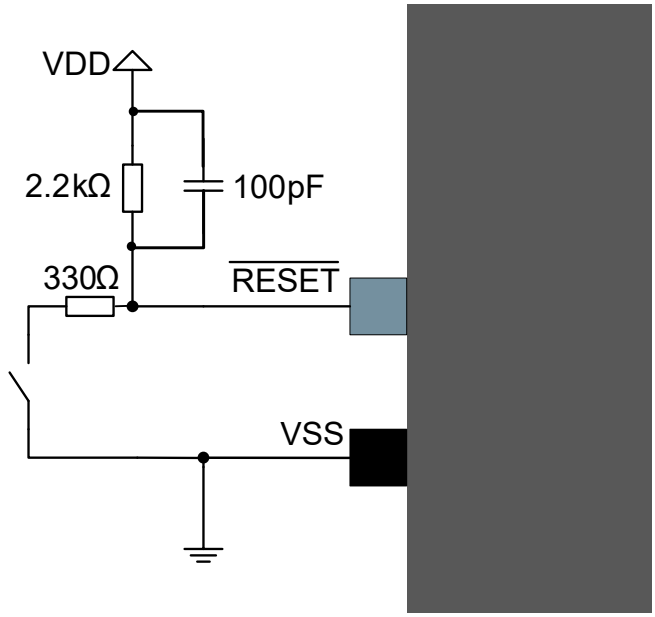
Figure 51-5. External Reset Circuit Schematic (General Purpose)



Note: These values are only given as a typical example. Refer to the Power Supply Electrical Characteristics to determine proper values given in system parameters to meet reset timing requirements (T_{RST}).

As shown in the following figure the reset circuit is intended to improve EFT immunity, but does not filter low-frequency glitches which makes it not suitable as an example for applications requiring debouncing on a reset button:

Figure 51-6. External Reset Circuit Schematic (EFT Immunity Enhancement)



Note: These values are only given as a typical example. Refer to the Power Supply Electrical Characteristics to determine proper values given in system parameters to meet reset timing requirements (T_{RST}).

51.5 Unused or Unconnected Pins

Unused I/O pins should not be allowed to float as inputs. It is recommended that unused inputs be ganged together and connected through a 1k resistor to digital ground individually or in multiple groups through 1k as the PCB layout permits. This minimizes the chip vulnerability to ESD and radiated EMI due to a high-voltage discharge event.

51.6 Clocks and Crystal Oscillators

The oscillator circuit must be placed on the same side of the board as the device. Also, place the oscillator circuit close to the respective oscillator pins, not exceeding one-half inch (12 mm) distance between them. The load capacitors should be placed next to the oscillator itself, on the same side of the board. Use a grounded copper pour around the oscillator circuit to isolate them from surrounding circuits. The grounded copper pour should be routed directly to the MCU ground. Do not run any signal traces or power traces inside the ground pour. Also, if using a two-sided board, avoid any traces on the other side of the board where the crystal is placed.

Important: Crystal selection must be done using the “Crystal Data Sheet” and the “Crystal Oscillator Parameters” as given in the XOSC Electrical Specifications and XOSC32K Electrical Specifications from the Electrical Characteristics chapter.

51.6.1 External Clock Source

Figure 51-7. External Clock Source Schematics

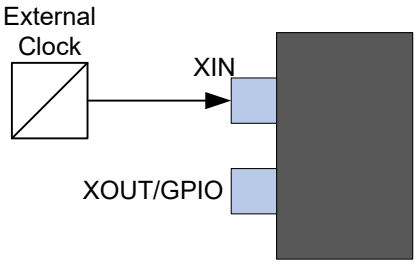
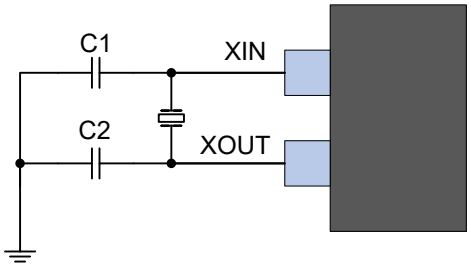


Table 51-3. External Clock Source Connections

Signal Name	Description	Recommended Pin Connection
XIN	Input for an external clock signal External Clock frequency range specified in the XOSC Electrical Specifications (XOSC_35 parameter)	XIN is used as input for an external clock signal
XOUT/GPIO	NC/GPIO	Can be left unconnected or used as normal GPIO

51.6.2 Crystal Oscillator

Figure 51-8. Crystal Oscillator Schematics



The crystal must be located as close to the device as possible. Long signal lines may cause too high load to operate the crystal, and cause crosstalk to other parts of the system.

Note: Crystal selection must be done using the “Crystal Data Sheet” and the “Crystal Oscillator Parameters” given in the *XOSC Electrical Specifications* from the “Electrical Characteristics” chapter.

Table 51-4. Crystal Oscillator Checklist

Signal Name	Description	Recommended Pin Connection
XIN	External Crystal Crystal frequency range specified in the XOSC Electrical Specifications (XOSC_1 parameter)	C1 Load Capacitor
XOUT		C2 Load Capacitor

51.6.3 External Slow Clock Source

Figure 51-9. External Slow Clock Source Schematics

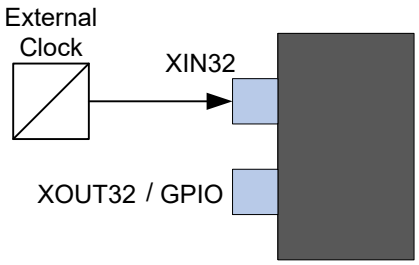


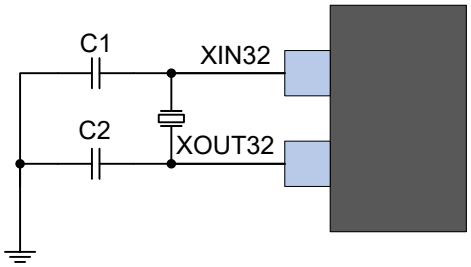
Table 51-5. External Slow Clock Source Connections

Signal Name	Description	Recommended Pin Connection
XIN32	Input for an external clock signal External Clock frequency range specified in the XOSC32K Electrical Specifications (XOSC32_19 parameter)	XIN32 is used as input for an external clock signal
XOUT32/GPIO	NC/GPIO	Available for use as a GPIO

51.6.4 32.768 kHz Crystal Oscillator

PIC32CM LE00/LS00/LS60 32.768 kHz crystal oscillator is optimized for very low-power consumption, hence close attention must be made when selecting crystals.

Figure 51-10. 32.768 kHz Crystal Oscillator Schematics



Note: 32.768 kHz crystal selection must be done using both the crystal data sheet and the crystal oscillator parameters given in the XOSC32K Electrical Specifications from the Electrical Characteristics chapter.

Table 51-6. 32.768 kHz Crystal Oscillator Checklist

Signal Name	Description	Recommended Pin Connection
XIN32	External Crystal Crystal frequency range specified in the XOSC32K Electrical Characteristics (XOSC32_1 parameter)	C1 Load Capacitor
XOUT32		C2 Load Capacitor

51.6.5 XOSC32K Jitter Minimization

⚠ CAUTION The transition time of the following pins must be greater than 50µs in order to not affect the XOSC32 cycle to cycle jitter.

Table 51-7. XOSC32K Jitter Minimization

Package	Pin Name
TQFP48 / VQFN48	PA02, PB03
TQFP64 / VQFN64	PA02, PB03
TQFP100	PC00, PB03

51.7 Programming and Debug

For programming and/or debugging the PIC32CM LE00/LS00/LS60, the device should be connected using the Serial Wire Debug, SWD, interface. Currently the SWD interface is supported by several Microchip and third party programmers and debuggers.

The PIC32CM LE00/LS00/LS60 Curiosity Pro evaluation board supports programming and debugging through the on-board embedded debugger so no external programmer or debugger is needed.

Refer to the related Microchip User Guides for details on debugging and programming connections and options. For connecting to any other programming or debugging tool, refer to that specific programmer or debugger's user guide.

51.7.1 Debugging and Programming Pins

The SWDIO and SWCLK pins are used for In-Circuit programming and debugging purposes. It is recommended to keep the trace length between the debug external connector and the debug pins on the device as short as possible to minimize ESD/EMI vulnerabilities. If the debug external connector is expected to experience an ESD event, a series resistor is recommended, with the value in the range of a few tens of Ohms, not to exceed 100 Ohms with protection using Transient Voltage Suppressors (TVS), at the user's discretion.

Important: SWCLK pin is by default, internally pulled-up after reset.

51.7.2 Cortex Debug Connector (10-pin)

For debuggers and/or programmers that support the Cortex Debug Connector (10-pin) interface the signals should be connected as shown in the following figure, with details described in the following table.

Figure 51-11. Cortex Debug Connector (10-pin)

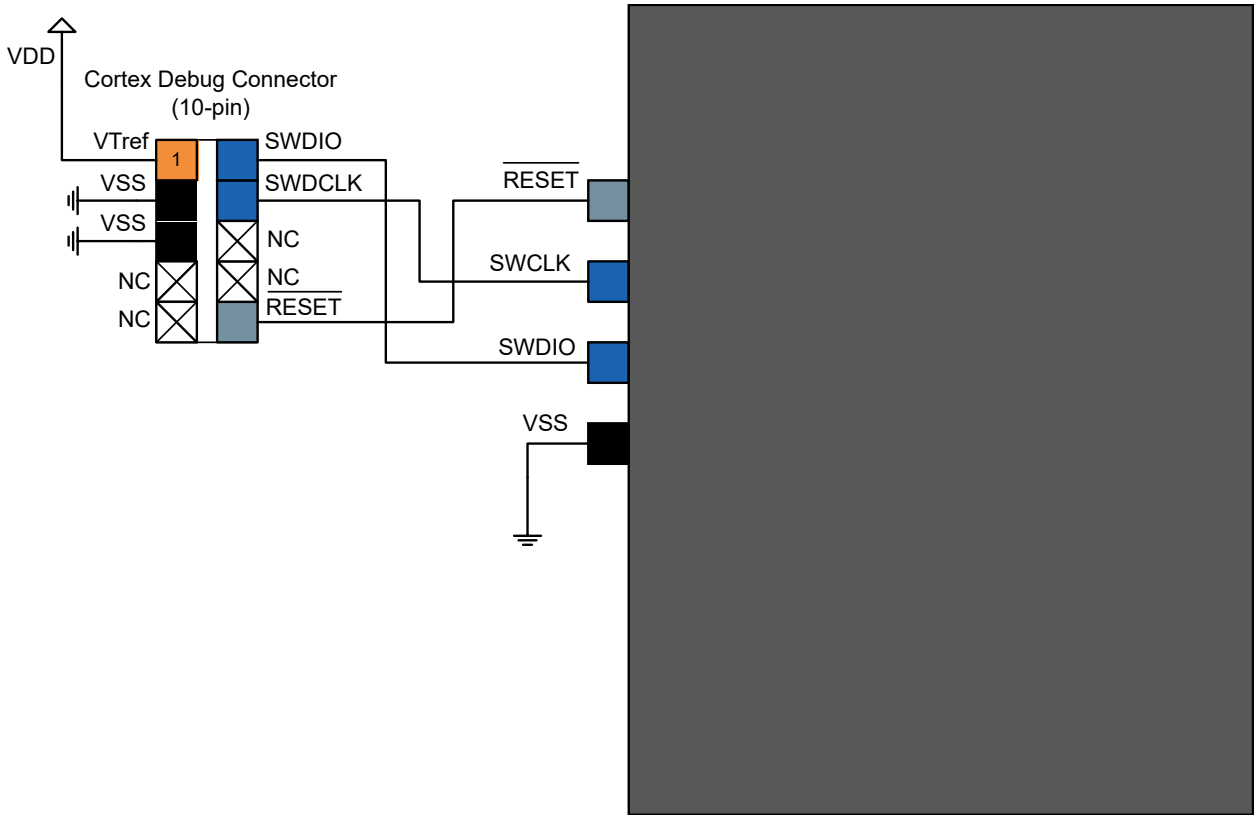


Table 51-8. Cortex Debug Connector (10-pin)

Header Signal Name	Description
SWDCLK	Serial wire clock pin
SWDIO	Serial wire bidirectional data pin
$\overline{\text{RESET}}$	Target device reset pin, active low
VTref	Target voltage sense, should be connected to the device VDD
VSS	Ground

51.7.3 20-pin IDC JTAG Connector

For debuggers and/or programmers that support the 20-pin IDC JTAG Connector, the signals should be connected, as shown in the following figure, with details described in the following table.

Figure 51-12. 20-pin IDC JTAG Connector

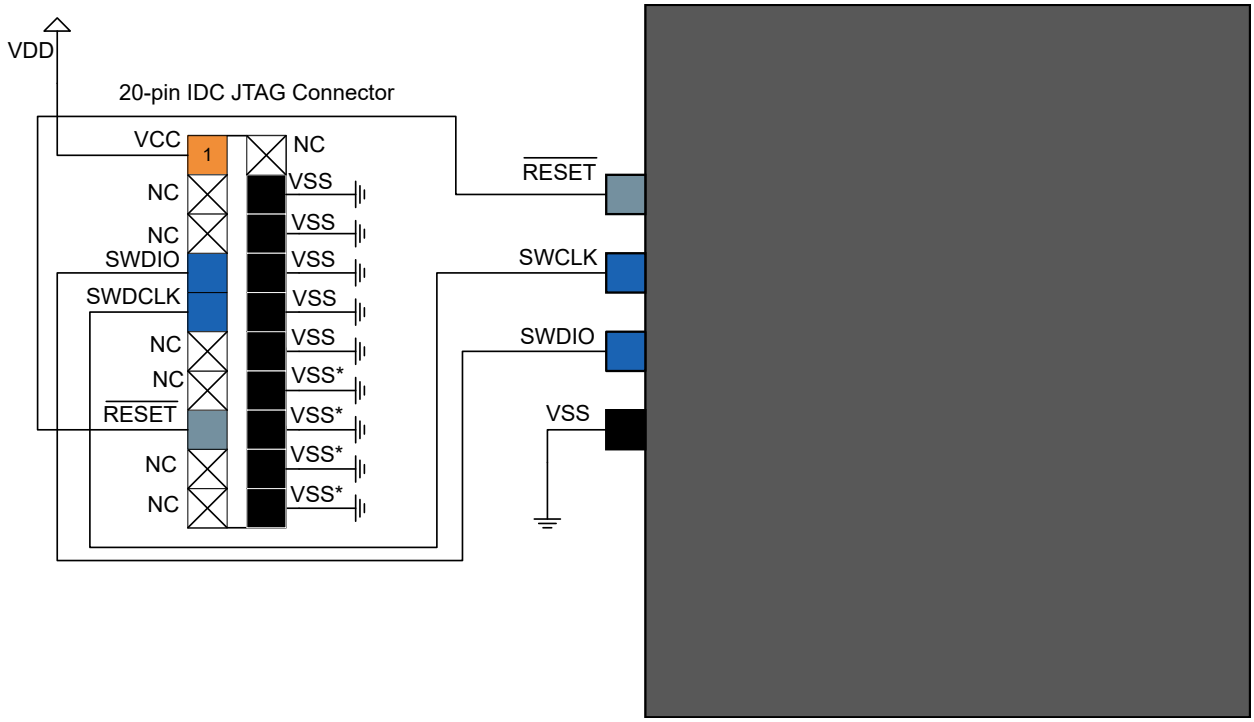


Table 51-9. 20-pin IDC JTAG Connector

Header Signal Name	Description
SWDCLK	Serial wire clock pin
SWDIO	Serial wire bidirectional data pin
$\overline{\text{RESET}}$	Target device reset pin, active low
VCC	Target voltage sense, should be connected to the device VDD
VSS	Ground
VSS*	These pins are reserved for firmware extension purposes. They can be left unconnected or connected to VSS in normal debug environment. They are not essential for SWD in general.

51.8 USB Interface

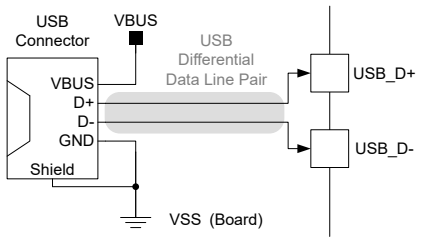
The USB interface consists of a differential data pair (D+/D-).

Refer to the Electrical Characteristics section for operating voltages which will allow USB operation.

Table 51-10. USB Interface Checklist

Signal Name	Recommended Pin Connection	Description
D+	• The impedance of the pair should be matched on the PCB to minimize reflections. • USB differential tracks should be routed with the same characteristics (length, width, number of vias, etc.) • For a tightly coupled differential pair,the signal routing should be as parallel as possible, with a minimum number of angles and vias.	USB full speed / low speed positive data upstream pin
D-		USB full speed / low speed negative data upstream pin

Figure 51-13. Low Cost USB Interface Example Schematic

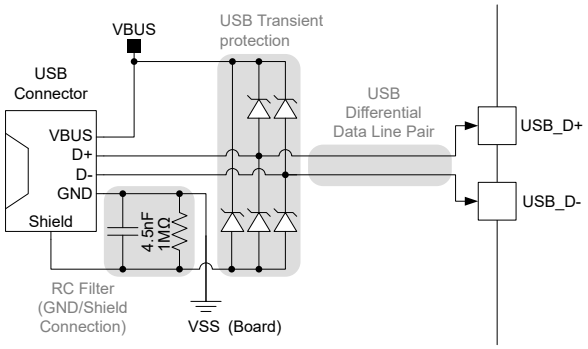


It is recommended to increase ESD protection on the USB D+, D- and VBUS lines using dedicated transient suppressors. These protections should be located as close as possible to the USB connector to reduce the potential discharge path and reduce discharge propagation within the entire system.

The USB FS cable includes a dedicated shield wire that should be connected to the board with caution. Special attention should be paid to the connection between the board ground plane and the shield from the USB connector and the cable.

Tying the shield directly to ground would create a direct path from the ground plane to the shield, turning the USB cable into an antenna. To limit the USB cable antenna effect, it is recommended to connect the shield and ground through an RC filter.

Figure 51-14. Protected USB Interface Example Schematic



51.9 Designing for High-Speed Peripherals

The PIC32C Family of devices have peripherals that operate at frequencies much higher than the typical for an embedded environment.

Due to these high-speed peripheral signals, it is important to consider several factors when designing a product that uses these peripherals, and the PCB on which these components will be placed. Adhering to these recommendations will help achieve the following goals:

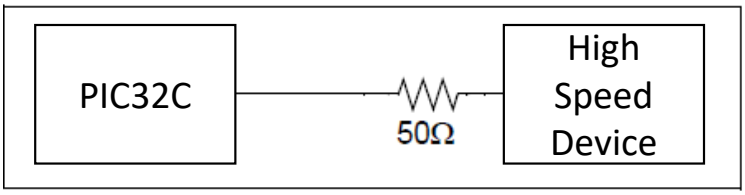
- Minimize the effects of electromagnetic interference for the proper operation of the product.
- Run all PCB high-speed signals first on component side of PCB.
- Ensure signals arrive at their intended destination at the same time by matching critical trace lengths on the PCB.
- Minimize crosstalk. Ensure continuous ground under all high-speed signals.
- Maintain signal integrity by the use of termination resistors in the 30-50 ohm range.
- Reduce system noise by using bulk and high frequency decoupling caps and inductors on power rails.
- Minimize ground bounce and power sag. Use a dedicated ground plane if possible or at a minimum a star ground configuration. Do not daisy chain ground and power traces to components.

51.9.1 System Design

51.9.1.1 Impedance Matching

When selecting parts to place on high-speed signal bus, if the remote I/O pin impedance of the peripheral device does not match the impedance of the pins on the PIC32C device to which it is connected, signal reflections could result, thereby degrading the quality of the signal. If it is not possible to select a product that matches impedance, place a series resistor at the load to create the matching impedance. See the following figure for an example.

Figure 51-15. Series Resistor



51.9.1.2 PCB Layout Recommendations

The following recommendations will help ensure the PCB layout will promote the goals previously listed.

- **Component Placement:**
  - Place bypass capacitors as close to their component power and ground pins as possible, and place them on the same side of the PCB.
  - Devices on the same bus that have larger setup times must be placed closer to the PIC32C family of devices.
- **Power and Ground:**
  - Multi-layer PCBs will allow separate power and ground planes
  - Each ground pin should be connected to the ground plane individually
  - Place bypass capacitor vias as close to the pad as possible (preferably inside the pad)
  - If power and ground planes are not used, maximize width for power and ground traces
  - Use low-ESR, surface-mount bypass capacitors
- **Clocks and Oscillators:**
  - Place crystals as close as possible to the PIC32C Family device XIN/XOUT pins
  - Do not route high-speed signals near the clock or oscillator
  - Avoid via usage and branches in high speed clock lines
  - Place termination resistors at the end of clock lines
- **Traces:**
  - Higher-priority signals must have the shortest traces
  - Avoid long run lengths on parallel traces to reduce coupling
  - Make the clock traces as straight as possible
  - Use rounded turns rather than right-angle turns
  - Have traces on different layers intersect on right angles to minimize crosstalk
  - Maximize the distance between traces, preferably no less than three times the trace width
  - Power traces should be as short and as wide as possible
  - High-speed traces must have a continuous ground beneath them

51.9.1.3 EMI/EMC/EFT (IEC 61000-4-4 and IEC 61000-4-2) Suppression Considerations

The use of LDO regulators is preferred to reduce overall system noise and provide a cleaner power source. However, when utilizing switching Buck/Boost regulators as the local power source for PIC32C devices, as well as in electrically noisy environments or test conditions required for IEC 61000-4-4 and IEC 61000-4-2, users should evaluate the use of Pie-Filters (i.e., L-C) on the power pins, as shown in the Schematic Checklist chapter. In addition to a less noisy power source, use of this type of T-Filter can greatly reduce susceptibility to EMI sources and events. Use Transient Voltage Suppressors (TVS) on power buses as well as on all external PCB signal connections. If design requirements mandate the use of a buck or boost regulator be sure that inductor used is of a shielded type.

51.10 Other Peripherals Considerations

ADC Accuracy
The ADC accuracy may depend on different parameters, such as its input sources, as well as its conversion speed.

Refer to ADC Electrical Specifications from Electrical Characteristics chapter.

PTC I/O Lines
The I/O lines used for analog X-lines and Y-lines must be connected to external capacitive touch sensor electrodes. External components are not required for normal operation.

Refer to PTC Electrical Specifications from Electrical Characteristics chapter.

52. Appendix

52.1 Appendix A: MPLAB® Harmony v3 UART-I²C Factory Bootloader for PIC32CM LE00/LS00/LS60

The Bootloader programmed at production by Microchip is an MPLAB Harmony v3-based Bootloader capable of programming an application binary using either the UART or I²C interface.

Key Features:

- The Bootloader is programmed at the Flash memory location (`0x00000000`) and takes up to 4KB of Flash memory. It is protected by default using the SULCK and BOOTPROT fuse settings from any erase-writes. The BOOTPROT value is set to 4KB (`0x1000`).
- The Bootloader startup code copies the Bootloader from the Flash memory to the SRAM from (`0x20001010`) before calling the main function. The main function (and therefore the entire Bootloader) runs from the SRAM to allow upgrading of the Bootloader code itself in Flash memory. The Bootloader code will unlock the BOOTPROT region before updating if the Bootloader region has to be programmed.
- The Bootloader provides a feature to read the current version of the Bootloader by using the *read version* command
- The Bootloader provides a feature to program the Fuse settings, which are part of the *USER Row* and the *BOCOR Row* using a separate command
- The Bootloader will provide two ways to trigger the Bootloader mode:
  - **External trigger:** Bootloader will be triggered based on the status of a dedicated GPIO pin
  - **Internal trigger:** Bootloader will be triggered based on the trigger pattern written at a specific location in the SRAM by the application firmware
- After a new application image is programmed, the Bootloader will verify the programmed application space by generating a CRC-32 value and comparing it with the CRC-32 received from the Host. The application CRC will not be verified after every reset before jumping to the application space for faster startup.
- The Bootloader will read the first four bytes of application space (`0x1000`) to decide if a valid application is present. If the contents of the first four bytes are not `0xFFFFFFFF`, then the Bootloader assumes a valid application is present and jumps to the application. If a valid application is not present, then the Bootloader will wait, and remain in Bootloader mode.

Key Features for TrustZone variants PIC32CM LS00/LS60:

- The Bootloader is part of the *Secure Flash BOOTPROT (BS)* region with *no BNSC region*.
- The Bootloader will always jump to the *Application Secure Region* at location `0x1000` after programming. Therefore, The applications reset handler must be part of the *Application Secure Region*, which can further call its Non-Secure region if any.

Key Requirements:

- By default the Bootloader expects the application to start from the `0x1000` location. Therefore, the application should be built to start from the `0x1000` Flash location.
- The proper fuse settings required by the application must be sent along with application binary. See Fuse Configurations in this chapter.
- External Pull-ups must be used for the I²C SDA and SCL lines.

Figure 52-1. UART-I²C Bootloader Memory Layout For PIC32CM LE00

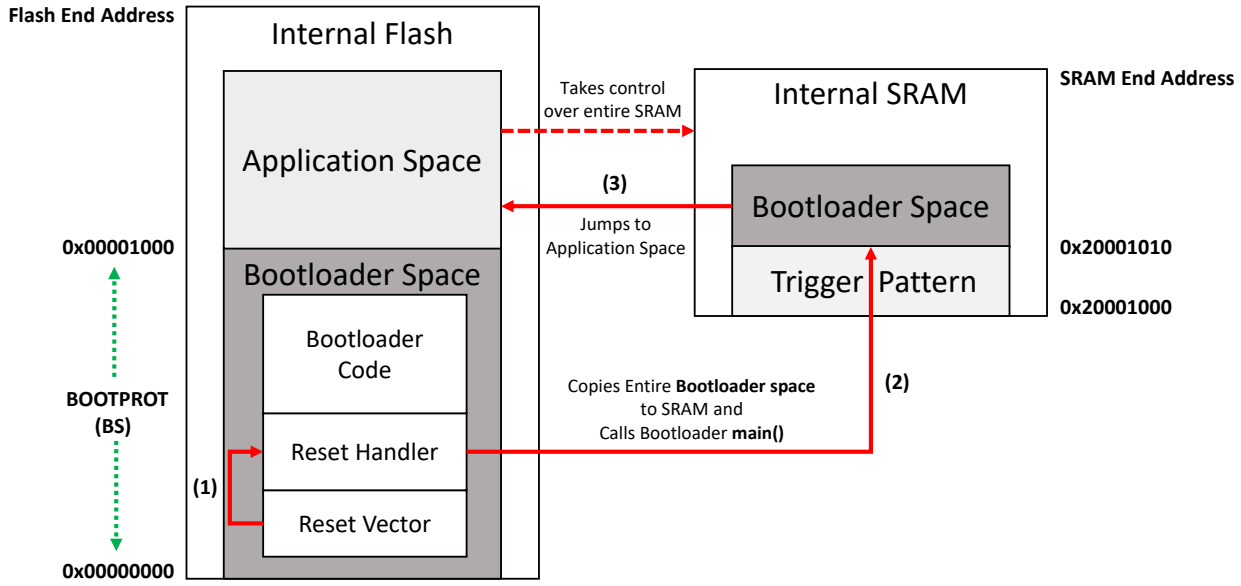


Figure 52-2. UART-I²C Bootloader Memory Layout For PIC32CM LS00/LS60

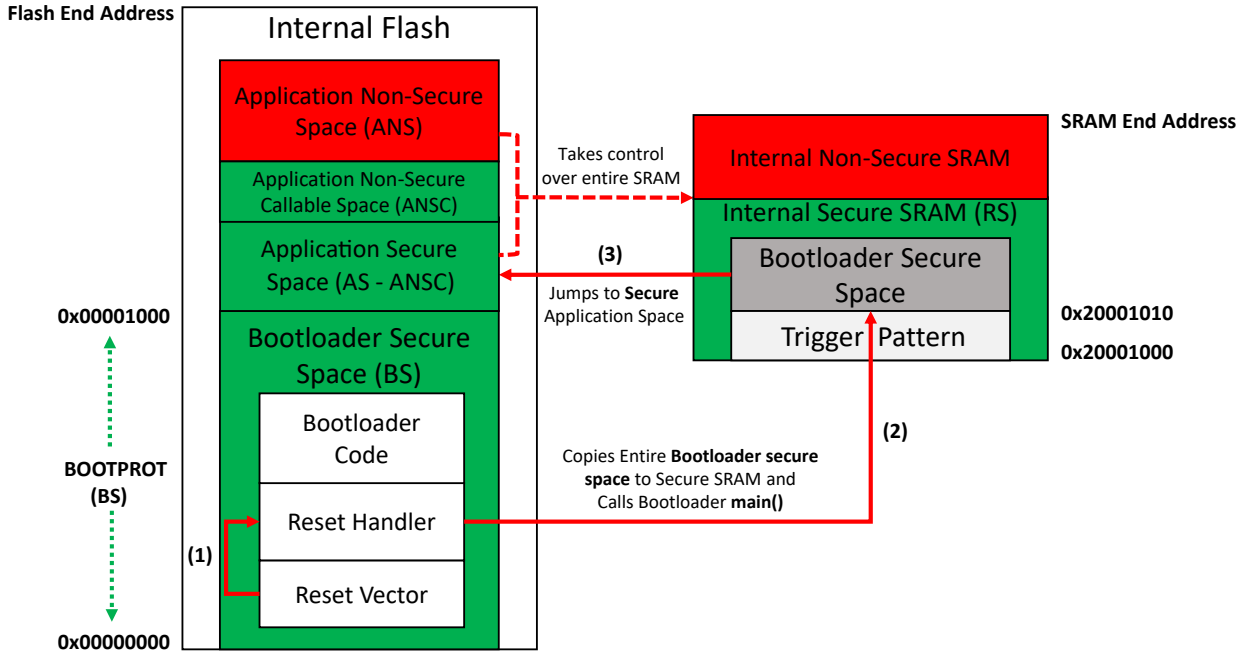
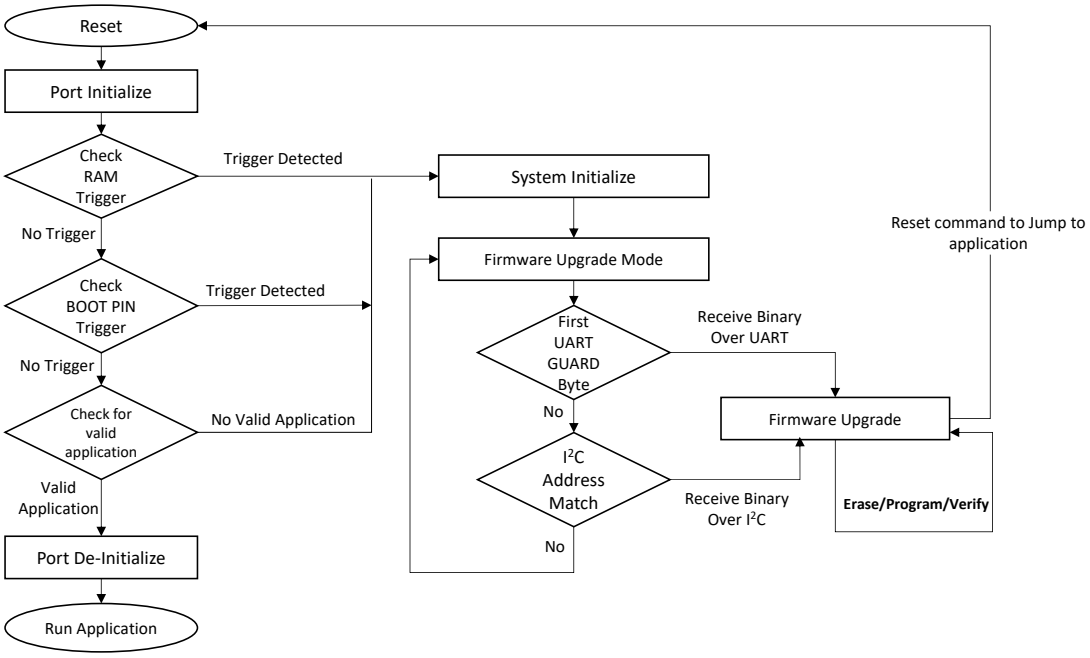


Figure 52-3. UART-I²C Bootloader System Level Execution Flow



52.1.1 UART I²C and Boot Pin Configurations

Table 52-1. Port Pin and SERCOM Instance for UART and I²C

Protocol	Port Pin Configuration	SERCOM Instance
UART	PB02 - TX - SERCOM5/PAD0 PB03 - RX - SERCOM5/PAD1	SERCOM 5 (Can be accessed only in the Secure region for TrustZone variants)
I²C	PA22 - SDA - SERCOM2/PAD0 - External Pull-up required PA23 - SCL - SERCOM2/PAD1 - External Pull-up required	SERCOM 2 (Can be accessed only in the Secure region for TrustZone variants)
Boot pin	PB08 - GPIO - Input - Pull-up enabled (Can be accessed only in the Secure region for TrustZone variants)	-

Table 52-2. UART and I²C Communication Configurations

Protocol	Communication Parameters
UART	115200 baud, 8-bit, Even parity, 1 Stop bit
I²C	Up to 400 kHz speed. I²C client 7-bit address `(0x40)`. Address is fixed and not programmable through any hardware setting.

52.1.2 Bootloader Entry Mechanisms

Default Entry Mechanism:

- The Bootloader will run automatically if there is no valid application firmware. The firmware is considered valid if the first word at the application start address is not `0xFFFFFFFF`.

- Normally this word contains the initial stack pointer value, so it will never be `0xFFFFFFFF` unless the device is erased

Trigger Entry Mechanism:

- **External trigger:**
  - The Boot pin must be used to trigger the Bootloader after reset, refer to the table below.
  - Drive the respective Boot pin to a low state, then reset the device. Upon reset, the Bootloader runs and checks the status of the Boot pin. If the status of the pin is low, the Bootloader enters the firmware upgrade mode.
- **Bootloader Trigger Pattern in SRAM:**
  - The application can trigger the Bootloader by writing a Bootloader trigger pattern with a length of four words (1 word = 4 bytes) to the SRAM location `0x20001000`, then resetting the device. The Bootloader will not be using the initial 4 KB of SRAM as it will be cleared by the BOOTROM at reset. Therefore, the bootloader trigger pattern has to be stored at `0x20001000` of the SRAM.
  - Upon reset, the Bootloader runs and checks the first four words of the SRAM location `0x20001000` for the presence of the Bootloader trigger pattern. If found, the Bootloader enters the firmware upgrade mode.
  - To invoke the Bootloader, the trigger pattern is `0x5048434D`.

Trigger mode	Trigger method
Boot pin	PB08 = Active-low (Internal pull-up is enabled. No hardware or software debouncing is implemented.)

SRAM trigger pattern (`0x20001000`)	`SRAM Word[0] = 0x5048434D`
	`SRAM Word[1] = 0x5048434D`
	`SRAM Word[2] = 0x5048434D`
	`SRAM Word[3] = 0x5048434D`

52.1.3 Fuse Configurations

- The Bootloader has the Fuse settings set to default except below fuse bits
  - The Bootloader region size (4096 Bytes) is specified using the BOOTPROT fuse bit. BOOTPROT = 0x10.
  - The Bootloader region is protected by setting the SULCK configuration at production to 0x6 (SULCK.BS = 0).
- The Fuse settings can be changed in following ways:
  - The Bootloader provides a separate command to program the Fuse Bits received from the host. Refer to the Bootloader protocol documentation and respective host utility documentation for more information.
  - The application can check the Fuse bits during the boot-up, then program any change required using the NVMCTRL peripheral.
- Fuse settings are placed in a higher memory location in the Flash (User Row and BOCOR Row). The Fuse settings need to be disabled for the application project, which will be boot-loaded, as the size of the binary file becomes too large when the fuse settings are enabled.
- When updating the Bootloader itself, make sure that the fuse settings for the Bootloader application are also disabled.
- **For PIC32CM LS00/LS60:**
  - Proper fuse settings required by the application have to be sent along with application binary.
  - Peripherals used by the Bootloader (SERCOM5, SERCOM2, BOOT Pin) should not be configured as Non-Secure while updating the Fuse settings.
  - When WDT is enabled from the Fuse Settings, the bootloader refreshes the WDT at intervals to avoid reset. Therefore, the WDT peripheral should not be configured as Non-Secure if it is enabled through Fuse settings.

52.1.4 Tools and References

Bootloader Source Code
The source code for the UART-I²C Bootloader application is provided as part of the MPLAB Harmony v3 `bootloader_apps_pic32cm_le_ls` package.

Refer to the UART I²C Bootloader for the application project and additional information on using the Bootloader.

UART Host Tool
A python Host utility is provided as part of the MPLAB Harmony v3 Bootloader package which can be used to communicate with the Bootloader to send a binary over the UART from the Host PC.

Refer to the UART Bootloader Host Script Help for additional information on usage of the Host utility.

I²C Embedded Host
The MPLAB Harmony v3-based I²C embedded Host application is provided as a reference which sends a binary over the I²C from a Host microcontroller.

For additional information, refer to the I²C Host App SD-Card Application.

Other References

- Refer to the Bootloader Library for understanding:
  - Bootloader framework
  - How the Bootloader library works
  - Bootloader library configurations
  - Bootloader memory layout
- Refer to the UART Bootloader Protocol for understanding:
  - Details on protocol
  - Various commands supported
- Refer to the I²C Bootloader Protocol for understanding:
  - Details on protocol
  - Various commands supported

53. Conventions

53.1 Numerical Notation

Table 53-1. Numerical Notation

Symbol	Description
165	Decimal number
0b0101	Binary number (example 0b0101 = 5 decimal)
'0101'	Binary numbers are given without prefix if unambiguous
0x3B24	Hexadecimal number
X	Represents an unknown or do not care value
Z	Represents a high-impedance (floating) state for either a signal or a bus

53.2 Memory Size and Type

Table 53-2. Memory Size and Bit Rate

Symbol	Description
KB (kbyte)	kilobyte (2^{10} = 1024)
MB (Mbyte)	megabyte (2^{20} = 1024*1024)
GB (Gbyte)	gigabyte (2^{30} = 1024*1024*1024)
b	bit (binary '0' or '1')
B	byte (8 bits)
1kbit/s	1,000 bit/s rate (not 1,024 bit/s)
1Mbit/s	1,000,000 bit/s rate
1Gbit/s	1,000,000,000 bit/s rate
word	32 bit
half-word	16 bit

53.3 Frequency and Time

Table 53-3. Frequency and Time

Symbol	Description
kHz	1 kHz = 10^3 Hz = 1,000 Hz
MHz	1 MHz = 10^6 Hz = 1,000,000 Hz
GHz	1 GHz = 10^9 Hz = 1,000,000,000 Hz
s	second
ms	millisecond

.....continued	
Symbol	**Description**
µs	microsecond
ns	nanosecond

53.4 Registers and Bits

Table 53-4. Register and Bit Mnemonics

Symbol	Description
R/W	Read/Write accessible register bit. The user can read from and write to this bit.
R	Read-only accessible register bit. The user can only read this bit. Writes will be ignored.
W	Write-only accessible register bit. The user can only write this bit. Reading this bit will return an undefined value.
BIT	Bit names are shown in uppercase. (Example ENABLE)
FIELD[n:m]	A set of bits from bit n down to m. (Example: PINA[3:0] = {PINA3, PINA2, PINA1, PINA0}
Reserved	Reserved bits are unused and reserved for future use. For compatibility with future devices, always write reserved bits to zero when the register is written. Reserved bits will always return zero when read. Reserved bit field values must not be written to a bit field. A reserved value will not be read from a read-only bit field. Do not write any value to reserved bits of a fuse.
PERIPHERAL*i*	If several instances of a peripheral exist, the peripheral name is followed by a number to indicate the number of the instance in the range 0-n. PERIPHERAL0 denotes one specific instance.
Reset	Value of a register after a Power-on Reset. This is also the value of registers in a peripheral after performing a software Reset of the peripheral, except for the Debug Control registers.
SET/CLR	Registers with SET/CLR suffix allows the user to clear and set bits in a register without doing a read-modify-write operation. These registers always come in pairs. Writing a ‘1’ to a bit in the CLR register will clear the corresponding bit in both registers, while writing a ‘1’ to a bit in the SET register will set the corresponding bit in both registers. Both registers will return the same value when read. If both registers are written simultaneously, the write to the CLR register will take precedence.

54. Acronyms and Abbreviations

The below table contains acronyms and abbreviations used in this document.

Table 54-1. Acronyms and Abbreviations

Abbreviation	Description
AC	Analog Comparator
ADC	Analog-to-Digital Converter
ADDR	Address
AES	Advanced Encryption Standard
AHB	Advanced High-performance Bus
AMBA	Advanced Microcontroller Bus Architecture
APB	AMBA Advanced Peripheral Bus
AREF	Analog Reference Voltage
AVDD	Analog Supply Voltage
BOD	Brown-out Detector
CAL	Calibration
CC	Compare/Capture
CCL	Configurable Custom Logic
CLK	Clock
CRC	Cyclic Redundancy Check
CTRL	Control
DAC	Digital-to-Analog Converter
DAP	Debug Access Port
DFLL	Digital Frequency Locked Loop
DPLL	Digital Phase Locked Loop
DMAC	DMA (Direct Memory Access) Controller
DSU	Device Service Unit
EEPROM	Electrically Erasable Programmable Read-Only Memory
EIC	External Interrupt Controller
EVSYS	Event System
FDPLL	Fractional Digital Phase Locked Loop, also DPLL
FREQM	Frequency Meter
GCLK	Generic Clock Controller
GPIO	General Purpose Input/Output
I^2C	Inter-Integrated Circuit
IF	Interrupt Flag
INT	Interrupt

.....continued

Abbreviation	Description
MBIST	Memory Built-In Self-Test
MEM-AP	Memory Access Port
MTB	Micro Trace Buffer
NMI	Non-maskable Interrupt
NVIC	Nested Vector Interrupt Controller
NVM	Nonvolatile Memory
NVMCTRL	Nonvolatile Memory Controller
OPAMP	Operation Amplifier
OSC	Oscillator
PAC	Peripheral Access Controller
PC	Program Counter
PER	Period
PM	Power Manager
POR	Power-on Reset
PORT	I/O Pin Controller
PTC	Peripheral Touch Controller
PWM	Pulse-Width Modulation
RAM	Random-Access Memory
REF	Reference
RTC	Real-Time Counter
RX	Receiver/Receive
SERCOM	Serial Communication Interface
SMBus	System Management Bus
SP	Stack Pointer
SPI	Serial Peripheral Interface
SRAM	Static Random Access Memory
SUPC	Supply Controller
SWD	Serial Wire Debug
TC	Timer/Counter
TRNG	True Random Number Generator
TX	Transmitter/Transmit
ULP	Ultra Low-Power
USART	Universal Synchronous and Asynchronous Serial Receiver and Transmitter
USB	Universal Serial Bus
VDD	Digital Supply Voltage

.....continued

Abbreviation	Description
VSS/AVSS	Ground
VREF	Voltage Reference
WDT	Watchdog Timer
XOSC	Crystal Oscillator

55. Data Sheet Revision History

Note: The data sheet revision is independent of the silicon revision. Refer to the Device Service Unit chapter for detailed information on Device Identification and Revision IDs for your specific device.

Revision F - July 2022
The following changes were incorporated in this revision:

Section	Updates
Configuration Summary	• Updated Table 1-2 to remove deprecated devices • 2. Ordering Information was updated with Tape and Reel in the revision history which was not in the previous revision.
Pinout and Packaging	• Updated the following packages to deprecate LS60 devices from the PIC32CM2532 designation: – 48-pin VQFN and 48-pin TQFP – 64-pin VQFN and 64-pin TQFP – 100-pin TQFP
Memories	• Updated Table 10-1 in Embedded Memories with a new device designation for PIC32CM2532 devices • Updated Table 10-2 in Flash with a new device designation for PIC32CM2532 devices • Updated Table 10-4 in Data Flash with a new device designation for PIC32CM2532 devices • Updated Table 10-6 in SRAM with a new device designation for PIC32CM2532 devices
SERCOM USART	• Added new text to the first paragraph in Start-of-Frame Detection
Electrical Characteristics at 85°C	• Added new sections of data for the PDSW Domain in Retention to the MCU Standby Current Consumption Electrical Specifications table
Schematic Checklist	• Updated the note for Figure 52-6 in External Reset Circuit
Appendix A	• Added New UART-I^{2}C Factory Bootloader Appendix

Revision E - April 2022
The following changes were made to this document:

Section	Updates
Features	• Updated the naming for the ATECC608B-TFLXTLS
1. Configuration Summary	Table 1-1. PIC32CM LE00/LS00/LS60 Family Features
Memories	• Updated the values in the tables for Bit position 13 in: – PIC32CM LE00 User Row – PIC32CM LS00 User Row – PIC32CM LS60 User Row
PIC32CM LS00/LS60 Specific Security Features	• Replaced the text in ATECC608B CryptoAuthentication Device (PIC32CM LS60 only)
Boot ROM	• Added new content to refer to Region Unlock Bits in Chip Erase

.....continued

Section	Updates
IDAU	Updated the following registers: 14.2.2. SCFGB, 14.2.3. SCFGA, and 14.2.4. SCFGR
NVMCTRL	• Added an Important note to Region Unlock Bits • Updated Figurre NVMCTRL - Secure SRAM (RS region) Size • Updated these registers: 29.6.10. SULCK and 29.6.11. NSULCK
SERCOM USART	• Corrected oscillator naming in Start-of-Frame Detection
SERCOM I^2C	• Added a new note to the Signal Description
TRNG	• Removed incorrect NIST References from the following sections: – Overview – Features • Added a new note to Initialization • Added a note to the ENABLE bit for the CTRLA Register
Analog Peripherals Considerations	• Added new bullet points to the Caution note in Analog Peripherals Considerations
OPAMP	• Replaced the Block Diagram with a new Image
TC	Updated the 39.7.2. CTRLBCLR register
TCC	Updated these registers: 40.7.2. CTRLBCLRand 40.7.3. CTRLBSET
Electrical Characteristics at 85°C	• Removed sections of data for the PDSW Domain in Retention from the MCU Standby Current Consumption Electrical Specifications table

Revision D - February 2022
The following changes were made to this document:

Section	Updates
Electrical Characteristics at 85°C	Added a new table for 49.8. MCU Off Current Consumption.
	Updated table 49.2. Absolute Maximum Ratings
	Updated Table 49-5. Thermal Packaging Characteristics (PIC32CM LE00/LS00) and Table 49-6. Thermal Packaging Characteristics (PIC32CM LS60) (

Revision C - February 2022
Along with the changes listed below, numerous typographical updates were done to the document.

Section	Updates
512-KB Flash, 64-KB SRAM with TrustZone, Crypto & Enhanced PTC	• Added new information to the Timers/Output Compare/Input Capture section for PWM Modes using TC and TCC peripherals • Updated anti-tamper verbiage for TrustRAM • Minor reformatting of the features arrangement
Configuration Summary	• Updated the entries for the TC and TCC
PIC32CM LS00/ LS60 Specific Security Features	• Updated the kB information in the figure, PIC32CM LS00/LS60 Default Memories Mapping - 512KB Flash Case in PIC32CM LS00/LS60 Memory Mapping Configuration Summary
Memories	• Updated the Flash topic with new verbiage to enable the cache

.....continued

Section	Updates
MCLK	• Updated Clock Ready Flag with new Bitfield names • Updated the INTFLAG Register with new verbiage for the CKRDY Bitfield
OSCCTRL	• Updated Clock Failure Detection Operation with new verbiage in the Note under Clock Failure Detection
SUPC	• Updated Voltage Regulators with new Active Mode verbiage • Updated the BOD33 Register to display the VREFSEL bitfield name instead of REFSEL
PM	• Made a minor typographical update to the INTFLAG Register
RTC	• Updated Features with new Tamper Detection verbiage • Updated the RTC Block Diagram (Tamper Detection Use Case with four tamper inputs/outputs) with minor editorial updates for Tamper Detection • Made updates throughout the entire chapter changing Active Layer Protection to Active Tamper Detection • Updated the following registers with new notes or other verbiage: – CTRLB in COUNT32 Mode – TAMPCTRL – TAMPCTRLB – CTRLB in COUNT16 Mode – COUNT in COUNT16 Mode – CTRLB in Clock/Calendar Mode
NVMCTRL	• Minor Verbiage update to Features
TRAM	• Updated the Anti-Tamper verbiage in Features
PORT	• Added a new Overview of the PORT Diagram to the Functional Description
EVSYS	• Updated the reset value for the CHANNELn Control Register, and verbiage for the ONDEMAND and RUNSTDBY Bitfields
SERCOM	• Updated Product naming and note verbiage in Features • Updated Product naming in Secure Pin Multiplexing (PIC32CM LS00 Only)
SERCOM USART	• Updated the MAXITER bitfield with new verbiage in the CTRLC Register
SERCOM SPI	• Removed an erroneous note from *Features* • Updated the note in Signal Description on table 36-1
SERCOM I^2C	• Updated the DRDY Bitfield with new verbiage for the Client INTFLAG Register • Updated the Bitfield access properties for the CMD and QCEN bits in the Host CTRLB Register

.....continued

Section	Updates
TC	• Updated the Note in Capture Operations • Renamed Event Capture Action to Event Capture Action on Events or I/Os and updated the topic text, along with minor typographical updates to the Input Capture Timing figure • Renamed Period and Pulse-Width (PPW) Capture Action to Period and Pulse-Width (PPW/PWP) Capture Action on Events and made minor typographical updates to the PWP Capture figure • Renamed Pulse Width Capture Action to Pulse-Width (PW) Capture Action on Events and made minor typographical updates to the Pulse-Width Capture on Channel 0 figure • Renamed Time Stamp Capture to Time-Stamp Capture on Events or I/Os and made minor typographical updates to the Time Stamp figure
TCC	• Updated the Signal Description with new information for WO_NUM-1 • Updated the Waveform Extension Stage Details figure with new Signal naming in Waveform Extension
CCL	• Replaced the Block Diagram with an updated version • Updated the Linked Lut Input Selection figure in Truth Table Inputs Selection
AC	• Minor editorial updates to Features • Updated the Block Diagram with a new figure
ADC	• Updated the verbiage and equation for the SAMPLEN bitfield in the SAMPCTRL Register
DAC	• Added a new note to Events • Updated verbiage for the STARTEI1 and STARTEI0 bitfields in the EVCTRL Register

.....continued

Section	Updates
Electrical Characteristics	• Made numerous typographical updates and minor updates to the units displayed in each table throughout the section • Updated references to REFSEL in Power Supply to VREFSEL • Added a new note to the Peripheral Active Current Electrical Specifications table • Added a XOSC_34 specification to the External XTAL and Clock AC Electrical Specifications table in XOSC Electrical Specifications • Added new 1.8V cases to both USART Electrical Specifications (PL0) and USART Electrical Specifications (PL2) • Updated *I^2S Electrical Specifications (PL2)* with new figures and numerous editorial updates to tables 49-38 and 49-39 • Added a new chapter called *SWD 2-WIRE Electrical Specifications* • The following sections had numerous updates to their tables for min, typ and/or max values – Power Supply – CPU Active Power – CPU Idle Power – CPU Standby Power – CPU Off Current Consumption – Wake-Up Timing Electrical Specifications – I/O Pin Electrical Specifications – Internal Voltage Reference Specifications – XOSC32K Electrical Specifications – OSC16M Electrical Specifications – OPAMP Electrical Specifications – PTC Electrical Specifications – SPI Mode Electrical Specifications (PL0) – SPI Mode Electrical Specifications (PL2) – USB Electrical Specifications
Packaging Information	Updated the 64-pin VQFN package to the most recent version
Schematic Checklist	• Updated the note in External Reset Circuit • Made minor editorial updates to Unused or Unconnected Pins • Removed obsolete information from Clocks and Crystal Oscillators • Updated the table in External Clock Source • Updated the figure with minor edits to the signal names in Crystal Oscillator • Updated the table in External Slow Clock Source • Added a new section called Designing for High-Speed Peripherals

Revision B - November 2020
Along with the changes listed below, numerous typographical updates were done to the document.

The following changes were made to this document:

Section	Updates
512-KB Flash, 64-KB SRAM with TrustZone, Crypto & Enhanced PTC	Updated Core section with a new information for DMIPS

.....continued

Section	Updates
PIC32CM LS00/LS60 Specific Security Features	Added a new topic ATECC608 CryptoAuthentication Device (PIC32CM LS60 only)
Boot ROM	Added a new topic ATECC608x -Based Secure Boot Verification Method
Memories	• Updated the PIC32CM LE00 UROW Bitfields Definition Table with new values for 43:42 and 233:44 • Updated the PIC32CM LS00 UROW Bitfields Definition Table with a new factory setting for Bit position 42 • Updated the PIC32CM LE00 BOCOR Bitfields Definition Table with two new rows for 31:0 and 39:32
PAC	Updated the table for the KEY bit of the WRCTRL Register with the value of SETLOCK for 0x3
DSU	Updated the REVISION Bit of the DID register with information for silicon revisions A0, A1, and B0
OSC32KCTRL	Added a Reserved field to the table for the STARTUP bit in the XOSC32K Register
PM	Updated Sleep Modes with new verbiage
RTC	Updated the COUNT register with a new note
SERCOM SPI	Updated the CTRLB Register for the AMODE bit correcting 2_ADDRS to 2ADDRS
SERCOM I^2C	Updated the CTRLB Register for the AMODE bit correcting 2_ADDRS to 2ADDRS
TCC	Updated the EVCTRL Register with new values in the tables for the CNTSEL and EVACT Bits
CCL	Corrected the INSELx bit of the LUTCTRL Register to read [x=0..2]
AC	Updated the table for the MUXPOS Bit of the COMPCTRL Register with a new reserved value for 0x5-0x7
OPAMP	Updated the RESCTRL Register with a new table value for 0x0 for the RES1MUX Bit

.....continued

Section	Updates
Electrical Characteristics	• Removed VBUS information from the 49.2. Absolute Maximum Ratings • Updated the Characteristics Column of the Operating Frequency vs. Voltage table to read “Param. No.” • Updated the Power Supply Electrical Specifications table with new values for REG_17, REG_19, and REG_21 • Removed USB_17 from the USB Electrical Specifications Table • Added new sub chapter Peripheral Active Current Specifications • The following sections had numerous updates to their tables for min, typ and/or max values – Power Supply – CPU Active Power – CPU Idle Power – CPU Standby Power – CPU Off Current Consumption – Wake-Up Timing Electrical Specifications – I/O Pin Electrical Specifications – Internal Voltage Reference Specifications – XOSC32K Electrical Specifications – OSC16M Electrical Specifications – OPAMP Electrical Specifications – PTC Electrical Specifications – SPI Mode Electrical Specifications (PL0) – SPI Mode Electrical Specifications (PL2) – USB Electrical Specifications
Schematic Checklist	• In Power Supply Connections, Figure 52-1 and Figure 52-2 were corrected with the addition of an underscore to the resistor names • Reworded Note 3 for the Power Supply Connections Table • In External Analog Reference Connections, Figure 52-3 and 52-4 were corrected with the addition of an underscore to the resistor names

Revision A - March 2020
This is the initial released version of this document.

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

Product Identification System

To order or obtain information, e.g., on pricing or delivery, refer to the factory or the listed sales office.

PIC32 CM XXXX LE XX XXX T - I / PT - PROTO

Microchip Brand

Product Family
CM = Entry Level (Cortex-M23)

Memory Size
5164 = 512KB FLASH 64 KB RAM
2532 = 256KB FLASH 32 KB RAM

Key Feature Set
LE = Low Power
LS = Low Power & Security

Family Variant
00 = Standard Device
60 = System in Package (SiP) with ATECC608B CryptoAuthentication™ Device

Recommended for Prototyping only

Package
Y8X = 48-pin TQFP
PT = 64-pin TQFP
PF = 100-pin TQFP
U5B = 48-pin VQFN
5LX = 64-pin VQFN

Temperature Range
I = -40°C to + 85°C (Industrial)

Tape and Reel Flag
T = Tape and Reel
No Character = Tray

Pin Count
048 = 48 pin
064 = 64 pin
100 = 100 pin

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, AnyRate, AVR, AVR logo, AVR Freaks, BesTime, BitCloud, chipKIT, chipKIT logo, CryptoMemory, CryptoRF, dsPIC, FlashFlex, flexPWR, HELDO, IGLOO, JukeBlox, KeeLoq, Kleer, LANCheck, LinkMD, maXStylus, maXTouch, MediaLB, megaAVR, Microsemi, Microsemi logo, MOST, MOST logo, MPLAB, OptoLyzer, PackeTime, PIC, picoPower, PICSTART, PIC32 logo, PolarFire, Prochip Designer, QTouch, SAM-BA, SenGenuity, SpyNIC, SST, SST Logo, SuperFlash, Symmetricom, SyncServer, Tachyon, TempTrackr, TimeSource, tinyAVR, UNI/O, Vectron, and XMEGA are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

APT, ClockWorks, The Embedded Control Solutions Company, EtherSynch, FlashTec, Hyper Speed Control, HyperLight Load, IntelliMOS, Libero, motorBench, mTouch, Powermite 3, Precision Edge, ProASIC, ProASIC Plus, ProASIC Plus logo, Quiet-Wire, SmartFusion, SyncWorld, Temux, TimeCesium, TimeHub, TimePictra, TimeProvider, Vite, WinPath, 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, BlueSky, BodyCom, CodeGuard, CryptoAuthentication, CryptoAutomotive, CryptoCompanion, CryptoController, dsPICDEM, dsPICDEM.net, Dynamic Average Matching, DAM, ECAN, EtherGREEN, In-Circuit Serial Programming, ICSP, INICnet, Inter-Chip Connectivity, JitterBlocker, KleerNet, KleerNet logo, memBrain, Mindi, MiWi, MPASM, MPF, MPLAB Certified logo, MPLIB, MPLINK, MultiTRAK, NetDetach, Omniscient Code Generation, PICDEM, PICDEM.net, PICkit, PICtail, PowerSmart, PureSilicon, QMatrix, REAL ICE, Ripple Blocker, SAM-ICE, Serial Quad I/O, SMART-I.S., SQI, SuperSwitcher, SuperSwitcher II, Total Endurance, TSHARC, USBCheck, VariSense, ViewSpan, WiperLock, Wireless DNA, 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.

All other trademarks mentioned herein are property of their respective companies.

ISBN: 978-1-6683-0797-7

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 - Ra'anana** Tel: 972-9-744-7705 **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