32-bit MCU with TrustZone and Integrated Security

PIC32CK SG01/SG00/GC01/GC00 Family

Product Page Links

Up to 2 MB Live-Update Flash and 512 KB SRAM with Hardware Security Module (HSM) for Secure Connectivity Applications

Operating Conditions

- 1.71V to 3.63V, -40°C to +85°C, DC to 120 MHz
- 1.71V to 3.63V, -40°C to +125°C, DC to 120 MHz

Core: Arm® Cortex®-M33 CPU running at up to 120 MHz

- 4 KB combined instruction-cache and data-cache
- Nested Vector Interrupt Controller (NVIC)
- Stack Limit Checking
- Single Precision Floating Point Unit (FPU)
- DSP Instructions
- TrustZone® for ARMv8-M (optional)
- Embedded Trace Module (ETM) with Instrumentation Trace Macrocell (ITM)
- Trace Port Interface Unit (TPIU)
- 16 secure and 16 non-secure MPU regions

Memories

- 2 MB/1 MB
- 128 KB of Boot Flash Memory (BFM)
- 64 KB of Configuration Flash Memory (CFM)
- Flash Supports:
  - Error Correction Code (ECC)
  - Dual bank with Read-While-Write (RWW) support
  - Write protection
  - In-band error reporting for both read and write accesses
- 512 KB, 256 KB, 128 KB SRAM Main Memory
  - 32 KB or Full SRAM can be retained in Standby mode and Hibernate mode
- Up to 4 KB of Tightly Coupled Memory (TCM)

Security Features

- Hardware Security Module (HSM)
  - AES-128, AES-192, and AES-256: Fully compliant with NIST FIPS 197
    - ECB, CBC, CFB, OFB, CTR, GCM, CCM, XTS, CMAC Modes
  - Triple DES support with up to 168-bit key length
  - HASH/MAC
    - SHA-1, SHA-256, SHA-224, SHA-384, SHA-512, and SHA3 capability
  - ChaCha20-Poly1305 Authenticated Encryption
  - Key Derivation Function (HKDF, KDF2...)
  - Public Key Cryptography: RSA, DSA, and ECC
    - RSA with or without Chinese Remainder Theorem (CRT). Up to 4096-bit key length
    - RSA with/without Chinese Remainder Theorem (CRT). Up to 4096-bit key length
      - Prime Field P-192, P-224, P-256, P-384, P-521
      - Binary Field K-163, K-233, K-283, K-409, K-571
      - Binary Field B-163, B-233, B-283, B-409, B-571
      - P-224, P-256, P-384, and P-521 Elliptic Curve – ECDSA Sign/Verify
    - DSA support up to 2048-bit key length

Security Features (continued...)

- Secure Flash
  - Four 4 KBytes of secure pages
  - Optimized for secrets storage
  - Data Scrambling with user-defined key (optional)
- TrustZone for flexible hardware isolation of memories and peripherals (optional)
  - Up to four regions for the Flash
  - Up to two regions for the SRAM
  - Individual security attribution for each peripheral, I/O, external interrupt line, and Event System Channel
  - Up to three debug access levels
- TrustRAM
  - Address and Data scrambling with user-defined key
  - Chip-level tamper detection on physical RAM to resist microprobing attacks
  - Rapid Tamper Erase on scrambling key and RAM data
  - Silent access for side channel attack resistance
  - Data remanence prevention

System

- Integrated Power-on Reset (POR) and programmable Brown-out Reset (BOR)
- Programmable Low-Voltage Detect Module (LVD)
- 32-channel event system for Inter-peripheral Core-independent Operation
- CRC-32 generator

Advanced Analog Features and Touch

- 12-bit ADC module:
  - 2 Msps with up to 12 external channels and two internal channels
  - Single and differential inputs
  - External reference support
  - Temperature sensor with +-10°C accuracy
- Two analog Comparators with window compare function
- 16 x 16 panel Peripheral Touch controller (PTC)
  - Supports large self-capacitor sensor
  - Support external compensation and integration capacitors

Input/Output

- High-current pins with up to 20 mA source/sink
- Up to 103 programmable I/O lines
- 16 external interrupts (EIC)
- One non-maskable interrupt (NMI)
- One 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

High-Performance Peripherals

- Two DMA instances:
  - One 8-channels instance, and one 4-channels instance respectively
  - Up to 16-words internal FIFO per channel
  - 4 different block transfer modes affecting the bus transfer protocol and speed
  - Built-in CRC
  - Assignable channel priority level (up to 4)
  - Software or hardware DMA triggers
  - Linked list Descriptor Control support
  - Optional timestamp
  - Word/half-word/byte transfer supported
  - Event in/out support
- Up to two Secure Digital Host Controller (SDHC) Interfaces
  - Up to 50 MHz operation
  - 4-bit or 1-bit interface
  - Compatibility with SD and SDHC memory card specification version 3.01
  - Compatibility with SDIO specification version 3.0
  - Compliant with JDEC specification, MMC memory cards V4.51
- One Ethernet MAC
  - 10/100 Mbps in MII and RMII with dedicated DMA
  - IEEE® 1588 Precision Time Protocol (PTP) support
  - IEEE 1588 Time Stamping Unit (TSU) support
  - IEEE 802.3AZ energy efficiency support
  - Support for 802.1AS and 1588 precision clock synchronization protocol
  - Wake on LAN support
- One 16-bit External Bus Interface
  - Supports SRAM, NOR,NAND Flash with on the fly scrambling
  - 8-bit or 16-bit data bus
  - Up to 256 MB memory
- One Full-Speed (12 Mbps) Universal Serial Bus (USB) 2.0 interface
  - Embedded host and device function
  - Eight endpoints
  - On-chip transceiver with integrated serial resistor
  - Support crystal-less operation in device mode
- One USB2.0-compliant High-Speed controller
  - Eight endpoints with 9 KB of dedicated SRAM
  - Low-speed, full-speed and high-speed support
  - Low-power mode (LPM) support
- Parallel Capture Controller (PCC)
  - Up to 14-bit parallel capture mode

Power Management

- Idle mode for fast wake up time
- Standby mode
- Hibernate mode up to full RAM retention
- SleepWalking Peripherals

Timers/Output Compare/Input Capture

- Four 32-bit Timer/Counters for Control (TCC) with extended functions:
  - With 6 compare/capture channels
  - Double buffered compare/capture channel
  - Dead-time Insertion
- Four 16-bit Timer/Counters for Control (TCC) with extended functions:
  - With 2 compare/capture channels
  - Double-buffered compare/capture channel
  - Dead-time Insertion
- 32-bit Real-Time Counter (RTC) with clock/calendar functions
- Watchdog Timer (WDT) with Window mode

Debugger Development Support

- In-circuit and in-application programming/debugging with SWD and JTAG
- Cortex®-M debugger port
- Supports 8 breakpoints and 4 watch points
- IEEE 1149-compatible (JTAG) boundary scan
- Non-intrusive hardware-based instruction trace, Secure Debugging

Software and Tools Support: Develop Prototypes Quickly with A Powerful, Easy-to-Use Ecosystem

- Get code off to a head start with MPLAB Code Configurator
- Graphically configure peripherals, software libraries, and supported RTOS with MPLAB Harmony v3
- Download MPLAB XC Compiler
- Take advantage of MPLAB X IDE's support for 32-bit MCUs
- Select the best debugger for the project: MPLAB ICE, ICD, or PICkit™

Communication Interfaces /Digital Peripherals

- Two CANFD modules
  - Support CAN 2.0 A/B
  - CAN FD ISO 11898-1:2015
- Up to 8 Serial Communication Interfaces (SERCOM), each configurable to operate as:
  - USART with full-duplex and single wire half-duplex configuration
  - I²C up to 3.4 MHz
  - SPI (up to 48 MHz)
  - ISO7816 T = 0 or T = 1 protocols
  - LIN Host/Client
- SQI configurable as additional SPI module (up to 80 MHz)
- Position Decoder (PDEC)
  - Modes: QDEC, HALL, COUNTER
- One Inter-IC Sound Interface (SPI_IXS)
  - 3-wire SPITM (supports all 4 SPI modes)
  - 4-wire Framed SPI modes
  - Audio Codec Protocols (I²S, Left/Right Justified and PCM/DSP, I8s, Am824
  - 4 x 32-bit, 8 x 16-bit or 16 x 8-bit FIFO
  - 2 modules support 32 MHz SCK
  - Modulation for MEMS Microphone

Clock Management

- 4 MHz to 48 MHz Crystal Oscillator (XOSC)
  - Clock failure detection with safe clock switch
- 32.768 kHz ultra low-power crystal oscillator (XOSC32K)
  - Clock failure detection with safe clock switch
- 32.768 kHz ultra low-power internal RC oscillator (OSCULP32K)
- 48 MHz digital Phase-Locked Loop (DFLL48M)
- 1.6 GHz Phase Locked Loop (PLL1G6)
- Frequency meter

Note: Refer to the Chapter Electrical Characteristics (85°C) for specific package limitations.

Table 1. Packages

Type	TQFP				TFBGA
Pin Count	48	64	100	144	144
I/O Pins (up to)	30	47	69	103	103
Contact/Lead Pitch	0.5 mm	0.5 mm	0.5 mm	0.4	0.8
Dimensions	7x7 mm	10x10 mm	14x14 mm	16x16 mm	10x10 mm

Table of Contents

1. Configuration Summary

1. Configuration Summary

Table 1-1. PIC32CK GC (Non-Secure) Device-Specific Feature

Device	Program Memory (KB)	Data Memory (KB)	Pins	Packages	External Bus Interface (EBI)	Ethernet	CAN FD	16-bit Timer Counters (TCC)	32-bit Timer Counters (TCC)	RTC Input/Output	USB HS	USB FS	External Interrupts (EIC)	Parallel Capture Controller (PCC)	I/O Pins	ETM	12-bit ADC pins (single/differential)	Peripheral Touch Driver (PTC)	MPU (regions)	TrustZone for ARMv8-M	Hardware Security Module(HSM)	Secure Boot	TrustRAM (KB)	Trust RAM address/data Scrambling	DMA Channels	Event System Channels	SERCOM	Secure pin Multiplexing (on SERCOM)	Brown Out Detection	Watchdog Timer	SDMMC	QSPI	CCL Look-up Table	PDEC	I2S	Generic Clock (GCLK)	TRNG	Analog Comparator (AC)	Temperature Sensor
PIC32CK1012GC01048	1024	128	48	VQFN	N	Y	1	2	4	2/2	N	Y	13	8-bit	31	N	7/1	15	16	N	N	N	1	Y	12	32	6	N	VDDIO, VDDCORE	Y	1	Y	4	Y	Y	8	Y	2	Y
PIC32CK1012GC01064	1024	128	64	TQFP64/ VQFN64	N	Y	1	2	4	2/2	N	Y	15	8-bit	47	N	10/2	23	16	N	N	N	1	Y	12	32	6	N	VDDIO, VDDCORE	Y	1	Y	4	Y	Y	8	Y	2	Y
PIC32CK1025GC01064	1024	256		TQFP64/ VQFN64	N	Y	2	4	4	2/2	N	Y	15	8-bit	47	N	10/2	24	16	N	N	N	1	Y	12	32	8	N	VDDIO, VDDCORE	Y	2	Y	4	Y	Y	8	Y	2	Y
PIC32CK2051GC01064	2048	512		TQFP64/ VQFN64	N	Y	2	4	4	2/2	N	Y	15	8-bit	47	N	10/2	24	16	N	N	N	1	Y	12	32	8	N	VDDIO, VDDCORE	Y	2	Y	4	Y	Y	8	Y	2	Y
PIC32CK1012GC01100	1024	128	100	TQFP100	N	Y	1	2	4	5/5	N	Y	16	14-bit	69	Y	12/3	32	16	N	N	N	1	Y	12	32	6	N	VDDIO, VDDCORE	Y	1	Y	4	Y	Y	8	Y	2	Y
PIC32CK1025GC01100	1024	256		TQFP100	Y	Y	2	4	4	5/5	Y	Y	16	14-bit	69	Y	12/3	32	16	N	N	N	1	Y	12	32	8	N	VDDIO, VDDCORE	Y	2	Y	4	Y	Y	8	Y	2	Y
PIC32CK2051GC01100	2048	512		TQFP100	Y	Y	2	4	4	5/5	Y	Y	16	14-bit	69	Y	12/3	32	16	N	N	N	1	Y	12	32	8	N	VDDIO, VDDCORE	Y	2	Y	4	Y	Y	8	Y	2	Y
PIC32CK1025GC01144	1024	512	144	TQFP144	Y	Y	2	4	4	8/8	Y	Y	16	14-bit	103	Y	12/3	32	16	N	N	N	1	Y	12	32	8	N	VDDIO, VDDCORE	Y	2	Y	4	Y	Y	8	Y	2	Y
PIC32CK2051GC01144	2048	512	144	TQFP144	Y	Y	2	4	4	8/8	Y	Y	16	14-bit	103	Y	12/3	32	16	N	N	N	1	Y	12	32	8	N	VDDIO, VDDCORE	Y	2	Y	4	Y	Y	8	Y	2	Y
PIC32CK1025GC01144	1024	512	144	TFBGA144	Y	Y	2	4	4	8/8	Y	Y	16	14-bit	103	Y	12/3	32	16	N	N	N	1	Y	12	32	8	N	VDDIO, VDDCORE	Y	2	Y	4	Y	Y	8	Y	2	Y
PIC32CK2051GC01144	2048	512	144	TFBGA144	Y	Y	2	4	4	8/8	Y	Y	16	14-bit	103	Y	12/3	32	16	N	N	N	1	Y	12	32	8	N	VDDIO, VDDCORE	Y	2	Y	4	Y	Y	8	Y	2	Y

Note: These products are subject to availability. Contact a local Microchip sales office for additional information.

Table 1-2. PIC32CK SG (Secure) Device-Specific Features

Device	Program Memory (KB)	Data Memory (KB)	Pins	Packages	External Bus Interface (EBI)	Ethernet	CAN FD	16-bit Timer Counters (TCC)	32-bit Timer Counters (TCC)	RTC Input/Output	USB HS	USB FS	External Interrupts (EIC)	Parallel Capture Controller (PCC)	I/O Pins	ETM	12-bit ADC pins (single/differential)	Peripheral Touch Driver (PTC)	MPU (regions)	TrustZone for ARMv8-M	Hardware Security Module(HSM)	Secure Boot	TrustRAM (KB)	Trust RAM address/data Scrambling	DMA Channels	Event System Channels	SERCOM	Secure pin Multiplexing (on SERCOM)	Brown Out Detection	Watchdog Timer	SDMMC	QSPI	CCL Look-up Table	PDEC	I2S	Generic Clock (GCLK)	TRNG	Analog Comparator (AC)	Temperature Sensor
PIC32CK1012SG01048	1024	128	48	VQFN	N	Y	1	2	4	2/2	N	Y	13	8-bit	31	N	7/1	15	16 sec, 16 nonsec	Y	Y	Y	1	Y	12	32	6	Y	VDDIO, VDDCORE	Y	1	Y	4	Y	Y	8	Y	2	Y
PIC32CK1012SG01064	1024	128		TQFP64/ VQFN64	N	Y	1	2	4	2/2	N	Y	15	8-bit	47	N	10/2	23	16 sec, 16 nonsec	Y	Y	Y	1	Y	12	32	6	Y	VDDIO, VDDCORE	Y	1	Y	4	Y	Y	8	Y	2	Y
PIC32CK1025SG01064	1024	256	64	TQFP64/ VQFN64	N	Y	2	4	4	2/2	N	Y	15	8-bit	47	N	10/2	24	16 sec, 16 nonsec	Y	Y	Y	1	Y	12	32	8	Y	VDDIO, VDDCORE	Y	2	Y	4	Y	Y	8	Y	2	Y
PIC32CK2051SG01064	2048	512		TQFP64/ VQFN64	N	Y	2	4	4	2/2	N	Y	15	8-bit	47	N	10/2	24	16 sec, 16 nonsec	Y	Y	Y	1	Y	12	32	8	Y	VDDIO, VDDCORE	Y	2	Y	4	Y	Y	8	Y	2	Y
PIC32CK1012SG01100	1024	128		TQFP100	N	Y	1	2	4	5/5	N	Y	16	14-bit	69	Y	12/3	32	16 sec, 16 nonsec	Y	Y	Y	1	Y	12	32	6	Y	VDDIO, VDDCORE	Y	1	Y	4	Y	Y	8	Y	2	Y
PIC32CK1025SG01100	1024	256	100	TQFP100	Y	Y	2	4	4	5/5	Y	Y	16	14-bit	69	Y	12/3	32	16 sec, 16 nonsec	Y	Y	Y	1	Y	12	32	8	Y	VDDIO, VDDCORE	Y	2	Y	4	Y	Y	8	Y	2	Y
PIC32CK2051SG01100	2048	512		TQFP100	Y	Y	2	4	4	5/5	Y	Y	16	14-bit	69	Y	12/3	32	16 sec, 16 nonsec	Y	Y	Y	1	Y	12	32	8	Y	VDDIO, VDDCORE	Y	2	Y	4	Y	Y	8	Y	2	Y
PIC32CK1025SG01144	1024	512	144	TQFP144	Y	Y	2	4	4	8/8	Y	Y	16	14-bit	103	Y	12/3	32	16 sec, 16 nonsec	Y	Y	Y	1	Y	12	32	8	Y	VDDIO, VDDCORE	Y	2	Y	4	Y	Y	8	Y	2	Y
PIC32CK2051SG01144	2048	512	144	TQFP144	Y	Y	2	4	4	8/8	Y	Y	16	14-bit	103	Y	12/3	32	16 sec, 16 nonsec	Y	Y	Y	1	Y	12	32	8	Y	VDDIO, VDDCORE	Y	2	Y	4	Y	Y	8	Y	2	Y
PIC32CK1025SG01144	1024	512	144	TFBGA144	Y	Y	2	4	4	8/8	Y	Y	16	14-bit	103	Y	12/3	32	16 sec, 16 nonsec	Y	Y	Y	1	Y	12	32	8	Y	VDDIO, VDDCORE	Y	2	Y	4	Y	Y	8	Y	2	Y
PIC32CK2051SG01144	2048	512	144	TFBGA144	Y	Y	2	4	4	8/8	Y	Y	16	14-bit	103	Y	12/3	32	16 sec, 16 nonsec	Y	Y	Y	1	Y	12	32	8	Y	VDDIO, VDDCORE	Y	2	Y	4	Y	Y	8	Y	2	Y

Note: These products are subject to availability. Contact a local Microchip sales office for additional information.

2. Guidelines for Getting Started

2.1. Basic Connection Requirements

Getting started with the PIC32CK SG/GC Family of 32-bit Microcontrollers (MCUs) requires attention to a minimal set of device pin connections before proceeding with development. The following is a list of pin names, which must be connected always:

- All VDDIO and VSS pins (see Decoupling Capacitors)
- All AVDD and AVSS pins, even if the ADC module is not used (see Decoupling Capacitors)
- $\overline{\text{RESET}}$ pin (see External $\overline{\text{RESET}}$ Pin)
- SWDIO, SWO, SWCLK pins, used for In-Circuit Programming and debugging purposes (see Debug/ Programming Pins)
- XIN and XOUT pins, when external oscillator source is used (see External Oscillator Pins)
- VUSB3V3 (This pin can be grounded when the USB feature is not used)

The following pins may be required:

- The ADC_VREFH pin is used when the external voltage reference for the ADC module is implemented
- USBRBIAS when USBHS is used
- TDI, TMS, TDO, and TCK if JTAG is used

2.2. Decoupling Capacitors

The use of decoupling capacitors on power supply pins, such as VREG, VDD, VSS, AVDD and AVSS is required, see the Schematic Checklist chapter.

Consider the following criteria when using decoupling capacitors:

- **Bulk capacitors:** Must be utilized on all power pins as indicated in the Schematic Checklist chapter.
- **Value and type of capacitor:** 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.

2.2.1. Bulk Capacitors

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, refer to the Schematic Checklist chapter.

2.3. External Reset ($\overline{\text{RESET}}$) Pin

Pulling the $\overline{\text{RESET}}$ pin low generates a device Reset.

For example, as illustrated in the Schematic Checklist chapter, it is recommended that the capacitor C, be isolated from the $\overline{\text{RESET}}$ pin. Place the components illustrated within one-half inch (12 mm) from the $\overline{\text{RESET}}$ pin.

2.4. Debugging or Programming Pins

The SWDIO, SWO, 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 Ωs, not to exceed 100 Ωs with protection using Transient Voltage Suppressors (TVS), at the user's discretion. Refer to the Schematic Checklist chapter.

2.5. JTAG

The TMS, TDO, TDI and TCK pins are used for testing and debugging according to the Joint Test Action Group (JTAG) standard. It is recommended to keep the trace length between the JTAG connector and the JTAG pins on the device as short as possible. If the JTAG connector is expected to experience an ESD event, a series resistor is recommended, with the value in the range of a few tens of Ωs, not to exceed 100 Ωs with protection using Transient Voltage Suppressors (TVS), at the user's discretion. Refer to the Schematic Checklist chapter.

2.6. Trace

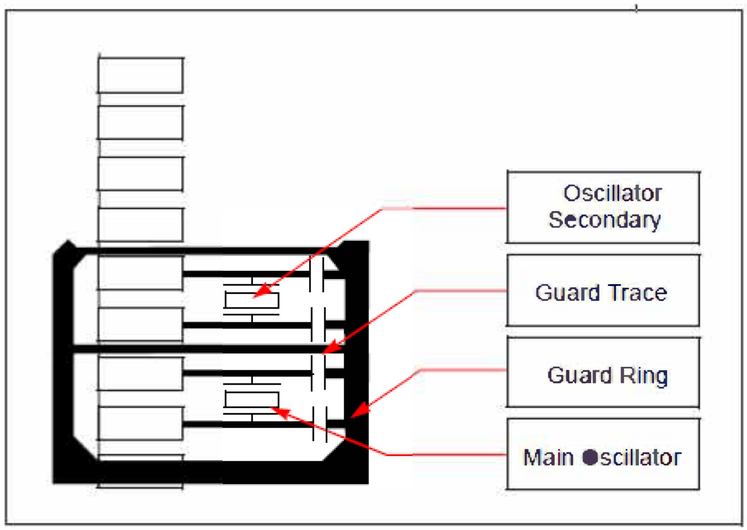
When present on select pin counts, the trace pins can be connected to a hardware trace-enabled programmer/debugger to provide a compressed real-time instruction trace. When used for trace, the TRACE_DATA0, TRACE_DATA1, TRACE_DATA2, TRACE_DATA3 and TRACECLK pins should be dedicated for this use. The trace hardware requires a 22 Ω series resistor between the trace pins and the external trace connector.

2.7. External Oscillator Pins

Many MCUs have options for at least two oscillators: a high-frequency primary oscillator and a low-frequency secondary oscillator.

The oscillator circuit should 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. A suggested layout is illustrated in the following figure.

Figure 2-1. Suggested Oscillator Circuit Placement



2.7.1. Crystal Oscillator Design Consideration

The following hypothetical example assumptions are used to calculate the Primary Oscillator loading capacitor values:

- CIN = XIN Pin Capacitance = 4 pF
- COUT = XOUT Pin Capacitance = 4 pF
- PCB stray capacitance (i.e., 12 mm length) = 2.5 pF
- C1 and C2 = the loading capacitors to use on your crystal circuit design to guarantee that the effective capacitance as seen by the crystal in circuit meets the crystal manufacturer specification MFG Crystal Data Sheet CLOAD spec:

CLOAD = {([CIN + C1] * [COUT + C2]) / [CIN + C1 + C2 + COUT] } + oscillator PCB stray capacitance

Hypothetical Example Crystal Load Capacitor Calculation

Crystal manufacturer data sheet spec example: *CLOAD* = 15 *pF*

Therefore:

MFG CLOAD = {([*CIN* + *C1*] * [*COUT* + *C2*]) / [*CIN* + *C1* + *C2* + *COUT*] } + *estimated oscillator PCB stray capacitance.*

Assuming *C1* = *C2* and PIC32C *Cin* = *Cout*, the formula can be further simplified and restated to solve for *C1* and *C2* by: *C1 = C2 = ((2 * MFG CLoad spec) - Cin - (2 * PCB capacitance)) = ((2 * 15) - 4 - (2 * 2.5 pF)) = (30 - 4 - 5) = 21 pF*

Therefore: *C1 = C2 = 21 pF* is the correct loading capacitors to use on your crystal circuit design to guarantee that the effective capacitance as seen by the crystal in circuit in this example is 15 pF to meet the crystal manufacturer specification.

Note: Do not add excessive gain such that the oscillator signal is clipped, flat on top of the sine wave. If so, you need to reduce the gain accordingly using the . Failure to do so can stress and age the crystal, which can result in an early failure. When measuring the oscillator signal you must use an FET active-powered scope probe with 1 pF or the scope probe itself will unduly change the gain and peak-to-peak levels.

2.8. Unused I/Os

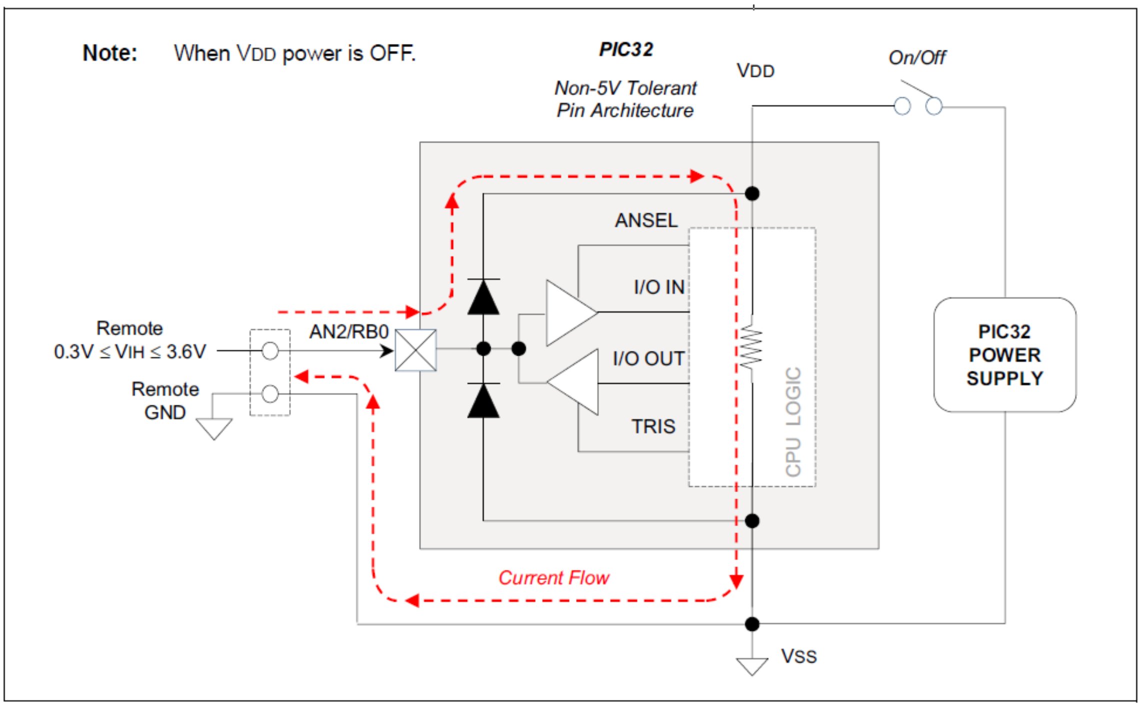
Unused I/O pins should not be allowed to float as inputs. It is recommended that unused inputs be ganged together and connecting through a 1K resistor to digital ground or individually or in multiple groups through 1K as the PCB layout permits. This minimizes the chip vulnerability to ESD and radiated EMI due to a high voltage discharge event.

2.9. Considerations When Interfacing to Remotely Powered Circuits

2.9.1. Non-5V Tolerant Input Pins

A quick review of the absolute maximum rating in the Electrical Characteristics (85°C) will indicate that the voltage on any non-5V tolerant pin may not exceed VDD + 0.4V unless the input current is limited to meet the respective injection current specifications defined by electrical spec parameters DI_19, DI_21, and DI_23 in I/O Pin Electrical Specifications. The following figure shows an example of a remote circuit using an independent power source, which is powered while connected to a PIC32C non-5V tolerant circuit that is not powered.

Figure 2-2. PIC32C Non-5V Tolerant I/O Pin Circuit Example



Without proper signal isolation, on non-5V tolerant pins, the remote signal can power the PIC32C device through the high side ESD protection diodes. Besides violating the absolute maximum rating specification when VDD of the PIC32C device is restored and ramping up or ramping down, it can also negatively affect the internal Power-on Reset (POR) and Brown-out Reset (BOR) circuits, which can lead to improper initialization of internal PIC32C logic circuits. In these cases, it is recommended to implement digital or analog signal isolation as depicted in the following figure, as appropriate or use series I/O pin resistors when possible to limit the injection current to less than 5mA/pin and/or 20mA total for the whole MCU. This is indicative of all industry microcontrollers and not just Microchip products.

Figure 2-3. Example Digital/Analog Signal Isolation Circuits

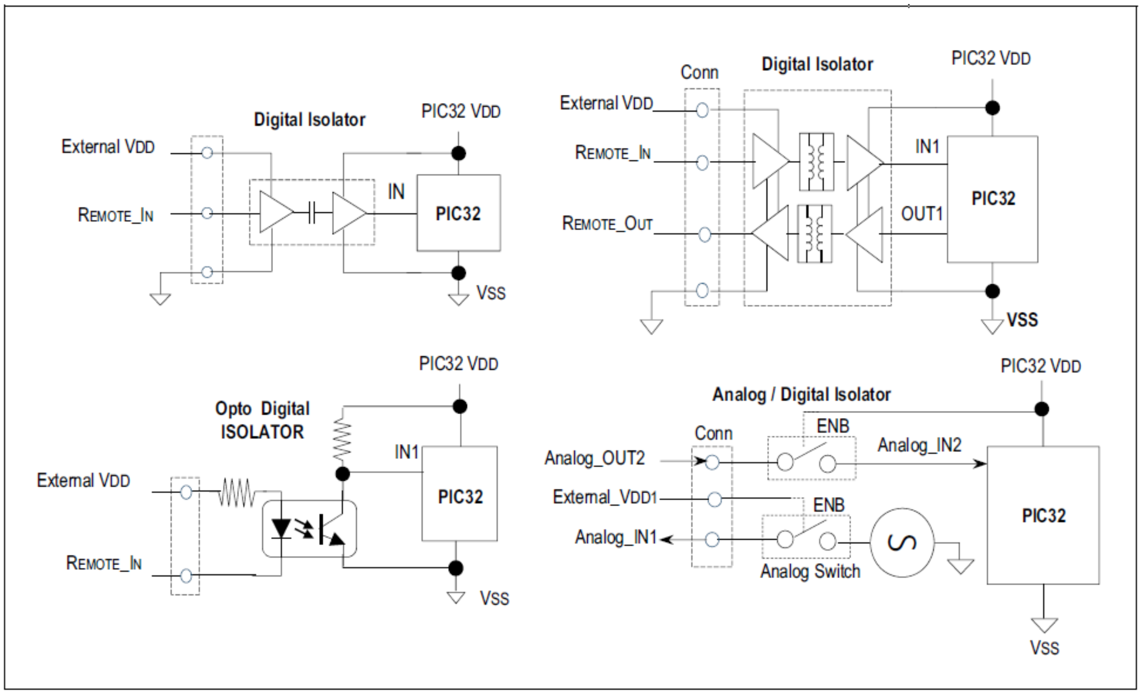


Table 2-1. Examples of Digital/Analog Isolators with Optional Level Translation

Example Digital/ Analog Signal Isolation Circuits	Inductive Coupling	Capacitive Coupling	Opto Coupling	Analog/Digital Switch
ADuM7241 / 40 ARZ (1 Mbps)	X	—	—	—
ADuM7241 / 40 CRZ (25 Mbps)	X	—	—	—
ISO721	—	X	—	—
LTV-829S (2 Channel)	—	—	X	—
LTV-849S (4 Channel)	—	—	X	—
FSA266 / NC7WB66	—	—	—	X

2.10. Designing for High-Speed Peripherals

The PIC32C Family of devices have peripherals that operate at frequencies much higher than typical for an embedded environment. The following list shows the peripherals that produce high-speed signals on their external pins:

Peripheral
USB (Universal Serial Bus)
SDIO
SPI
EBI (External Bus Interface)

Designing for High-Speed Peripherals (continued)

Peripheral
QSPI (Quad SPI)
Ethernet MAC

Due to these high-speed peripheral signals, it is important to consider several factors when designing a product that uses these peripherals, as well as the PCB on which these components will be placed. Adhering to these recommendations will help achieve the following goals:

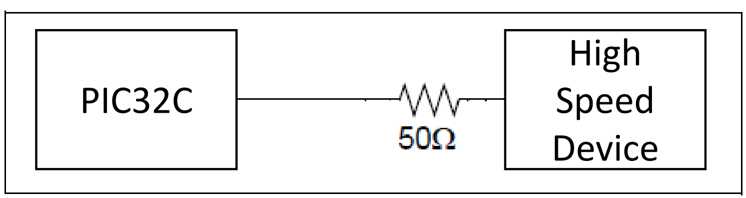
- Minimize the effects of electromagnetic interference for the proper operation of the product
- Run all PCB high-speed signals first on component side of PCB
- Ensure signals arrive at their intended destination at the same time by matching critical trace lengths on the PCB
- Minimize crosstalk. Insure continuous ground under all high-speed signals.
- Maintain signal integrity by the use of termination resistors in the 30-50 Ω 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.

2.10.1. System Design

2.10.1.1. 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 2-4. Series Resistor



2.10.1.2. 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 PIC32MK GPK/MCM with CAN FD 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

2.10.1.3. EMI/EMC/EFT (IEC 61000-4-4 and IEC 61000-4-2) Supression 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 Pi-Filters (i.e., L-C) on the power pins, as shown in the Chapter Power Supplies and Startup Considerations. 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 inductor used is shielded type.

3. Ordering Information

PIC32 CK XXXX SG XX XXX T - I / PT -XXX

Microchip Brand

Product Family
CK = Mid Performance
(Arm Cortex M33)

Memory Size
1025 = 1024KB Flash and 256KB RAM
2051 = 2048KB Flash and 512KB RAM
1012 = 1024KB Flash and 128KB RAM

Key Feature Set
SG = Security (with TrustZone and HSM)
GC = General Purpose (without TrustZone and HSM)

Option Field
01 = Ethernet, CAN
00 = None Present

Pin Count
064 = 64-pin
100 = 100-pin
144 = 144-pin
048 = 48-pin

Pattern
Three-digit QTP, SQTP code
or special requirement
(blank otherwise)

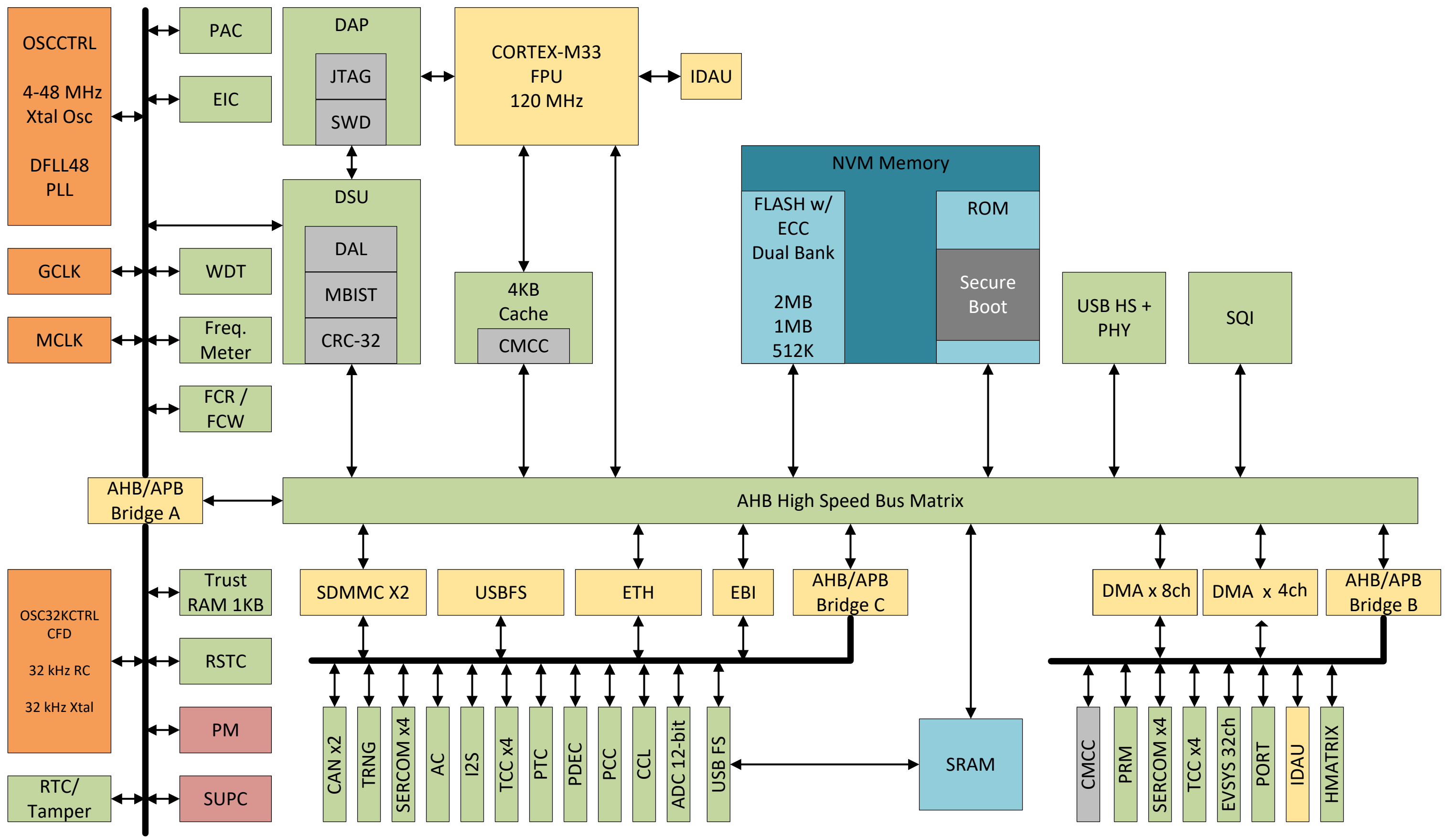
Package
4YW = 48L VQFN
LZX = 64L VQFN
3ZB = 64L TQFP
4SB = 100L TQFP
4KB = 144L TQFP
HQB = 144L TFBGA

Temperature Range
I = -40°C to + 85°C (Industrial)
E = -40°C to + 125°C (Extended)

Tape and Reel Flag
T = Tape and Reel
No Character = Tray/Tube (Default)

4. Block Diagram

Figure 4-1. PIC32CK SG01/SG00/GC01/GC00



5. Package and Pinout

5.1. 48-Pin Packages

5.1.1. 48-Pin VQFN Packages

Table 5-1. 48-Pin VQFN

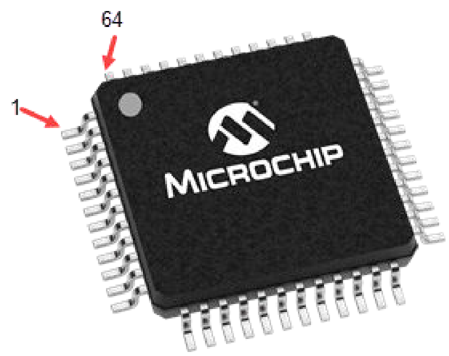
	Top View
PIC32CK1012GC01048 PIC32CK1012SG01048	

Table 5-2. 48-pin VQFN

Pin	SUPC	Priority Function	MUXEN = 0	PMUX Values															XOSC I/O
				MUXEN = 1															
				0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7	0x8	0x9	0xA	0xB	0xC	0xD	0xE	
		RTC Security Tamper	PORT	EIC	ADCN/ADC/AC	SERCOM	SERCOM Alt	TCC	TCC Alt/PDEC	SQI/CAN/USB/CM33 Trace	SDMMC	SPI_IXS	PCC	ETH	MISC	PTC/DEBUG	ETH Alt	SQI Alt	
1	-	-	**PA00**	EIC_EXTINT14	-	SERCOM0_PAD2	SERCOM1_PAD0	TCC2_WO0	-	-	-	-	-	-	CCL_OUT1	BSDAP_TDI	-	-	-
2	-	-	**PA01**	EIC_EXTINT1	-	SERCOM0_PAD3	SERCOM1_PAD1	TCC2_WO3	-	CPU0_SWO	-	-	-	-	GCLK_IO1	BSDAP_TDO	-	-	-
3	-	-	PA02[2]	EIC_EXTINT2	-	-	-	-	-	-	-	-	-	-	-	BSDAP_TMS_SWDIO	-	-	-
4	-	-	PA03[2]	EIC_EXTINT3	-	-	-	-	-	-	-	-	-	-	GCLK_IO0	BSDAP_TCK_SWCLK	-	-	-
5	-	-	RESET_N	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
6	-	-	VSS	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
7	-	-	VDDIO	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
8	-	-	PA06	EIC_EXTINT6	ADC_VINP2/ AC_AIN0	-	SERCOM5_PAD1	TCC3_WO2	TCC3_WO6	-	-	-	-	-	CCL_IN11	-	-	-	-
9	-	-	PA07	EIC_EXTINT7	ADC_VINM2/ ADC_VINP3/ AC_AIN1	-	SERCOM5_PAD2	TCC3_WO3	TCC3_WO7	-	-	-	-	-	CCL_OUT3	-	-	-	-
10	-	-	PA09	EIC_EXTINT9	ADC_VINP5/ AC_AIN0	-	-	-	TCC3_WO0	-	-	SPI_IXS_FSYNC	-	-	GCLK_IO0	PTC_DRV0	-	-	-
11	-	-	PA10	EIC_EXTINT10	ADC_VINP6/ AC_AIN1	-	SERCOM4_PAD2	-	TCC3_WO1	SQI_CS2	SDMMC0_SDCD	-	PCC_DEN1	-	GCLK_IO1	PTC_DRV1	-	-	-
12	-	-	PA11	EIC_EXTINT11	ADC_VINP7	SERCOM5_PAD0	-	TCC4_WO0	TCC3_WO2	-	-	SPI_IXS_SCK	-	ETH_MDC	AC_CMP0	PTC_DRV2	ETH_MDC	-	-
13	-	-	PA12	EIC_EXTINT12	ADC_VINP8	SERCOM5_PAD1	-	TCC4_WO1	TCC3_WO3	-	SDMMC0_SDCD	SPI_IXS_SDI	-	ETH_MDIO	AC_CMP1	PTC_DRV3	ETH_MDIO	-	-
14	-	RTC_OUT1	PB00	EIC_EXTINT0	ADC_VINP9	SERCOM5_PAD2	-	TCC5_WO0	TCC3_WO4	CAN0_TX	SDMMC0_SDWP	SPI_IXS_SDO	-	-	CCL_IN6	PTC_DRV4	-	-	-
15	-	RTC_IN1	PB01	EIC_EXTINT1	ADC_VREFH	SERCOM5_PAD3	-	TCC5_WO1	TCC3_WO5	CAN0_RX	-	-	-	-	CCL_IN7	PTC_DRV5	-	-	-
16	-	-	AVSS	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
17	-	-	PB02	EIC_EXTINT2	-	-	-	-	-	-	-	-	-	-	CCL_IN0	-	-	-	OSC32KCTRL_XIN32
18	-	-	PB03	EIC_EXTINT3	-	-	-	-	-	-	-	-	-	-	CCL_IN1	-	-	-	OSC32KCTRL_XOUT32
19	-	-	AVDD	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
20	-	-	PB04	EIC_EXTINT4	-	SERCOM4_PAD0	SERCOM4_PAD0	TCC0_WO0	PDEC_QDI0	-	-	-	-	-	CCL_IN8	-	-	-	-
21	SUPC_OUT0	RTC_OUT0	PB05	EIC_EXTINT5	-	SERCOM4_PAD1	SERCOM4_PAD1	TCC0_WO1	PDEC_QDI1	-	-	-	-	-	CCL_OUT3	-	-	-	-
22	-	RTC_IN0	PB06	EIC_EXTINT6	-	SERCOM4_PAD3	SERCOM4_PAD3	TCC0_WO3	PDEC_QDI2	-	-	-	-	-	CCL_OUT0	-	-	-	-
23	-	-	VDDIO	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
24	-	-	VDDREG	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
25	-	-	VSS	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
26	-	-	PC00	EIC_EXTINT0	-	SERCOM4_PAD2	SERCOM3_PAD3	-	TCC0_WO1	-	SDMMC0_SDDAT1	-	PCC_DATA7	ETH_TX_CLK/ ETH_REF_CLK	GCLK_IO2	PTC_DRV10	ETH_TX_CLK	SQI_CS1	-
27	-	-	PC01	EIC_EXTINT1	-	SERCOM4_PAD3	SERCOM3_PAD2	-	TCC0_WO2	USB_SOF_1KHZ	SDMMC0_SDDAT2	-	PCC_DATA0	-	GCLK_IO3	PTC_DRV11	ETH_RXD1	SQI_CS0	-

Table 5-2. 48-pin VQFN (continued)

Pin	SUPC	Priority Function	MUXEN = 0	PMUX Values MUXEN = 1															XOSC I/O
				0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7	0x8	0x9	0xA	0xB	0xC	0xD	0xE	
		RTC Security Tamper	PORT	EIC	ADCN/ADC/AC	SERCOM	SERCOM Alt	TCC	TCC Alt/PDEC	SQI/CAN/USB/CM33 Trace	SDMMC	SPI_IXS	PCC	ETH	MISC	PTC/DEBUG	ETH Alt	SQI Alt	
28	-	-	PC02	EIC_EXTINT2	-	SERCOM3_PAD0	SERCOM2_PAD3	-	TCC0_WO3	CAN0_TX	SDMMC0_SDDAT3	-	PCC_DATA1	ETH_MDIO	GCLK_IO4	PTC_DRV12	ETH_RXD0	SQI_DATA0	-
29	-	-	PC03	EIC_EXTINT3	-	SERCOM3_PAD1	-	-	TCC0_WO4	CAN0_RX	SDMMC0_SDWP	-	PCC_DATA2	ETH_MDC	GCLK_IO5	PTC_DRV13	ETH_RXER	SQI_DATA2	-
30	-	-	VDDIO	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
31	-	-	VSS	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
32	-	-	PC05	EIC_EXTINT5	-	SERCOM3_PAD3	SERCOM2_PAD2	-	TCC1_WO0	-	SDMMC0_SDCK	-	PCC_CLK	ETH_RXD2	GCLK_IO7	PTC_DRV15	ETH_TXEN	SQI_DATA3	-
33	-	-	PC06	EIC_EXTINT6	AC_AIN2	SERCOM2_PAD0	SERCOM3_PAD1	-	TCC1_WO1	CAN0_TX	SDMMC0_SDCMD	-	PCC_DATA3	ETH_RXD1	CCL_IN5	PTC_DRV16	ETH_TXD0	SQI_CLK	-
34	-	-	PC07	EIC_EXTINT7	AC_AIN3	SERCOM2_PAD1	SERCOM3_PAD0	-	TCC1_WO2	CAN0_RX	SDMMC0_SDDAT0	-	PCC_DATA4	ETH_RXD0	CCL_OUT2	PTC_DRV17	ETH_TXD1	SQI_DATA1	-
35	-	-	VDDREG	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
36	-	-	PC10	EIC_EXTINT10	-	SERCOM0_PAD0	SERCOM2_PAD1	TCC2_WO2	TCC1_WO5	-	SDMMC0_SDCD	-	PCC_DATA5	ETH_RXDV	CCL_IN3	PTC_DRV20	ETH_RXDV	-	-
37	-	-	PC11	EIC_EXTINT11	-	SERCOM0_PAD1	SERCOM2_PAD0	TCC2_WO1	TCC1_WO6	-	-	-	PCC_DATA6	ETH_TXEN	CCL_IN4	PTC_DRV21	-	-	-
38	-	-	VSS	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
39	-	-	VDDIO	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
40	-	-	PD00	EIC_EXTINT0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	OSCCTRL_ XIN
41	-	-	PD01	EIC_EXTINT1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	OSCCTRL_ XOUT
42	-	-	PD02	EIC_NMI	-	-	-	-	-	USB_SOF_1KHZ	-	-	PCC_DEN2	ETH_TSUCOMP	CCL_IN2	-	ETH_TSUCOMP	-	-
43	-	-	-	-	-	-	-	-	-	USB_DM	-	-	-	-	-	-	-	-	-
44	-	-	-	-	-	-	-	-	-	USB_DP	-	-	-	-	-	-	-	-	-
45	-	-	PD06	EIC_EXTINT6	-	SERCOM0_PAD0	SERCOM1_PAD2	TCC2_WO4	TCC2_WO1	SQI_CS0	SDMMC0_SDCMD	SPI_IXS_FSYNC	-	ETH_MDIO	GCLK_IO4	-	ETH_MDIO	-	-
46	-	-	PD07	EIC_EXTINT7	-	SERCOM0_PAD1	SERCOM1_PAD3	TCC2_WO5	TCC2_WO2	SQI_DATA0	SDMMC0_SDDAT0	SPI_IXS_SDI	-	ETH_MDC	GCLK_IO5	-	ETH_MDC	-	-
47	-	-	VSS	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
48	-	-	VDDIO	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Notes:

1. Pin column items displayed in **BOLD** are 5.5V tolerant pins.
2. PA02 and PA03 are dedicated SWDIO and SWCLK pins. They are not recommended for GPIO use.
3. VDDA and AVDD are used interchangeably.

5.2. 64-Pin Packages

5.2.1. 64-Pin TQFP/VQFN Packages

Table 5-3. 64-Pin TQFP/VQFN Package

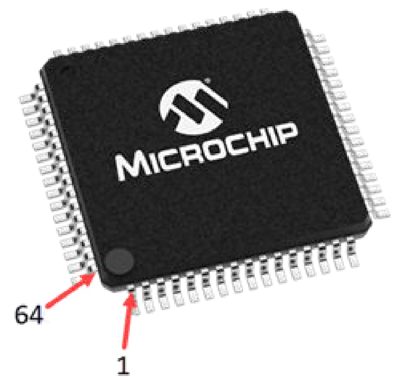
	Top View
PIC32CK1012GC01064 PIC32CK1012SG01064	

Table 5-4. 64-Pin TQFP/VQFN Package

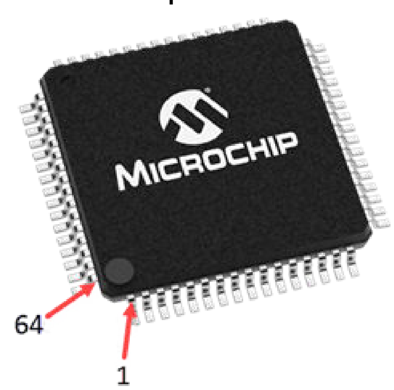
	Top View
PIC32CK(1025/2051)GC01064 PIC32CK(1025/2051)SG01064	

Table 5-5. 64-Pin TQFP/VQFN Package Pinout (PIC32CK1012 Devices)

Pin	SUPC	Priority Function	MUXEN = 0	PMUX Values MUXEN = 1													XOSC I/O
				0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7	0x8	0x9	0xA	0xB	0xC	
		RTC Security Tamper	PORT	EIC	ADCN/ADC/AC	SERCOM	SERCOM Alt	TCC	TCC Alt/PDEC	SQI/CAN/USB/CM33 Trace	SDMMC	SPI_IXS	PCC	ETH	MISC	PTC/DEBUG	
1	-	-	**PA00**	EIC_EXTINT14	-	SERCOM0_PAD2	SERCOM1_PAD0	TCC2_WO0	-	-	-	-	-	-	CCL_OUT1	BSDAP_TDI	-
2	-	-	**PA01**	EIC_EXTINT1	-	SERCOM0_PAD3	SERCOM1_PAD1	TCC2_WO3	-	CPU0_SWO	-	-	-	-	GCLK_IO1	BSDAP_TDO	-
3	-	-	PA02[2]	EIC_EXTINT2	-	-	-	-	-	-	-	-	-	-	-	BSDAP_TMS_SWDIO	-
4	-	-	PA03[2]	EIC_EXTINT3	-	-	-	-	-	-	-	-	-	-	GCLK_IO0	BSDAP_TCK_SWCLK	-
5	-	-	RESET_N	-	-	-	-	-	-	-	-	-	-	-	-	-	-
6	-	-	VSS	-	-	-	-	-	-	-	-	-	-	-	-	-	-
7	-	-	VDDIO	-	-	-	-	-	-	-	-	-	-	-	-	-	-
8	-	-	PA04	EIC_EXTINT4	ADC_VINP0	-	-	TCC3_WO0	-	-	-	-	-	-	CCL_IN9	-	-
9	-	-	PA05	EIC_EXTINT5	ADC_VINM0/ ADC_VINP1	-	SERCOM5_PAD0	TCC3_WO1	-	-	-	-	-	-	CCL_IN10	-	-
10	-	-	PA06	EIC_EXTINT6	ADC_VINP2/ AC_AIN0	-	SERCOM5_PAD1	TCC3_WO2	TCC3_WO6	-	-	-	-	-	CCL_IN11	-	-
11	-	-	PA07	EIC_EXTINT7	ADC_VINM2/ ADC_VINP3/ AC_AIN1	-	SERCOM5_PAD2	TCC3_WO3	TCC3_WO7	-	-	-	-	-	CCL_OUT3	-	-
12	-	-	PA08	EIC_EXTINT8	ADC_VINP4	-	SERCOM5_PAD3	TCC3_WO4	-	-	-	-	-	-	-	-	-
13	-	-	PA09	EIC_EXTINT9	ADC_VINP5/ AC_AIN0	-	-	-	TCC3_WO0	-	-	SPI_IXS_FSYNC	-	-	GCLK_IO0	PTC_DRV0	-
14	-	-	PA10	EIC_EXTINT10	ADC_VINP6/ AC_AIN1	-	SERCOM4_PAD2	-	TCC3_WO1	SQI_CS2	SDMMC0_SDCD	-	PCC_DEN1	-	GCLK_IO1	PTC_DRV1	-
15	-	-	PA11	EIC_EXTINT11	ADC_VINP7	SERCOM5_PAD0	-	TCC4_WO0	TCC3_WO2	-	-	SPI_IXS_SCK	-	ETH_MDC	AC_CMP0	PTC_DRV2	-
16	-	-	PA12	EIC_EXTINT12	ADC_VINP8	SERCOM5_PAD1	-	TCC4_WO1	TCC3_WO3	-	SDMMC0_SDCD	SPI_IXS_SDI	-	ETH_MDIO	AC_CMP1	PTC_DRV3	-
17	-	RTC_OUT1	PB00	EIC_EXTINT0	ADC_VINP9	SERCOM5_PAD2	-	TCC5_WO0	TCC3_WO4	CAN0_TX	SDMMC0_SDWP	SPI_IXS_SDO	-	-	CCL_IN6	PTC_DRV4	-
18	-	RTC_IN1	PB01	EIC_EXTINT1	ADC_VREFH	SERCOM5_PAD3	-	TCC5_WO1	TCC3_WO5	CAN0_RX	-	-	-	-	CCL_IN7	PTC_DRV5	-
19	-	-	AVSS	-	-	-	-	-	-	-	-	-	-	-	-	-	-
20	-	-	PB02	EIC_EXTINT2	-	-	-	-	-	-	-	-	-	-	CCL_IN0	-	OSC32KCTRL_XIN32
21	-	-	PB03	EIC_EXTINT3	-	-	-	-	-	-	-	-	-	-	CCL_IN1	-	OSC32KCTRL_XOUT32
22	-	-	AVDD	-	-	-	-	-	-	-	-	-	-	-	-	-	-
23	-	-	PB04	EIC_EXTINT4	-	SERCOM4_PAD0	SERCOM4_PAD0	TCC0_WO0	PDEC_QDI0	-	-	-	-	-	CCL_IN8	-	-
24	SUPC_OUT0	RTC_OUT0	PB05	EIC_EXTINT5	-	SERCOM4_PAD1	SERCOM4_PAD1	TCC0_WO1	PDEC_QDI1	-	-	-	-	-	CCL_OUT3	-	-
25		RTC_IN0	PB06	EIC_EXTINT6	-	SERCOM4_PAD3	SERCOM4_PAD3	TCC0_WO3	PDEC_QDI2	-	-	-	-	-	CCL_OUT0	-	-
26	-	-	VDDIO	-	-	-	-	-	-	-	-	-	-	-	-	-	-
27	-	-	VSS	-	-	-	-	-	-	-	-	-	-	-	-	-	-
28	-	-	VDDREG	-	-	-	-	-	-	-	-	-	-	-	-	-	-
29	-	-	PB07	EIC_EXTINT7	-	SERCOM3_PAD2	SERCOM4_PAD2	-	PDEC_QDI0	-	SDMMC0_SDCMD	-	PCC_DATA3	-	CCL_IN0	PTC_DRV6	-
30	-	-	PB08	EIC_EXTINT8	-	SERCOM3_PAD3	SERCOM4_PAD3	-	PDEC_QDI1	CAN0_TX	-	-	PCC_DATA4	-	CCL_IN1	PTC_DRV7	-

Table 5-5. 64-Pin TQFP/VQFN Package Pinout (PIC32CK1012 Devices) (continued)

Pin	SUPC	Priority Function	MUXEN = 0	PMUX Values													XOSC I/O
				MUXEN = 1													
				0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7	0x8	0x9	0xA	0xB	0xC	
		RTC Security Tamper	PORT	EIC	ADCN/ADC/AC	SERCOM	SERCOM Alt	TCC	TCC Alt/PDEC	SQI/CAN/USB/CM33 Trace	SDMMC	SPI_IXS	PCC	ETH	MISC	PTC/DEBUG	
31	-	-	PB09	EIC_EXTINT9	-	SERCOM4_PAD0	SERCOM4_PAD1	-	PDEC_QDI2	CAN0_RX	-	-	PCC_DATA5	-	CCL_IN2	PTC_DRV8	-
32	-	-	PB10	EIC_EXTINT10	-	SERCOM4_PAD1	SERCOM4_PAD0	-	TCC0_WO0	-	SDMMC0_SDDAT0	-	PCC_DATA6	ETH_MDC	CCL_OUT0	PTC_DRV9	-
33	-	-	PC00	EIC_EXTINT0	-	SERCOM4_PAD2	SERCOM3_PAD3	-	TCC0_WO1	-	SDMMC0_SDDAT1	-	PCC_DATA7	ETH_TX_CLK/ ETH_REF_CLK	GCLK_IO2	PTC_DRV10	-
34	-	-	PC01	EIC_EXTINT1	-	SERCOM4_PAD3	SERCOM3_PAD2	-	TCC0_WO2	USB_SOF_1KHZ	SDMMC0_SDDAT2	-	PCC_DATA0	-	GCLK_IO3	PTC_DRV11	-
35	-	-	PC02	EIC_EXTINT2	-	SERCOM3_PAD0	SERCOM2_PAD3	-	TCC0_WO3	CAN0_TX	SDMMC0_SDDAT3	-	PCC_DATA1	ETH_MDIO	GCLK_IO4	PTC_DRV12	-
36	-	-	PC03	EIC_EXTINT3	-	SERCOM3_PAD1	-	-	TCC0_WO4	CAN0_RX	SDMMC0_SDWP	-	PCC_DATA2	ETH_MDC	GCLK_IO5	PTC_DRV13	-
37	-	-	PC04	EIC_EXTINT4	-	SERCOM3_PAD2	-	-	TCC0_WO5	-	SDMMC0_SDCD	-	PCC_DEN1	ETH_MDIO	GCLK_IO6	PTC_DRV14	-
38	-	-	VDDIO	-	-	-	-	-	-	-	-	-	-	-	-	-	-
39	-	-	VSS	-	-	-	-	-	-	-	-	-	-	-	-	-	-
40	-	-	PC05	EIC_EXTINT5	-	SERCOM3_PAD3	SERCOM2_PAD2	-	TCC1_WO0	-	SDMMC0_SDCK	-	PCC_CLK	ETH_RXD2	GCLK_IO7	PTC_DRV15	-
41	-	-	PC06	EIC_EXTINT6	AC_AIN2	SERCOM2_PAD0	SERCOM3_PAD1	-	TCC1_WO1	CAN0_TX	SDMMC0_SDCMD	-	PCC_DATA3	ETH_RXD1	CCL_IN5	PTC_DRV16	-
42	-	-	PC07	EIC_EXTINT7	AC_AIN3	SERCOM2_PAD1	SERCOM3_PAD0	-	TCC1_WO2	CAN0_RX	SDMMC0_SDDAT0	-	PCC_DATA4	ETH_RXD0	CCL_OUT2	PTC_DRV17	-
43	-	-	VDDREG	-	-	-	-	-	-	-	-	-	-	-	-	-	-
44	-	-	PC09	EIC_EXTINT9	-	SERCOM2_PAD2	SERCOM3_PAD2	-	TCC1_WO3	-	-	-	-	ETH_RXER	AC_CMP0	PTC_DRV19	-
45	-	-	PC10	EIC_EXTINT10	-	SERCOM0_PAD0	SERCOM2_PAD1	TCC2_WO2	TCC1_WO5	-	SDMMC0_SDCD	-	PCC_DATA5	ETH_RXDV	CCL_IN3	PTC_DRV20	-
46	-	-	PC11	EIC_EXTINT11	-	SERCOM0_PAD1	SERCOM2_PAD0	TCC2_WO1	TCC1_WO6	-	-	-	PCC_DATA6	ETH_TXEN	CCL_IN4	PTC_DRV21	-
47	-	-	PC12	EIC_EXTINT12	-	SERCOM0_PAD2	SERCOM2_PAD2	-	TCC1_WO7	-	-	-	-	ETH_TXD0	CCL_IN5	PTC_DRV22	-
48	-	-	PC13	EIC_EXTINT13	-	SERCOM0_PAD3	SERCOM2_PAD3	-	-	-	-	-	-	ETH_TXD1	CCL_OUT1	PTC_DRV18	-
49	-	-	VSS	-	-	-	-	-	-	-	-	-	-	-	-	-	-
50	-	-	VDDIO	-	-	-	-	-	-	-	-	-	-	-	-	-	-
51	-	-	PD00	EIC_EXTINT0	-	-	-	-	-	-	-	-	-	-	-	-	OSCCTRL_XIN
52	-	-	PD01	EIC_EXTINT1	-	-	-	-	-	-	-	-	-	-	-	-	OSCCTRL_XOUT
53	-	-	PD02	EIC_NMI	-	-	-	-	-	USB_SOF_1KHZ	-	-	PCC_DEN2	ETH_TSUCOMP	CCL_IN2	-	-
54	-	-	-	-	-	-	-	-	-	USB_DM	-	-	-	-	-	-	-
55	-	-	-	-	-	-	-	-	-	USB_DP	-	-	-	-	-	-	-
56	-	-	PD05	EIC_EXTINT5	-	SERCOM0_PAD2	-	-	TCC2_WO0	SQI_CS1	SDMMC0_SDWP	-	-	-	GCLK_IO3	-	-
57	-	-	PD06	EIC_EXTINT6	-	SERCOM0_PAD0	SERCOM1_PAD2	TCC2_WO4	TCC2_WO1	SQI_CS0	SDMMC0_SDCMD	SPI_IXS_FSYNC	-	ETH_MDIO	GCLK_IO4	-	-
58	-	-	PD07	EIC_EXTINT7	-	SERCOM0_PAD1	SERCOM1_PAD3	TCC2_WO5	TCC2_WO2	SQI_DATA0	SDMMC0_SDDAT0	SPI_IXS_SDI	-	ETH_MDC	GCLK_IO5	-	-
59	-	-	VSS	-	-	-	-	-	-	-	-	-	-	-	-	-	-
60	-	-	VDDIO	-	-	-	-	-	-	-	-	-	-	-	-	-	-
61	-	-	PD08	EIC_EXTINT8	-	SERCOM1_PAD0	-	-	TCC2_WO3	SQI_DATA2	SDMMC0_SDDAT2	SPI_IXS_SDO	-	-	CCL_IN6	-	-
62	-	-	PD09	EIC_EXTINT9	-	SERCOM1_PAD1	-	-	TCC2_WO4	SQI_DATA3	SDMMC0_SDDAT3	-	-	-	CCL_IN7	-	-
63	-	-	PD10	EIC_EXTINT10	-	SERCOM1_PAD2	-	-	TCC2_WO5	SQI_CLK	SDMMC0_SDCK	SPI_IXS_SCK	-	-	CCL_IN8	-	-
64	-	-	PD11	EIC_EXTINT11	-	SERCOM1_PAD3	-	TCC2_WO2	TCC2_WO6	SQI_DATA1	SDMMC0_SDDAT1	-	-	-	CCL_OUT2	-	-

Table 5-6. 64-Pin TQFP/VQFN Package Pinout (PIC32CK1025/2051 Devices)

Pin	Priority Function	MUXEN=0	MUXEN=1 PMUX Values 0x0	0x1	0x2	0x3	0x5	0x6	0x7	0x8	0x9	0xa	0xb	0xc	0xd	External Oscillator I/O
	RTC Security Tamper Input Loop(s)	Port	EIC	ADCN/ADC/AC	SERCOM	SERCOM Alt	TCC	TCC Alt/PDEC	SQI/CAN/USB/CM33 Trace	SDMMC	SPI_IXS	PCC	ETH	MISC	PTC/Debug	
1	-	**PA00**	EIC_EXTINT14	-	-	SERCOM6_PAD0	TCC2_WO0	-	-	-	-	-	-	-	BSDAP_TDI	-
2	-	**PA01**	EIC_EXTINT1	-	SERCOM7_PAD0	SERCOM6_PAD1	TCC2_WO3	-	CPU0_SWO	-	-	-	-	GCLK_IO1	BSDAP_TDO	-
3	-	PA02 [2]	-	-	-	-	-	-	-	-	-	-	-	-	BSDAP_TMS_SWDIO	-
4	-	PA03 [2]	-	-	-	-	-	-	-	-	-	-	-	-	BSDAP_TCK_SWCLK	-
5	-	RESET_N	-	-	-	-	-	-	-	-	-	-	-	-	-	-
6	-	VSS	-	-	-	-	-	-	-	-	-	-	-	-	-	-
7	-	VDDIO	-	-	-	-	-	-	-	-	-	-	-	-	-	-
8	-	PA04	EIC_EXTINT4	ADC_VINP0	SERCOM7_PAD1	-	TCC3_WO0	-	-	-	-	-	-	LUT3_IN0	-	-
9	-	PA05	EIC_EXTINT5	ADC_VINM0/ ADC_VINP1	SERCOM7_PAD2	SERCOM5_PAD0	TCC3_WO1	-	-	-	-	-	-	LUT3_IN1	-	-
10	-	PA06	EIC_EXTINT6	AC_AIN0/ ADC_VINP2	SERCOM7_PAD3	SERCOM5_PAD1	TCC3_WO2	-	-	-	-	-	-	LUT3_IN2	-	-
11	-	PA07	EIC_EXTINT7	AC_AIN1/ ADC_VINM2/ ADC_VINP3	SERCOM6_PAD0	SERCOM5_PAD2	TCC3_WO3	-	-	-	-	-	-	LUT3_OUT	-	-
12	-	PA08	EIC_EXTINT8	ADC_VINP4	SERCOM6_PAD1	SERCOM5_PAD3	TCC3_WO4	-	-	-	-	-	-	-	-	-
13	-	PA09	EIC_EXTINT9	ADC_VINP5	SERCOM6_PAD2	SERCOM6_PAD3	-	TCC3_WO0	-	-	SPI_IXS_FSYNC	-	-	GCLK_IO0	PTC_DRV0	-
14	-	PA10	EIC_EXTINT10	ADC_VINP6	SERCOM6_PAD3	-	-	TCC3_WO1	SQI_CS2	SDMMC0_SDCD	-	PCC_DEN1	-	GCLK_IO1	PTC_DRV1	-
15	-	PA11	EIC_EXTINT11	ADC_VINP7	SERCOM5_PAD0	SERCOM7_PAD1	TCC4_WO0	TCC3_WO2	-	-	SPI_IXS_SCK	-	ETH_MDC	AC_CMP0	PTC_DRV2	-
16	-	PA12	EIC_EXTINT12	ADC_VINP8	SERCOM5_PAD1	SERCOM7_PAD0	TCC4_WO1	TCC3_WO3	-	SDMMC0_SDCD	SPI_IXS_SDI	-	ETH_MDIO	AC_CMP1	PTC_DRV3	-
17	RTC_OUT1	PB00	EIC_EXTINT0	ADC_VINP9	SERCOM5_PAD2	SERCOM7_PAD2	TCC5_WO0	TCC3_WO4	CAN1_TX	SDMMC0_SDWP	SPI_IXS_SDO	-	-	-	PTC_DRV4	-
18	RTC_IN1	PB01	EIC_EXTINT1	ADC_VREFH	SERCOM5_PAD3	SERCOM7_PAD3	TCC5_WO1	TCC3_WO5	CAN1_RX	-	-	-	-	-	PTC_DRV5	-
19	-	AVSS	-	-	-	-	-	-	-	-	-	-	-	-	-	-
20	-	PB02	EIC_EXTINT2	-	-	-	-	-	-	-	-	-	-	-	-	OSC32KCTRL_XIN32
21	-	PB03	EIC_EXTINT3	-	-	-	-	-	-	-	-	-	-	-	-	OSC32KCTRL_XOUT32
22	-	AVDD	-	-	-	-	-	-	-	-	-	-	-	-	-	-
23	-	PB04	EIC_EXTINT4	-	SERCOM4_PAD0	-	TCC0_WO0	PDEC_QDI0	-	-	-	-	-	-	-	-
24	RTC_OUT0	PB05	EIC_EXTINT5	-	SERCOM4_PAD1	-	TCC0_WO1	PDEC_QDI1	-	-	-	-	-	-	-	-
25	RTC_IN0	PB06	EIC_EXTINT6	-	SERCOM4_PAD3	-	TCC0_WO3	-	-	-	-	-	-	-	-	-
26	-	VDDIO	-	-	-	-	-	-	-	-	-	-	-	-	-	-
27	-	VSS	-	-	-	-	-	-	-	-	-	-	-	-	-	-
28	-	VDDREG	-	-	-	-	-	-	-	-	-	-	-	-	-	-
29	-	PB07	EIC_EXTINT7	-	SERCOM3_PAD2	SERCOM4_PAD2	TCC7_WO0	PDEC_QDI0	-	SDMMC1_SDCMD	-	PCC_DATA3	-	LUT0_IN0	PTC_DRV6	-
30	-	PB08	EIC_EXTINT8	-	SERCOM3_PAD3	SERCOM4_PAD3	TCC7_WO1	PDEC_QDI1	CAN0_TX	-	-	PCC_DATA4	-	LUT0_IN1	PTC_DRV7	-

Table 5-6. 64-Pin TQFP/VQFN Package Pinout (PIC32CK1025/2051 Devices) (continued)

Pin	Priority Function	MUXEN=0	MUXEN=1													External Oscillator I/O
			PMUX Values													
			0x0	0x1	0x2	0x3	0x5	0x6	0x7	0x8	0x9	0xa	0xb	0xc	0xd	
	RTC Security Tamper Input Loop(s)	Port	EIC	ADCN/ADC/AC	SERCOM	SERCOM Alt	TCC	TCC Alt/PDEC	SQI/CAN/USB/CM33 Trace	SDMMC	SPI_IXS	PCC	ETH	MISC	PTC/Debug	
31	-	PB09	EIC_EXTINT9	-	SERCOM4_PAD0	SERCOM4_PAD1	-	PDEC_QDI2	CAN0_RX	-	-	PCC_DATA5	-	LUT0_IN2	PTC_DRV8	-
32	-	PB10	EIC_EXTINT10	-	SERCOM4_PAD1	SERCOM4_PAD0	-	TCC0_WO0	-	SDMMC1_SDDAT0	-	PCC_DATA6	ETH_MDC	LUT0_OUT	PTC_DRV9	-
33	-	PC00	EIC_EXTINT0	-	SERCOM4_PAD2	-	-	TCC0_WO1	-	SDMMC1_SDDAT1	-	PCC_DATA7	ETH_REF_CLK/ ETH_TX_CLK	GCLK_IO2	PTC_DRV10	-
34	-	PC01	EIC_EXTINT1	-	SERCOM4_PAD3	-	TCC6_WO0	TCC0_WO2	USB_SOF_1KHZ	SDMMC1_SDDAT2	-	PCC_DATA0	-	GCLK_IO3	PTC_DRV11	-
35	-	PC02	EIC_EXTINT2	-	SERCOM3_PAD0	-	TCC6_WO1	TCC0_WO3	CAN0_TX	SDMMC1_SDDAT3	-	PCC_DATA1	ETH_MDIO	GCLK_IO4	PTC_DRV12	-
36	-	PC03	EIC_EXTINT3	-	SERCOM3_PAD1	-	TCC7_WO0	TCC0_WO4	CAN0_RX	SDMMC1_SDWP	-	PCC_DATA2	ETH_MDC	GCLK_IO5	PTC_DRV13	-
37	-	PC04	EIC_EXTINT4	-	SERCOM3_PAD2	-	TCC7_WO1	TCC0_WO5	-	SDMMC1_SDCD	-	PCC_DEN1	ETH_MDIO	GCLK_IO6	PTC_DRV14	-
38	-	VDDIO	-	-	-	-	-	-	-	-	-	-	-	-	-	-
39	-	VSS	-	-	-	-	-	-	-	-	-	-	-	-	-	-
40	-	PC05	EIC_EXTINT5	-	SERCOM3_PAD3	-	-	TCC1_WO0	-	SDMMC1_SDCK	-	PCC_CLK	ETH_RXD2	GCLK_IO7	PTC_DRV15	-
41	-	PC06	EIC_EXTINT6	AC_AIN2	SERCOM2_PAD0	SERCOM3_PAD1	-	TCC1_WO1	CAN1_TX	-	-	-	ETH_RXD1	-	PTC_DRV16	-
42	-	PC07	EIC_EXTINT7	AC_AIN3	SERCOM2_PAD1	SERCOM3_PAD0	-	TCC1_WO2	CAN1_RX	-	-	-	ETH_RXD0	-	PTC_DRV17	-
43	-	VDDREG	-	-	-	-	-	-	-	-	-	-	-	-	-	-
44	-	PC09	EIC_EXTINT9	-	SERCOM2_PAD2	SERCOM3_PAD2	-	TCC1_WO3	-	-	-	-	ETH_RXER	AC_CMP0	PTC_DRV19	-
45	-	PC10	EIC_EXTINT10	-	SERCOM0_PAD0	SERCOM2_PAD1	-	TCC1_WO5	-	-	-	-	ETH_RXDV	LUT1_IN0	PTC_DRV20	-
46	-	PC11	EIC_EXTINT11	-	SERCOM0_PAD1	SERCOM2_PAD0	-	TCC1_WO6	-	-	-	-	ETH_TXEN	LUT1_IN1	PTC_DRV21	-
47	-	PC12	EIC_EXTINT12	-	SERCOM0_PAD2	SERCOM2_PAD2	-	TCC1_WO7	-	-	-	-	ETH_TXD0	LUT1_IN2	PTC_DRV22	-
48	-	PC13	EIC_EXTINT13	-	SERCOM0_PAD3	SERCOM2_PAD3	-	-	-	-	-	-	ETH_TXD1	LUT1_OUT	PTC_DRV18	-
49	-	VSS	-	-	-	-	-	-	-	-	-	-	-	-	-	-
50	-	VDDIO	-	-	-	-	-	-	-	-	-	-	-	-	-	-
51	-	PD00	EIC_EXTINT0	-	-	-	-	-	-	-	-	-	-	-	-	OSCCTRL_XIN
52	-	PD01	EIC_EXTINT1	-	-	-	-	-	-	-	-	-	-	-	-	OSCCTRL_XOUT
53	-	PD02	EIC_NMI	-	-	-	-	-	USB_SOF_1KHZ	-	-	PCC_DEN2	ETH_TSUCOMP	-	-	-
54	-	-	-	-	-	-	-	-	USB_DM	-	-	-	-	-	-	-
55	-	-	-	-	-	-	-	-	USB_DP	-	-	-	-	-	-	-
56	-	PD05	EIC_EXTINT5	-	SERCOM0_PAD2	-	-	TCC2_WO0	SQI_CS1	SDMMC0_SDWP	-	-	-	GCLK_IO3	-	-
57	-	PD06	EIC_EXTINT6	-	SERCOM0_PAD0	-	TCC2_WO4	TCC2_WO1	SQI_CS0	SDMMC0_SDCMD	SPI_IXS_FSYNC	-	ETH_MDIO	GCLK_IO4	-	-
58	-	PD07	EIC_EXTINT7	-	SERCOM0_PAD1	-	TCC2_WO5	TCC2_WO2	SQI_DATA0	SDMMC0_SDDAT0	SPI_IXS_SDI	-	ETH_MDC	GCLK_IO5	-	-
59	-	VSS	-	-	-	-	-	-	-	-	-	-	-	-	-	-
60	-	VDDIO	-	-	-	-	-	-	-	-	-	-	-	-	-	-
61	-	PD08	EIC_EXTINT8	-	SERCOM1_PAD0	-	-	TCC2_WO3	SQI_DATA2	SDMMC0_SDDAT2	SPI_IXS_SDO	-	-	LUT2_IN0	-	-
62	-	PD09	EIC_EXTINT9	-	SERCOM1_PAD1	-	-	TCC2_WO4	SQI_DATA3	SDMMC0_SDDAT3	-	-	-	LUT2_IN1	-	-

Table 5-6. 64-Pin TQFP/VQFN Package Pinout (PIC32CK1025/2051 Devices) (continued)

Pin	Priority Function	MUXEN=0	MUXEN=1 PMUX Values													External Oscillator I/O
			0x0	0x1	0x2	0x3	0x5	0x6	0x7	0x8	0x9	0xa	0xb	0xc	0xd	
	RTC Security Tamper Input Loop(s)	Port	EIC	ADCN/ADC/AC	SERCOM	SERCOM Alt	TCC	TCC Alt/PDEC	SQI/CAN/USB/CM33 Trace	SDMMC	SPI_IXS	PCC	ETH	MISC	PTC/Debug	
63	-	PD10	EIC_EXTINT10	-	SERCOM1_PAD2	-	-	TCC2_WO5	SQI_CLK	SDMMC0_SDCK	SPI_IXS_SCK	-	-	LUT2_IN2	-	-
64	-	PD11	EIC_EXTINT11	-	SERCOM1_PAD3	-	TCC2_WO2	TCC2_WO6	SQI_DATA1	SDMMC0_SDDAT1	-	-	-	LUT2_OUT	-	-

Notes:
1. Pin column items displayed in **BOLD** are 5.5V tolerant pins.
2. PA02 and PA03 are dedicated SWDIO and SWCLK pins. They are not recommended for GPIO use.
3. VDDA and AVDD are used interchangeably.

5.3. 100-Pin Packages

5.3.1. 100-Pin TQFP Package

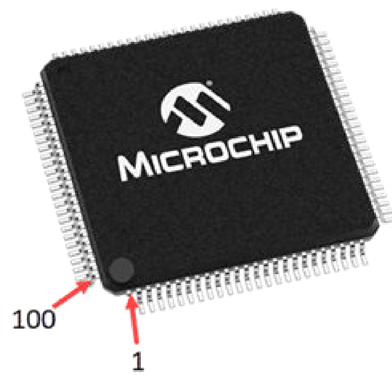
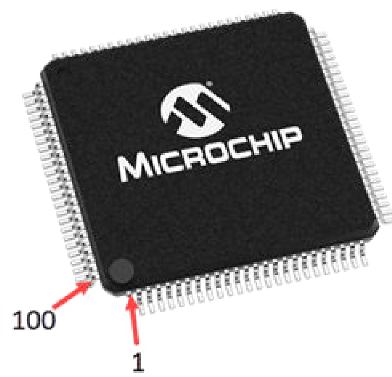
PIC32CK1012GC01100 PIC32CK1012SG01100	**Top View** 
PIC32CK(1025/2051)GC01100 PIC32CK(1025/2051)SG01100	**Top View** 

Table 5-7. 100-Pin TQFP Package Pinout (PIC32CK1012 Devices)

Pin	SUPC	Priority Function	MUXEN = 0	PMUX Values MUXEN = 1													XOSC I/O
				0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7	0x8	0x9	0xA	0xB	0xC	
		RTC Security Tamper	PORT	EIC	ADCN/ADC/AC	SERCOM	SERCOM Alt	TCC	TCC Alt/PDEC	SQI/CAN/USB/CM33 Trace	SDMMC	SPI_IXS	PCC	ETH	MISC	PTC/DEBUG	
1	-	-	**PA00**	EIC_EXTINT14	-	SERCOM0_PAD2	SERCOM1_PAD0	TCC2_WO0	-	-	-	-	-	-	CCL_OUT1	BSDAP_TDI	-
2	-	-	**PA01**	EIC_EXTINT1	-	SERCOM0_PAD3	SERCOM1_PAD1	TCC2_WO3	-	CPU0_SWO	-	-	-	-	GCLK_IO1	BSDAP_TDO	-
3	-	-	PA02[2]	EIC_EXTINT2	-	-	-	-	-	-	-	-	-	-	-	BSDAP_TMS_SWDIO	-
4	-	-	PA03[2]	EIC_EXTINT3	-	-	-	-	-	-	-	-	-	-	GCLK_IO0	BSDAP_TCK_SWCLK	-
5	-	-	RESET_N	-	-	-	-	-	-	-	-	-	-	-	-	-	-
6	-	-	VSS	-	-	-	-	-	-	-	-	-	-	-	-	-	-
7	-	-	VDDIO	-	-	-	-	-	-	-	-	-	-	-	-	-	-
8	-	-	PA13	EIC_EXTINT13	-	SERCOM5_PAD0	-	TCC4_WO0	-	CPU0_TRACEDATA3	-	-	-	-	CCL_IN9	-	-
9	-	-	PA14	EIC_EXTINT14	-	SERCOM5_PAD1	-	TCC4_WO1	-	CPU0_TRACEDATA2	-	-	-	-	CCL_IN10	-	-
10	-	-	PA15	EIC_EXTINT15	-	SERCOM5_PAD2	-	TCC5_WO0	-	CPU0_TRACEDATA1	-	-	-	-	CCL_IN11	-	-
11	-	-	PA16	EIC_EXTINT0	-	SERCOM5_PAD3	-	TCC5_WO1	-	CPU0_TRACECLK	-	-	-	-	CCL_OUT3	-	-
12	-	-	PA17	EIC_EXTINT1	-	-	-	-	-	CPU0_TRACEDATA0	-	-	-	-	GCLK_IO6	-	-
13	-	-	PA04	EIC_EXTINT4	ADC_VINP0	-	-	TCC3_WO0	-	-	-	-	-	-	CCL_IN9	-	-
14	-	-	PA05	EIC_EXTINT5	ADC_VINM0/ ADC_VINP1	-	SERCOM5_PAD0	TCC3_WO1	-	-	-	-	-	-	CCL_IN10	-	-
15	-	-	PA06	EIC_EXTINT6	ADC_VINP2/ AC_AIN0	-	SERCOM5_PAD1	TCC3_WO2	TCC3_WO6	-	-	-	-	-	CCL_IN11	-	-
16	-	-	PA07	EIC_EXTINT7	ADC_VINM2/ ADC_VINP3/ AC_AIN1	-	SERCOM5_PAD2	TCC3_WO3	TCC3_WO7	-	-	-	-	-	CCL_OUT3	-	-
17	-	-	PA08	EIC_EXTINT8	ADC_VINP4	-	SERCOM5_PAD3	TCC3_WO4	-	-	-	-	-	-	-	-	-
18	-	-	PA18	EIC_EXTINT2	ADC_VINM4	-	-	TCC3_WO5	-	-	-	-	-	-	-	PTC_ECI0	-
19	-	-	VSS	-	-	-	-	-	-	-	-	-	-	-	-	-	-
20	-	-	VDDIO	-	-	-	-	-	-	-	-	-	-	-	-	-	-
21	-	-	PA19	EIC_EXTINT3	-	-	-	TCC3_WO7	-	-	-	-	-	-	-	PTC_ECI1	-
22	-	-	PA09	EIC_EXTINT9	ADC_VINP5/ AC_AIN0	-	-	-	TCC3_WO0	-	-	SPI_IXS_FSYNC	-	-	GCLK_IO0	PTC_DRV0	-
23	-	-	PA10	EIC_EXTINT10	ADC_VINP6/ AC_AIN1	-	SERCOM4_PAD2	-	TCC3_WO1	SQI_CS2	SDMMC0_SDCD	-	PCC_DEN1	-	GCLK_IO1	PTC_DRV1	-
24	-	-	PA11	EIC_EXTINT11	ADC_VINP7	SERCOM5_PAD0	-	TCC4_WO0	TCC3_WO2	-	-	SPI_IXS_SCK	-	ETH_MDC	AC_CMP0	PTC_DRV2	-
25	-	-	PA12	EIC_EXTINT12	ADC_VINP8	SERCOM5_PAD1	-	TCC4_WO1	TCC3_WO3	-	SDMMC0_SDCD	SPI_IXS_SDI	-	ETH_MDIO	AC_CMP1	PTC_DRV3	-
26	-	RTC_OUT1	PB00	EIC_EXTINT0	ADC_VINP9	SERCOM5_PAD2	-	TCC5_WO0	TCC3_WO4	CAN0_TX	SDMMC0_SDWP	SPI_IXS_SDO	-	-	CCL_IN6	PTC_DRV4	-
27	-	RTC_IN1	PB01	EIC_EXTINT1	ADC_VREFPH	SERCOM5_PAD3	-	TCC5_WO1	TCC3_WO5	CAN0_RX	-	-	-	-	CCL_IN7	PTC_DRV5	-
28	-	RTC_OUT2	PB11	EIC_EXTINT11	ADC_VINP10	-	-	-	TCC3_WO6	-	-	-	-	-	-	-	-
29	-	RTC_OUT3	PB12	EIC_EXTINT12	ADC_VINP11	-	-	TCC4_WO0	TCC3_WO7	-	-	-	-	-	-	-	-
30	-	-	AVSS	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Table 5-7. 100-Pin TQFP Package Pinout (PIC32CK1012 Devices) (continued)

Pin	SUPC	Priority Function RTC Security Tamper	MUXEN = 0 PORT	PMUX Values MUXEN = 1 0x0 EIC	0x1 ADCN/ADC/AC	0x2 SERCOM	0x3 SERCOM Alt	0x4 TCC	0x5 TCC Alt/PDEC	0x6 SQI/CAN/USB/CM33 Trace	0x7 SDMMC	0x8 SPI_IXS	0x9 PCC	0xA ETH	0xB MISC	0xC PTC/DEBUG	XOSC I/O
31	-	-	PB02	EIC_EXTINT2	-	-	-	-	-	-	-	-	-	-	CCL_IN0	-	OSC32KCTRL_XIN32
32	-	-	PB03	EIC_EXTINT3	-	-	-	-	-	-	-	-	-	-	CCL_IN1	-	OSC32KCTRL_XOUT32
33	-	-	AVDD	-	-	-	-	-	-	-	-	-	-	-	-	-	-
34	-	-	VDDIO	-	-	-	-	-	-	-	-	-	-	-	-	-	-
35	-	-	VSS	-	-	-	-	-	-	-	-	-	-	-	-	-	-
36	-	-	PB04	EIC_EXTINT4	-	SERCOM4_PAD0	SERCOM4_PAD0	TCC0_WO0	PDEC_QDI0	-	-	-	-	-	CCL_IN8	-	-
37	SUPC_OUT0	RTC_OUT0	PB05	EIC_EXTINT5	-	SERCOM4_PAD1	SERCOM4_PAD1	TCC0_WO1	PDEC_QDI1	-	-	-	-	-	CCL_OUT3	-	-
38	SUPC_OUT1	RTC_IN4	PB13	EIC_EXTINT13	-	SERCOM4_PAD2	-	TCC0_WO2	PDEC_QDI2	-	-	-	-	-	-	-	-
39	-	RTC_IN0	PB06	EIC_EXTINT6	-	SERCOM4_PAD3	SERCOM4_PAD3	TCC0_WO3	PDEC_QDI2	-	-	-	-	-	CCL_OUT0	-	-
40	-	RTC_IN2	PB14	EIC_EXTINT14	-	-	-	TCC0_WO4	-	-	-	-	-	-	-	-	-
41	-	RTC_IN3	PB15	EIC_EXTINT15	-	-	-	TCC0_WO5	-	-	-	-	-	-	-	-	-
42	-	-	VDDIO	-	-	-	-	-	-	-	-	-	-	-	-	-	-
43	-	-	VSS	-	-	-	-	-	-	-	-	-	-	-	-	-	-
44	-	-	VDDREG	-	-	-	-	-	-	-	-	-	-	-	-	-	-
45	-	RTC_OUT4	PB16	EIC_EXTINT0	-	SERCOM3_PAD0	-	-	-	CAN0_TX	-	-	-	-	-	PTC_DRV24	-
46	-	-	PB17	EIC_EXTINT1	-	SERCOM3_PAD1	-	-	-	CAN0_RX	-	-	-	-	-	PTC_DRV25	-
47	-	-	PB07	EIC_EXTINT7	-	SERCOM3_PAD2	SERCOM4_PAD2	-	PDEC_QDI0	-	SDMMC0_SDCMD	-	PCC_DATA3	-	CCL_IN0	PTC_DRV6	-
48	-	-	PB08	EIC_EXTINT8	-	SERCOM3_PAD3	SERCOM4_PAD3	-	PDEC_QDI1	CAN0_TX	-	-	PCC_DATA4	-	CCL_IN1	PTC_DRV7	-
49	-	-	PB09	EIC_EXTINT9	-	SERCOM4_PAD0	SERCOM4_PAD1	-	PDEC_QDI2	CAN0_RX	-	-	PCC_DATA5	-	CCL_IN2	PTC_DRV8	-
50	-	-	PB10	EIC_EXTINT10	-	SERCOM4_PAD1	SERCOM4_PAD0	-	TCC0_WO0	-	SDMMC0_SDDAT0	-	PCC_DATA6	ETH_MDC	CCL_OUT0	PTC_DRV9	-
51	-	-	PC00	EIC_EXTINT0	-	SERCOM4_PAD2	SERCOM3_PAD3	-	TCC0_WO1	-	SDMMC0_SDDAT1	-	PCC_DATA7	ETH_TX_CLK/ ETH_REF_CLK	GCLK_IO2	PTC_DRV10	-
52	-	-	PC01	EIC_EXTINT1	-	SERCOM4_PAD3	SERCOM3_PAD2	-	TCC0_WO2	USB_SOF_1KHZ	SDMMC0_SDDAT2	-	PCC_DATA0	-	GCLK_IO3	PTC_DRV11	-
53	-	-	PC02	EIC_EXTINT2	-	SERCOM3_PAD0	SERCOM2_PAD3	-	TCC0_WO3	CAN0_TX	SDMMC0_SDDAT3	-	PCC_DATA1	ETH_MDIO	GCLK_IO4	PTC_DRV12	-
54	-	-	PC03	EIC_EXTINT3	-	SERCOM3_PAD1	-	-	TCC0_WO4	CAN0_RX	SDMMC0_SDWP	-	PCC_DATA2	ETH_MDC	GCLK_IO5	PTC_DRV13	-
55	-	-	PC04	EIC_EXTINT4	-	SERCOM3_PAD2	-	-	TCC0_WO5	-	SDMMC0_SDCD	-	PCC_DEN1	ETH_MDIO	GCLK_IO6	PTC_DRV14	-
56	-	-	VDDIO	-	-	-	-	-	-	-	-	-	-	-	-	-	-
57	-	-	VSS	-	-	-	-	-	-	-	-	-	-	-	-	-	-
58	-	-	PC14	EIC_EXTINT14	-	SERCOM2_PAD3	-	-	-	-	-	-	PCC_DEN2	ETH_RXD3	-	PTC_DRV26	-
59	-	-	PC05	EIC_EXTINT5	-	SERCOM3_PAD3	SERCOM2_PAD2	-	TCC1_WO0	-	SDMMC0_SDCK	-	PCC_CLK	ETH_RXD2	GCLK_IO7	PTC_DRV15	-
60	-	-	PC06	EIC_EXTINT6	AC_AIN2	SERCOM2_PAD0	SERCOM3_PAD1	-	TCC1_WO1	CAN0_TX	SDMMC0_SDCMD	-	PCC_DATA3	ETH_RXD1	CCL_IN5	PTC_DRV16	-
61	-	-	PC07	EIC_EXTINT7	AC_AIN3	SERCOM2_PAD1	SERCOM3_PAD0	-	TCC1_WO2	CAN0_RX	SDMMC0_SDDAT0	-	PCC_DATA4	ETH_RXD0	CCL_OUT2	PTC_DRV17	-
62	-	-	VDDREG	-	-	-	-	-	-	-	-	-	-	-	-	-	-
63	-	-	PC09	EIC_EXTINT9	-	SERCOM2_PAD2	SERCOM3_PAD2	-	TCC1_WO3	-	-	-	-	ETH_RXER	AC_CMP0	PTC_DRV19	-

Table 5-7. 100-Pin TQFP Package Pinout (PIC32CK1012 Devices) (continued)

Pin	SUPC	Priority Function / RTC Security Tamper	MUXEN = 0 / PORT	PMUX Values, MUXEN = 1: 0x0 EIC	0x1 ADCN/ADC/AC	0x2 SERCOM	0x3 SERCOM Alt	0x4 TCC	0x5 TCC Alt/PDEC	0x6 SQI/CAN/USB/CM33 Trace	0x7 SDMMC	0x8 SPI_IXS	0x9 PCC	0xA ETH	0xB MISC	0xC PTC/DEBUG	XOSC I/O
64	-	-	PC15	EIC_EXTINT15	-	SERCOM2_PAD0	SERCOM3_PAD3	TCC1_WO0	TCC1_WO4	-	-	-	PCC_DATA8	ETH_RX_CLK	AC_CMP1	PTC_DRV27	-
65	-	-	PC16	EIC_EXTINT0	-	SERCOM2_PAD1	-	TCC1_WO1	-	-	-	-	PCC_DATA9	ETH_TXER	-	PTC_DRV28	-
66	-	-	PC17	EIC_EXTINT1	-	SERCOM2_PAD2	-	TCC1_WO2	-	-	-	-	PCC_DATA10	ETH_CRS	-	PTC_DRV29	-
67	-	-	PC18	EIC_EXTINT2	-	SERCOM2_PAD3	-	TCC1_WO3	-	-	-	-	PCC_DATA11	ETH_COL	-	PTC_DRV30	-
68	-	-	PC10	EIC_EXTINT10	-	SERCOM0_PAD0	SERCOM2_PAD1	TCC2_WO2	TCC1_WO5	-	SDMMC0_SDCD	-	PCC_DATA5	ETH_RXDV	CCL_IN3	PTC_DRV20	-
69	-	-	VDDIO	-	-	-	-	-	-	-	-	-	-	-	-	-	-
70	-	-	VSS	-	-	-	-	-	-	-	-	-	-	-	-	-	-
71	-	-	PC11	EIC_EXTINT11	-	SERCOM0_PAD1	SERCOM2_PAD0	TCC2_WO1	TCC1_WO6	-	-	-	PCC_DATA6	ETH_TXEN	CCL_IN4	PTC_DRV21	-
72	-	-	PC12	EIC_EXTINT12	-	SERCOM0_PAD2	SERCOM2_PAD2	-	TCC1_WO7	-	-	-	-	ETH_TXD0	CCL_IN5	PTC_DRV22	-
73	-	-	PC13	EIC_EXTINT13	-	SERCOM0_PAD3	SERCOM2_PAD3	-	-	-	-	-	-	ETH_TXD1	CCL_OUT1	PTC_DRV18	-
74	-	-	PC19	EIC_EXTINT3	-	-	-	TCC1_WO4	-	-	-	-	PCC_DATA12	ETH_TXD2	-	PTC_DRV31	-
75	-	-	PC20	EIC_EXTINT4	-	-	-	TCC1_WO5	-	-	-	-	PCC_DATA13	ETH_TXD3	-	PTC_DRV23	-
76	-	-	NC	-	-	-	-	-	-	-	-	-	-	-	-	-	-
77	-	-	NC	-	-	-	-	-	-	-	-	-	-	-	-	-	-
78	-	-	NC	-	-	-	-	-	-	-	-	-	-	-	-	-	-
79	-	-	VSS	-	-	-	-	-	-	-	-	-	-	-	-	-	-
80	-	-	VSS	-	-	-	-	-	-	-	-	-	-	-	-	-	-
81	-	-	NC	-	-	-	-	-	-	-	-	-	-	-	-	-	-
82	-	-	NC	-	-	-	-	-	-	-	-	-	-	-	-	-	-
83	-	-	VDDREG_USB	-	-	-	-	-	-	-	-	-	-	-	-	-	-
84	-	-	VSS	-	-	-	-	-	-	-	-	-	-	-	-	-	-
85	-	-	VDDIO	-	-	-	-	-	-	-	-	-	-	-	-	-	-
86	-	-	PD00	EIC_EXTINT0	-	-	-	-	-	-	-	-	-	-	-	-	OSCCTRL_XIN
87	-	-	PD01	EIC_EXTINT1	-	-	-	-	-	-	-	-	-	-	-	-	OSCCTRL_XOUT
88	-	-	PD02	EIC_NMI	-	-	-	-	-	USB_SOF_1KHZ	-	-	PCC_DEN2	ETH_TSUCOMP	CCL_IN2	-	-
89	-	-	-	-	-	-	-	-	-	USB_DM	-	-	-	-	-	-	-
90	-	-	-	-	-	-	-	-	-	USB_DP	-	-	-	-	-	-	-
91	-	-	PD12	EIC_EXTINT12	-	SERCOM0_PAD3	-	TCC2_WO1	TCC2_WO7	SQI_CS3	SDMMC0_SDCD	-	-	-	GCLK_IO2	-	-
92	-	-	PD05	EIC_EXTINT5	-	SERCOM0_PAD2	-	-	TCC2_WO0	SQI_CS1	SDMMC0_SDWP	-	-	-	GCLK_IO3	-	-
93	-	-	PD06	EIC_EXTINT6	-	SERCOM0_PAD0	SERCOM1_PAD2	TCC2_WO4	TCC2_WO1	SQI_CS0	SDMMC0_SDCMD	SPI_IXS_FSYNC	-	ETH_MDIO	GCLK_IO4	-	-
94	-	-	PD07	EIC_EXTINT7	-	SERCOM0_PAD1	SERCOM1_PAD3	TCC2_WO5	TCC2_WO2	SQI_DATA0	SDMMC0_SDDAT0	SPI_IXS_SDI	-	ETH_MDC	GCLK_IO5	-	-
95	-	-	VSS	-	-	-	-	-	-	-	-	-	-	-	-	-	-
96	-	-	VDDIO	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Table 5-7. 100-Pin TQFP Package Pinout (PIC32CK1012 Devices) (continued)

Pin	SUPC	Priority Function	MUXEN = 0	PMUX Values MUXEN = 1													XOSC I/O
				0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7	0x8	0x9	0xA	0xB	0xC	
		RTC Security Tamper	PORT	EIC	ADCN/ADC/AC	SERCOM	SERCOM Alt	TCC	TCC Alt/PDEC	SQI/CAN/USB/CM33 Trace	SDMMC	SPI_IXS	PCC	ETH	MISC	PTC/DEBUG	
97	-	-	PD08	EIC_EXTINT8	-	SERCOM1_PAD0	-	-	TCC2_WO3	SQI_DATA2	SDMMC0_SDDAT2	SPI_IXS_SDO	-	-	CCL_IN6	-	-
98	-	-	PD09	EIC_EXTINT9	-	SERCOM1_PAD1	-	-	TCC2_WO4	SQI_DATA3	SDMMC0_SDDAT3	-	-	-	CCL_IN7	-	-
99	-	-	PD10	EIC_EXTINT10	-	SERCOM1_PAD2	-	-	TCC2_WO5	SQI_CLK	SDMMC0_SDCK	SPI_IXS_SCK	-	-	CCL_IN8	-	-
100	-	-	PD11	EIC_EXTINT11	-	SERCOM1_PAD3	-	TCC2_WO2	TCC2_WO6	SQI_DATA1	SDMMC0_SDDAT1	-	-	-	CCL_OUT2	-	-

Table 5-8. 100-Pin TQFP Package Pinout (PIC32CK1025/2051 Devices)

Pin	Priority Function	MUXEN=0	MUXEN=1														External Oscillator I/O
			PMUX Values														
			0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7	0x8	0x9	0xa	0xb	0xc	0xd	
	RTC Security Tamper Input Loop(s)	Port	EIC	ADCN/ADC/AC	SERCOM	SERCOM Alt	EBI	TCC	TCC Alt/PDEC	SQI/CAN/USB/CM33 Trace	SDMMC	SPI_IXS	PCC	ETH	MISC	PTC/DEBUG	
1	-	**PA00**	EIC_EXTINT14	-	-	SERCOM6_PAD0	-	TCC2_WO0	-	-	-	-	-	-	-	BSDAP_TDI	-
2	-	**PA01**	EIC_EXTINT1	-	SERCOM7_PAD0	SERCOM6_PAD1	-	TCC2_WO3	-	CPU0_SWO	-	-	-	-	GCLK_IO1	BSDAP_TDO	-
3	-	PA02[2]	-	-	-	-	-	-	-	-	-	-	-	-	-	BSDAP_TMS_SWDIO	-
4	-	PA03[2]	-	-	-	-	-	-	-	-	-	-	-	-	-	BSDAP_TCK_SWCLK	-
5	-	RESET_N	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
6	-	VSS	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
7	-	VDDIO	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
8	-	PA13	EIC_EXTINT13	-	SERCOM5_PAD0	-	EBI_D0	TCC4_WO0	-	CPU0_TRACEDATA3	-	-	-	-	LUT3_IN0	-	-
9	-	PA14	EIC_EXTINT14	-	SERCOM5_PAD1	-	EBI_D1	TCC4_WO1	-	CPU0_TRACEDATA2	-	-	-	-	LUT3_IN1	-	-
10	-	PA15	EIC_EXTINT15	-	SERCOM5_PAD2	-	EBI_D2	TCC5_WO0	-	CPU0_TRACEDATA1	-	-	-	-	LUT3_IN2	-	-
11	-	PA16	EIC_EXTINT0	-	SERCOM5_PAD3	-	EBI_D3	TCC5_WO1	-	CPU0_TRACECLK	-	-	-	-	LUT3_OUT	-	-
12	-	PA17	EIC_EXTINT1	-	-	-	EBI_D4	-	-	CPU0_TRACEDATA0	-	-	-	-	GCLK_IO6	-	-
13	-	PA04	EIC_EXTINT4	ADC_VINP0	SERCOM7_PAD1	-	EBI_D5	TCC3_WO0	-	-	-	-	-	-	LUT3_IN0	-	-
14	-	PA05	EIC_EXTINT5	ADC_VINM0/ ADC_VINP1	SERCOM7_PAD2	SERCOM5_PAD0	EBI_D6	TCC3_WO1	-	-	-	-	-	-	LUT3_IN1	-	-
15	-	PA06	EIC_EXTINT6	AC_AIN0/ ADC_VINP2	SERCOM7_PAD3	SERCOM5_PAD1	EBI_D7	TCC3_WO2	-	-	-	-	-	-	LUT3_IN2	-	-
16	-	PA07	EIC_EXTINT7	AC_AIN1/ ADC_VINM2/ ADC_VINP3	SERCOM6_PAD0	SERCOM5_PAD2	EBI_D8	TCC3_WO3	-	-	-	-	-	-	LUT3_OUT	-	-
17	-	PA08	EIC_EXTINT8	ADC_VINP4	SERCOM6_PAD1	SERCOM5_PAD3	EBI_D9	TCC3_WO4	-	-	-	-	-	-	-	-	-
18	-	PA18	EIC_EXTINT2	ADC_VINM4	-	SERCOM6_PAD0	-	TCC3_WO5	-	-	-	-	-	-	-	PTC_ECI0	-
19	-	VSS	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
20	-	VDDIO	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
21	-	PA19	EIC_EXTINT3	-	-	SERCOM6_PAD2	-	TCC3_WO7	-	-	-	-	-	-	-	PTC_ECI1	-

Table 5-8. 100-Pin TQFP Package Pinout (PIC32CK1025/2051 Devices) (continued)

Pin	Priority Function	MUXEN=0	MUXEN=1														External Oscillator I/O
			PMUX Values														
			0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7	0x8	0x9	0xa	0xb	0xc	0xd	
	RTC Security Tamper Input Loop(s)	Port	EIC	ADCN/ADC/AC	SERCOM	SERCOM Alt	EBI	TCC	TCC Alt/PDEC	SQI/CAN/USB/CM33 Trace	SDMMC	SPI_IXS	PCC	ETH	MISC	PTC/DEBUG	
22	-	PA09	EIC_EXTINT9	ADC_VINP5	SERCOM6_PAD2	SERCOM6_PAD3	-	-	TCC3_WO0	-	-	SPI_IXS_FSYNC	-	-	GCLK_IO0	PTC_DRV0	-
23	-	PA10	EIC_EXTINT10	ADC_VINP6	SERCOM6_PAD3	-	EBI_NWAIT	-	TCC3_WO1	SQI_CS2	SDMMC0_SDCD	-	PCC_DEN1	-	GCLK_IO1	PTC_DRV1	-
24	-	PA11	EIC_EXTINT11	ADC_VINP7	SERCOM5_PAD0	SERCOM7_PAD1	-	TCC4_WO0	TCC3_WO2	-	-	SPI_IXS_SCK	-	ETH_MDC	AC_CMP0	PTC_DRV2	-
25	-	PA12	EIC_EXTINT12	ADC_VINP8	SERCOM5_PAD1	SERCOM7_PAD0	-	TCC4_WO1	TCC3_WO3	-	SDMMC0_SDCD	SPI_IXS_SDI	-	ETH_MDIO	AC_CMP1	PTC_DRV3	-
26	RTC_OUT1	PB00	EIC_EXTINT0	ADC_VINP9	SERCOM5_PAD2	SERCOM7_PAD2	-	TCC5_WO0	TCC3_WO4	CAN1_TX	SDMMC0_SDWP	SPI_IXS_SDO	-	-	-	PTC_DRV4	-
27	RTC_IN1	PB01	EIC_EXTINT1	ADC_VREFH	SERCOM5_PAD3	SERCOM7_PAD3	-	TCC5_WO1	TCC3_WO5	CAN1_RX	-	-	-	-	-	PTC_DRV5	-
28	RTC_OUT2	PB11	EIC_EXTINT11	ADC_VINP10	-	-	EBI_D10	-	TCC3_WO6	-	-	-	-	-	-	-	-
29	RTC_OUT3	PB12	EIC_EXTINT12	ADC_VINP11	SERCOM4_PAD0	SERCOM7_PAD1	EBI_D11	TCC4_WO0	TCC3_WO7	-	-	-	-	-	-	-	-
30	-	AVSS	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
31	-	PB02	EIC_EXTINT2	-	-	-	-	-	-	-	-	-	-	-	-	-	OSC32KCTRL_XIN32
32	-	PB03	EIC_EXTINT3	-	-	-	-	-	-	-	-	-	-	-	-	-	OSC32KCTRL_XOUT32
33	-	AVDD	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
34	-	VDDIO	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
35	-	VSS	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
36	-	PB04	EIC_EXTINT4	-	SERCOM4_PAD0	-	-	TCC0_WO0	PDEC_QDI0	-	-	-	-	-	-	-	-
37	RTC_OUT0	PB05	EIC_EXTINT5	-	SERCOM4_PAD1	-	EBI_D12	TCC0_WO1	PDEC_QDI1	-	-	-	-	-	-	-	-
38	RTC_IN4	PB13	EIC_EXTINT13	-	SERCOM4_PAD2	-	EBI_D13	TCC0_WO2	PDEC_QDI2	-	-	-	-	-	-	-	-
39	RTC_IN0	PB06	EIC_EXTINT6	-	SERCOM4_PAD3	-	EBI_D14	TCC0_WO3	-	-	-	-	-	-	-	-	-
40	RTC_IN2	PB14	EIC_EXTINT14	-	-	-	EBI_D15	TCC0_WO4	-	-	-	-	-	-	-	-	-
41	RTC_IN3	PB15	EIC_EXTINT15	-	-	-	EBI_NRD	TCC0_WO5	-	-	-	-	-	-	-	-	-
42	-	VDDIO	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
43	-	VSS	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
44	-	VDDREG	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Table 5-8. 100-Pin TQFP Package Pinout (PIC32CK1025/2051 Devices) (continued)

Pin	Priority Function: RTC Security Tamper Input Loop(s)	MUXEN=0: Port	MUXEN=1 PMUX Values 0x0: EIC	0x1: ADCN/ADC/AC	0x2: SERCOM	0x3: SERCOM Alt	0x4: EBI	0x5: TCC	0x6: TCC Alt/PDEC	0x7: SQI/CAN/USB/CM33 Trace	0x8: SDMMC	0x9: SPI_IXS	0xa: PCC	0xb: ETH	0xc: MISC	0xd: PTC/DEBUG	External Oscillator I/O
45	RTC_OUT4	PB16	EIC_EXTINT0	-	SERCOM3_PAD0	-	-	TCC6_WO0	-	CAN0_TX	-	-	-	-	-	PTC_DRV24	-
46	-	PB17	EIC_EXTINT1	-	SERCOM3_PAD1	-	-	TCC6_WO1	-	CAN0_RX	-	-	-	-	-	PTC_DRV25	-
47	-	PB07	EIC_EXTINT7	-	SERCOM3_PAD2	SERCOM4_PAD2	EBI_NBS1_NWR1	TCC7_WO0	PDEC_QDI0	-	SDMMC1_SDCMD	-	PCC_DATA3	-	LUT0_IN0	PTC_DRV6	-
48	-	PB08	EIC_EXTINT8	-	SERCOM3_PAD3	SERCOM4_PAD3	EBI_NBS0_A0	TCC7_WO1	PDEC_QDI1	CAN0_TX	-	-	PCC_DATA4	-	LUT0_IN1	PTC_DRV7	-
49	-	PB09	EIC_EXTINT9	-	SERCOM4_PAD0	SERCOM4_PAD1	EBI_A15	-	PDEC_QDI2	CAN0_RX	-	-	PCC_DATA5	-	LUT0_IN2	PTC_DRV8	-
50	-	PB10	EIC_EXTINT10	-	SERCOM4_PAD1	SERCOM4_PAD0	EBI_A14	-	TCC0_WO0	-	SDMMC1_SDDAT0	-	PCC_DATA6	ETH_MDC	LUT0_OUT	PTC_DRV9	-
51	-	PC00	EIC_EXTINT0	-	SERCOM4_PAD2	-	EBI_A13	-	TCC0_WO1	-	SDMMC1_SDDAT1	-	PCC_DATA7	ETH_REF_CLK/ ETH_TX_CLK	GCLK_IO2	PTC_DRV10	-
52	-	PC01	EIC_EXTINT1	-	SERCOM4_PAD3	-	EBI_A12	TCC6_WO0	TCC0_WO2	USB_SOF_1KHZ	SDMMC1_SDDAT2	-	PCC_DATA0	-	GCLK_IO3	PTC_DRV11	-
53	-	PC02	EIC_EXTINT2	-	SERCOM3_PAD0	-	EBI_A11	TCC6_WO1	TCC0_WO3	CAN0_TX	SDMMC1_SDDAT3	-	PCC_DATA1	ETH_MDIO	GCLK_IO4	PTC_DRV12	-
54	-	PC03	EIC_EXTINT3	-	SERCOM3_PAD1	-	EBI_A10	TCC7_WO0	TCC0_WO4	CAN0_RX	SDMMC1_SDWP	-	PCC_DATA2	ETH_MDC	GCLK_IO5	PTC_DRV13	-
55	-	PC04	EIC_EXTINT4	-	SERCOM3_PAD2	-	EBI_A9	TCC7_WO1	TCC0_WO5	-	SDMMC1_SDCD	-	PCC_DEN1	ETH_MDIO	GCLK_IO6	PTC_DRV14	-
56	-	VDDIO	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
57	-	VSS	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
58	-	PC14	EIC_EXTINT14	-	SERCOM2_PAD2	-	EBI_A8	-	-	-	-	-	PCC_DEN2	ETH_RXD3	-	PTC_DRV26	-
59	-	PC05	EIC_EXTINT5	-	SERCOM3_PAD3	-	EBI_A7	-	TCC1_WO0	-	SDMMC1_SDCK	-	PCC_CLK	ETH_RXD2	GCLK_IO7	PTC_DRV15	-
60	-	PC06	EIC_EXTINT6	AC_AIN2	SERCOM2_PAD0	SERCOM3_PAD1	-	-	TCC1_WO1	CAN1_TX	-	-	-	ETH_RXD1	-	PTC_DRV16	-
61	-	PC07	EIC_EXTINT7	AC_AIN3	SERCOM2_PAD1	SERCOM3_PAD0	-	-	TCC1_WO2	CAN1_RX	-	-	-	ETH_RXD0	-	PTC_DRV17	-
62	-	VDDREG	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
63	-	PC09	EIC_EXTINT9	-	SERCOM2_PAD2	SERCOM3_PAD2	-	-	TCC1_WO3	-	-	-	-	ETH_RXER	AC_CMP0	PTC_DRV19	-
64	-	PC15	EIC_EXTINT15	-	SERCOM2_PAD0	SERCOM3_PAD3	EBI_A6	TCC1_WO0	TCC1_WO4	-	-	-	PCC_DATA8	ETH_RX_CLK	AC_CMP1	PTC_DRV27	-
65	-	PC16	EIC_EXTINT0	-	SERCOM2_PAD1	-	EBI_A5	TCC1_WO1	-	-	-	-	PCC_DATA9	ETH_TXER	-	PTC_DRV28	-

Table 5-8. 100-Pin TQFP Package Pinout (PIC32CK1025/2051 Devices) (continued)

Pin	Priority Function: RTC Security Tamper Input Loop(s)	MUXEN=0: Port	MUXEN=1 PMUX Values 0x0: EIC	0x1: ADCN/ADC/AC	0x2: SERCOM	0x3: SERCOM Alt	0x4: EBI	0x5: TCC	0x6: TCC Alt/PDEC	0x7: SQI/CAN/USB/CM33 Trace	0x8: SDMMC	0x9: SPI_IXS	0xa: PCC	0xb: ETH	0xc: MISC	0xd: PTC/DEBUG	External Oscillator I/O
66	-	PC17	EIC_EXTINT1	-	SERCOM2_PAD2	-	EBI_A4	TCC1_WO2	-	-	-	-	PCC_DATA10	ETH_CRS	-	PTC_DRV29	-
67	-	PC18	EIC_EXTINT2	-	SERCOM2_PAD3	-	EBI_A3	TCC1_WO3	-	-	-	-	PCC_DATA11	ETH_COL	-	PTC_DRV30	-
68	-	PC10	EIC_EXTINT10	-	SERCOM0_PAD0	SERCOM2_PAD1	-	-	TCC1_WO5	-	-	-	-	ETH_RXDV	LUT1_IN0	PTC_DRV20	-
69	-	VDDIO	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
70	-	VSS	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
71	-	PC11	EIC_EXTINT11	-	SERCOM0_PAD1	SERCOM2_PAD0	-	-	TCC1_WO6	-	-	-	-	ETH_TXEN	LUT1_IN1	PTC_DRV21	-
72	-	PC12	EIC_EXTINT12	-	SERCOM0_PAD2	SERCOM2_PAD2	-	-	TCC1_WO7	-	-	-	-	ETH_TXD0	LUT1_IN2	PTC_DRV22	-
73	-	PC13	EIC_EXTINT13	-	SERCOM0_PAD3	SERCOM2_PAD3	-	-	-	-	-	-	-	ETH_TXD1	LUT1_OUT	PTC_DRV18	-
74	-	PC19	EIC_EXTINT3	-	-	-	EBI_A2	TCC1_WO4	-	-	-	-	PCC_DATA12	ETH_TXD2	-	PTC_DRV31	-
75	-	PC20	EIC_EXTINT4	-	-	-	EBI_NBS2_NWR2_A1	TCC1_WO5	-	-	-	-	PCC_DATA13	ETH_TXD3	-	PTC_DRV23	-
76	-	USB0PN	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
77	-	USB0PP	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
78	-	USB0ID	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
79	-	VUSB3V3	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
80	-	USBVSSA	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
81	-	VBUS	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
82	-	USRBIAS	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
83	-	VDDREG_USB	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
84	-	VSS	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
85	-	VDDIO	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
86	-	PD00	EIC_EXTINT0	-	-	-	-	-	-	-	-	-	-	-	-	-	OSCCTRL_XIN
87	-	PD01	EIC_EXTINT1	-	-	-	-	-	-	-	-	-	-	-	-	-	OSCCTRL_XOUT
88	-	PD02	EIC_NMI	-	-	-	EBI_A19	-	-	USB_SOF_1KHZ	-	-	PCC_DEN2	ETH_TSUCOMP	-	-	-
89	-	-	-	-	-	-	-	-	-	USB_DM	-	-	-	-	-	-	-
90	-	-	-	-	-	-	-	-	-	USB_DP	-	-	-	-	-	-	-

Table 5-8. 100-Pin TQFP Package Pinout (PIC32CK1025/2051 Devices) (continued)

Pin	Priority Function	MUXEN=0	MUXEN=1														External Oscillator I/O
			PMUX Values														
			0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7	0x8	0x9	0xa	0xb	0xc	0xd	
	RTC Security Tamper Input Loop(s)	Port	EIC	ADCN/ADC/AC	SERCOM	SERCOM Alt	EBI	TCC	TCC Alt/PDEC	SQI/CAN/USB/CM33 Trace	SDMMC	SPI_IXS	PCC	ETH	MISC	PTC/DEBUG	
91	-	PD12	EIC_EXTINT12	-	SERCOM0_PAD3	-	EBI_NCS0	TCC2_WO1	TCC2_WO7	SQI_CS3	SDMMC0_SDCD	-	-	-	GCLK_IO2	-	-
92	-	PD05	EIC_EXTINT5	-	SERCOM0_PAD2	-	EBI_NWE_NWR0	-	TCC2_WO0	SQI_CS1	SDMMC0_SDWP	-	-	-	GCLK_IO3	-	-
93	-	PD06	EIC_EXTINT6	-	SERCOM0_PAD0	-	EBI_A18	TCC2_WO4	TCC2_WO1	SQI_CS0	SDMMC0_SDCMD	SPI_IXS_FSYNC	-	ETH_MDIO	GCLK_IO4	-	-
94	-	PD07	EIC_EXTINT7	-	SERCOM0_PAD1	-	EBI_A17	TCC2_WO5	TCC2_WO2	SQI_DATA0	SDMMC0_SDDAT0	SPI_IXS_SDI	-	ETH_MDC	GCLK_IO5	-	-
95	-	VSS	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
96	-	VDDIO	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
97	-	PD08	EIC_EXTINT8	-	SERCOM1_PAD0	-	EBI_A16	-	TCC2_WO3	SQI_DATA2	SDMMC0_SDDAT2	SPI_IXS_SDO	-	-	LUT2_IN0	-	-
98	-	PD09	EIC_EXTINT9	-	SERCOM1_PAD1	-	EBI_NCS1	-	TCC2_WO4	SQI_DATA3	SDMMC0_SDDAT3	-	-	-	LUT2_IN1	-	-
99	-	PD10	EIC_EXTINT10	-	SERCOM1_PAD2	-	-	-	TCC2_WO5	SQI_CLK	SDMMC0_SDCK	SPI_IXS_SCK	-	-	LUT2_IN2	-	-
100	-	PD11	EIC_EXTINT11	-	SERCOM1_PAD3	-	EBI_NWAIT	TCC2_WO2	TCC2_WO6	SQI_DATA1	SDMMC0_SDDAT1	-	-	-	LUT2_OUT	-	-

Notes:
1. Pin column items displayed in **BOLD** are 5.5V tolerant pins.
2. PA02 and PA03 are dedicated SWDIO and SWCLK pins. They are not recommended for GPIO use.
3. VDDA and AVDD are used interchangeably.

5.4. 144-Pin Packages

5.4.1. 144-Pin TQFP Package

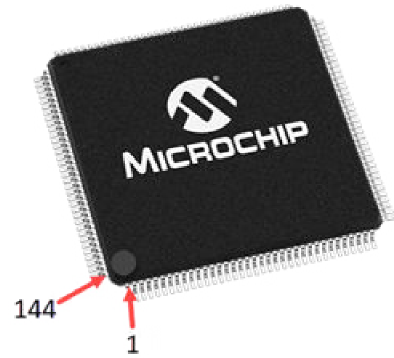
	Top View
PIC32CK(1025/2051)GC01144 PIC32CK(1025/2051)SG01144	

Table 5-9. 144-Pin TQFP Package Pinout

Pin	Priority Function	MUXEN=0	MUXEN=1 PMUX Values 0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7	0x8	0x9	0xa	0xb	0xc	0xd	External Oscillator I/O
	RTC Security Tamper Input Loop(s)	Port	EIC	ADCN/ADC/AC	SERCOM	SERCOM Alt	EBI	TCC	TCC Alt/PDEC	SQI/CAN/USB/CM33 Trace	SDMMC	SPI_IXS	PCC	ETH	MISC	PTC/DEBUG	
1	-	**PA00**	EIC_EXTINT14	-	-	SERCOM6_PAD0	-	TCC2_WO0	-	-	-	-	-	-	-	BSDAP_TDI	-
2	-	**PA01**	EIC_EXTINT1	-	SERCOM7_PAD0	SERCOM6_PAD1	-	TCC2_WO3	-	CPU0_SWO	-	-	-	-	GCLK_IO1	BSDAP_TDO	-
3	-	PA02 (2)	-	-	-	-	-	-	-	-	-	-	-	-	-	BSDAP_TMS_SWDIO	-
4	-	PA03(2)	-	-	-	-	-	-	-	-	-	-	-	-	-	BSDAP_TCK_SWCLK	-
5	-	RESET_N	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
6	-	VSS	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
7	-	VDDIO	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
8	-	PA20	EIC_EXTINT4	-	SERCOM7_PAD1	-	-	TCC4_WO0	PDEC_QDI0	-	-	-	PCC_CLK	-	GCLK_IO2	-	-
9	-	PA21	EIC_EXTINT5	-	SERCOM7_PAD0	-	-	TCC4_WO1	PDEC_QDI1	-	-	-	PCC_DATA0	-	GCLK_IO3	-	-
10	-	PA22	EIC_EXTINT6	-	SERCOM7_PAD2	-	-	TCC5_WO0	PDEC_QDI2	-	-	-	PCC_DATA1	-	GCLK_IO4	-	-
11	-	PA23	EIC_EXTINT7	-	SERCOM7_PAD3	-	-	TCC5_WO1	-	-	-	-	PCC_DATA2	-	GCLK_IO5	-	-
12	-	PA13	EIC_EXTINT13	-	SERCOM5_PAD0	-	EBI_D0	TCC4_WO0	-	CPU0_TRACEDATA3	-	-	-	-	LUT3_IN0	-	-
13	-	PA14	EIC_EXTINT14	-	SERCOM5_PAD1	-	EBI_D1	TCC4_WO1	-	CPU0_TRACEDATA2	-	-	-	-	LUT3_IN1	-	-
14	-	PA15	EIC_EXTINT15	-	SERCOM5_PAD2	-	EBI_D2	TCC5_WO0	-	CPU0_TRACEDATA1	-	-	-	-	LUT3_IN2	-	-
15	-	PA16	EIC_EXTINT0	-	SERCOM5_PAD3	-	EBI_D3	TCC5_WO1	-	CPU0_TRACECLK	-	-	-	-	LUT3_OUT	-	-
16	-	PA17	EIC_EXTINT1	-	-	-	EBI_D4	-	-	CPU0_TRACEDATA0	-	-	-	-	GCLK_IO6	-	-
17	-	PA24	EIC_EXTINT8	-	SERCOM5_PAD0	-	-	-	-	-	-	-	PCC_DATA3	-	GCLK_IO7	-	-
18	-	VSS	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
19	-	VDDIO	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
20	-	PA25	EIC_EXTINT9	-	SERCOM5_PAD1	-	-	-	-	-	-	-	PCC_DATA4	-	-	-	-
21	-	PA26	EIC_EXTINT10	-	SERCOM5_PAD2	-	-	-	-	-	-	-	PCC_DATA5	-	-	-	-
22	-	PA19	EIC_EXTINT3	-	SERCOM5_PAD3	SERCOM6_PAD2	-	TCC3_WO7	-	-	-	-	PCC_DATA6	-	-	PTC_ECI1	-

Table 5-9. 144-Pin TQFP Package Pinout (continued)

Pin	Priority Function	MUXEN=0	MUXEN=1														External Oscillator I/O
			PMUX Values														
			0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7	0x8	0x9	0xa	0xb	0xc	0xd	
	RTC Security Tamper Input Loop(s)	Port	EIC	ADCN/ADC/AC	SERCOM	SERCOM Alt	EBI	TCC	TCC Alt/PDEC	SQI/CAN/USB/CM33 Trace	SDMMC	SPI_IXS	PCC	ETH	MISC	PTC/DEBUG	
23	-	PA04	EIC_EXTINT4	ADC_VINP0	SERCOM7_PAD1	-	EBI_D5	TCC3_WO0	-	-	-	-	-	-	LUT3_IN0	-	-
24	-	PA05	EIC_EXTINT5	ADC_VINM0/ADC_VINP1	SERCOM7_PAD2	SERCOM5_PAD0	EBI_D6	TCC3_WO1	-	-	-	-	-	-	LUT3_IN1	-	-
25	-	PA06	EIC_EXTINT6	AC_AIN0/ADC_VINP2	SERCOM7_PAD3	SERCOM5_PAD1	EBI_D7	TCC3_WO2	-	-	-	-	-	-	LUT3_IN2	-	-
26	-	PA07	EIC_EXTINT7	AC_AIN1/ADC_VINM2/ADC_VINP3	SERCOM6_PAD0	SERCOM5_PAD2	EBI_D8	TCC3_WO3	-	-	-	-	-	-	LUT3_OUT	-	-
27	-	PA08	EIC_EXTINT8	ADC_VINP4	SERCOM6_PAD1	SERCOM5_PAD3	EBI_D9	TCC3_WO4	-	-	-	-	-	-	-	-	-
28	-	PA18	EIC_EXTINT2	ADC_VINM4	-	SERCOM6_PAD0	-	TCC3_WO5	-	-	-	-	-	-	-	PTC_ECI0	-
29	-	PA28	EIC_EXTINT12	-	-	SERCOM6_PAD1	-	TCC3_WO6	-	-	-	-	PCC_DATA7	-	-	-	-
30	-	VSS	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
31	-	VDDIO	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
32	-	PA19	EIC_EXTINT3	-	-	SERCOM6_PAD2	-	TCC3_WO7	-	-	-	-	-	-	-	PTC_ECI1	-
33	-	PA09	EIC_EXTINT9	ADC_VINP5	SERCOM6_PAD2	SERCOM6_PAD3	-	-	TCC3_WO0	-	-	SPI_IXS_FSYNC	-	-	GCLK_IO0	PTC_DRV0	-
34	-	PA10	EIC_EXTINT10	ADC_VINP6	SERCOM6_PAD3	-	EBI_NWAIT	-	TCC3_WO1	SQI_CS2	SDMMC0_SDCD	-	PCC_DEN1	-	GCLK_IO1	PTC_DRV1	-
35	-	PA11	EIC_EXTINT11	ADC_VINP7	SERCOM5_PAD0	SERCOM7_PAD1	-	TCC4_WO0	TCC3_WO2	-	-	SPI_IXS_SCK	-	ETH_MDC	AC_CMP0	PTC_DRV2	-
36	-	PA12	EIC_EXTINT12	ADC_VINP8	SERCOM5_PAD1	SERCOM7_PAD0	-	TCC4_WO1	TCC3_WO3	-	SDMMC0_SDCD	SPI_IXS_SDI	-	ETH_MDIO	AC_CMP1	PTC_DRV3	-
37	RTC_OUT1	PB00	EIC_EXTINT0	ADC_VINP9	SERCOM5_PAD2	SERCOM7_PAD2	-	TCC5_WO0	TCC3_WO4	CAN1_TX	SDMMC0_SDWP	SPI_IXS_SDO	-	-	-	PTC_DRV4	-
38	RTC_IN1	PB01	EIC_EXTINT1	ADC_VREFH	SERCOM5_PAD3	SERCOM7_PAD3	-	TCC5_WO1	TCC3_WO5	CAN1_RX	-	-	-	-	-	PTC_DRV5	-
39	RTC_OUT2	PB11	EIC_EXTINT11	ADC_VINP10	-	-	EBI_D10	-	TCC3_WO6	-	-	-	-	-	-	-	-
40	RTC_OUT3	PB12	EIC_EXTINT12	ADC_VINP11	SERCOM4_PAD0	SERCOM7_PAD1	EBI_D11	TCC4_WO0	TCC3_WO7	-	-	-	-	-	-	-	-
41	RTC_OUT5	PB18	EIC_EXTINT2	-	SERCOM4_PAD1	SERCOM7_PAD0	-	TCC4_WO1	-	-	-	-	PCC_DATA8	-	-	-	-
42	RTC_IN5	PB19	EIC_EXTINT3	-	SERCOM4_PAD2	SERCOM7_PAD2	-	TCC5_WO0	-	-	-	-	PCC_DATA9	-	-	-	-

Table 5-9. 144-Pin TQFP Package Pinout (continued)

Pin	Priority Function	MUXEN=0	MUXEN=1 PMUX Values 0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7	0x8	0x9	0xa	0xb	0xc	0xd	External Oscillator I/O
	RTC Security Tamper Input Loop(s)	Port	EIC	ADCN/ADC/AC	SERCOM	SERCOM Alt	EBI	TCC	TCC Alt/PDEC	SQI/CAN/USB/CM33 Trace	SDMMC	SPI_IXS	PCC	ETH	MISC	PTC/DEBUG	
43	-	PB20	EIC_EXTINT4	-	SERCOM4_PAD3	SERCOM7_PAD3	-	TCC5_WO1	-	-	-	-	PCC_DATA10	-	-	-	-
44	-	AVSS	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
45	-	PB02	EIC_EXTINT2	-	-	-	-	-	-	-	-	-	-	-	-	-	OSC32KCTRL_XIN32
46	-	PB03	EIC_EXTINT3	-	-	-	-	-	-	-	-	-	-	-	-	-	OSC32KCTRL_XOUT32
47	-	AVDD	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
48	-	VDDIO	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
49	-	VSS	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
50	-	PB04	EIC_EXTINT4	-	SERCOM4_PAD0	-	-	TCC0_WO0	PDEC_QDI0	-	-	-	-	-	-	-	-
51	RTC_OUT0	PB05	EIC_EXTINT5	-	SERCOM4_PAD1	-	EBI_D12	TCC0_WO1	PDEC_QDI1	-	-	-	-	-	-	-	-
52	RTC_IN4	PB13	EIC_EXTINT13	-	SERCOM4_PAD2	-	EBI_D13	TCC0_WO2	PDEC_QDI2	-	-	-	-	-	-	-	-
53	RTC_IN0	PB06	EIC_EXTINT6	-	SERCOM4_PAD3	-	EBI_D14	TCC0_WO3	-	-	-	-	-	-	-	-	-
54	RTC_IN2	PB14	EIC_EXTINT14	-	-	-	EBI_D15	TCC0_WO4	-	-	-	-	-	-	-	-	-
55	RTC_IN3	PB15	EIC_EXTINT15	-	-	-	EBI_NRD	TCC0_WO5	-	-	-	-	-	-	-	-	-
56	-	VDDIO	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
57	-	VSS	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
58	-	VDDREG	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
59	RTC_OUT6	PB21	EIC_EXTINT5	-	SERCOM3_PAD0	-	EBI_A22	TCC0_WO6	-	CAN0_TX	-	-	-	-	-	-	-
60	RTC_IN6	PB22	EIC_EXTINT6	-	SERCOM3_PAD1	-	EBI_A21	TCC0_WO7	-	CAN0_RX	-	-	-	-	LUT0_IN0	-	-
61	RTC_OUT7	PB23	EIC_EXTINT7	-	SERCOM3_PAD2	-	EBI_A20	TCC6_WO0	-	CAN1_TX	-	-	-	-	LUT0_IN1	-	-
62	RTC_IN7	PB24	EIC_EXTINT8	-	SERCOM3_PAD3	-	EBI_A19	TCC6_WO1	-	CAN1_RX	-	-	-	-	LUT0_IN2	-	-
63	-	PB25	EIC_EXTINT9	-	-	-	EBI_A18	TCC7_WO0	-	-	-	-	-	-	LUT0_OUT	-	-
64	-	PB26	EIC_EXTINT10	-	-	-	-	TCC7_WO1	-	-	-	-	PCC_DATA11	-	-	-	-
65	RTC_OUT4	PB16	EIC_EXTINT0	-	SERCOM3_PAD0	-	-	TCC6_WO0	-	CAN0_TX	-	-	-	-	-	PTC_DRV24	-
66	-	VDDIO	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
67	-	VSS	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Table 5-9. 144-Pin TQFP Package Pinout (continued)

Pin	Priority Function	MUXEN=0	MUXEN=1														
			PMUX Values														
			0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7	0x8	0x9	0xa	0xb	0xc	0xd	
	RTC Security Tamper Input Loop(s)	Port	EIC	ADCN/ADC/AC	SERCOM	SERCOM Alt	EBI	TCC	TCC Alt/PDEC	SQI/CAN/USB/CM33 Trace	SDMMC	SPI_IXS	PCC	ETH	MISC	PTC/DEBUG	External Oscillator I/O
68	-	PB17	EIC_EXTINT1	-	SERCOM3_PAD1	-	-	TCC6_WO1	-	CAN0_RX	-	-	-	-	-	PTC_DRV25	-
69	-	PB07	EIC_EXTINT7	-	SERCOM3_PAD2	SERCOM4_PAD2	EBI_NBS1_NWR1	TCC7_WO0	PDEC_QDI0	-	SDMMC1_SDCMD	-	PCC_DATA3	-	LUT0_IN0	PTC_DRV6	-
70	-	PB08	EIC_EXTINT8	-	SERCOM3_PAD3	SERCOM4_PAD3	EBI_NBS0_A0	TCC7_WO1	PDEC_QDI1	CAN0_TX	-	-	PCC_DATA4	-	LUT0_IN1	PTC_DRV7	-
71	-	PB09	EIC_EXTINT9	-	SERCOM4_PAD0	SERCOM4_PAD1	EBI_A15	-	PDEC_QDI2	CAN0_RX	-	-	PCC_DATA5	-	LUT0_IN2	PTC_DRV8	-
72	-	PB10	EIC_EXTINT10	-	SERCOM4_PAD1	SERCOM4_PAD0	EBI_A14	-	TCC0_WO0	-	SDMMC1_SDDAT0	-	PCC_DATA6	ETH_MDC	LUT0_OUT	PTC_DRV9	-
73	-	PC00	EIC_EXTINT0	-	SERCOM4_PAD2	-	EBI_A13	-	TCC0_WO1	-	SDMMC1_SDDAT1	-	PCC_DATA7	ETH_REF_CLK / ETH_TX_CLK	GCLK_IO2	PTC_DRV10	-
74	-	PC01	EIC_EXTINT1	-	SERCOM4_PAD3	-	EBI_A12	TCC6_WO0	TCC0_WO2	USB_SOF_1KHZ	SDMMC1_SDDAT2	-	PCC_DATA0	-	GCLK_IO3	PTC_DRV11	-
75	-	PC02	EIC_EXTINT2	-	SERCOM3_PAD0	-	EBI_A11	TCC6_WO1	TCC0_WO3	CAN0_TX	SDMMC1_SDDAT3	-	PCC_DATA1	ETH_MDIO	GCLK_IO4	PTC_DRV12	-
76	-	PC03	EIC_EXTINT3	-	SERCOM3_PAD1	-	EBI_A10	TCC7_WO0	TCC0_WO4	CAN0_RX	SDMMC1_SDWP	-	PCC_DATA2	ETH_MDC	GCLK_IO5	PTC_DRV13	-
77	-	PC04	EIC_EXTINT4	-	SERCOM3_PAD2	-	EBI_A9	TCC7_WO1	TCC0_WO5	-	SDMMC1_SDCD	-	PCC_DEN1	ETH_MDIO	GCLK_IO6	PTC_DRV14	-
78	-	VDDIO	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
79	-	VSS	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
80	-	PC21	EIC_EXTINT5	-	SERCOM2_PAD0	-	-	-	TCC0_WO6	-	-	-	PCC_DATA12	ETH_MDIO	AC_CMP0	-	-
81	-	PC22	EIC_EXTINT6	-	SERCOM2_PAD1	-	EBI_NWAIT	-	TCC0_WO7	-	-	-	PCC_DATA13	-	AC_CMP1	-	-
82	-	PC14	EIC_EXTINT14	-	SERCOM2_PAD2	-	EBI_A8	-	-	-	-	-	PCC_DEN2	ETH_RXD3	-	PTC_DRV26	-
83	-	PC23	EIC_EXTINT7	-	SERCOM2_PAD3	-	EBI_NCS1	-	-	-	-	-	-	-	-	-	-
84	-	PC05	EIC_EXTINT5	-	SERCOM3_PAD3	-	EBI_A7	-	TCC1_WO0	-	SDMMC1_SDCK	-	PCC_CLK	ETH_RXD2	GCLK_IO7	PTC_DRV15	-
85	-	PC06	EIC_EXTINT6	AC_AIN2	SERCOM2_PAD0	SERCOM3_PAD1	-	-	TCC1_WO1	CAN1_TX	-	-	-	ETH_RXD1	-	PTC_DRV16	-
86	-	PC07	EIC_EXTINT7	AC_AIN3	SERCOM2_PAD1	SERCOM3_PAD0	-	-	TCC1_WO2	CAN1_RX	-	-	-	ETH_RXD0	-	PTC_DRV17	-
87	-	PC24	EIC_EXTINT8	-	-	-	EBI_NWE_NWR0	-	-	-	-	-	-	-	LUT1_IN0	-	-

Table 5-9. 144-Pin TQFP Package Pinout (continued)

Pin	Priority Function	MUXEN=0	MUXEN=1														External Oscillator I/O
			PMUX Values														
			0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7	0x8	0x9	0xa	0xb	0xc	0xd	
	RTC Security Tamper Input Loop(s)	Port	EIC	ADCN/ADC/AC	SERCOM	SERCOM Alt	EBI	TCC	TCC Alt/PDEC	SQI/CAN/USB/CM33 Trace	SDMMC	SPI_IXS	PCC	ETH	MISC	PTC/DEBUG	
88	-	PC25	EIC_EXTINT9	-	-	-	EBI_NCS0	-	-	-	-	-	-	-	LUT1_IN1	-	-
89	-	VDDREG	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
90	-	VDDIO	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
91	-	VSS	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
92	-	PC26	EIC_EXTINT10	-	SERCOM2_PAD3	-	EBI_A17	-	-	-	-	-	-	-	LUT1_IN2	-	-
93	-	PC27	EIC_EXTINT11	-	-	-	EBI_A16	-	-	-	-	-	-	-	LUT1_OUT	-	-
94	-	PC09	EIC_EXTINT9	-	SERCOM2_PAD2	SERCOM3_PAD2	-	-	TCC1_WO3	-	-	-	-	ETH_RXER	AC_CMP0	PTC_DRV19	-
95	-	PC15	EIC_EXTINT15	-	SERCOM2_PAD0	SERCOM3_PAD3	EBI_A6	TCC1_WO0	TCC1_WO4	-	-	-	PCC_DATA8	ETH_RX_CLK	AC_CMP1	PTC_DRV27	-
96	-	PC28	EIC_EXTINT12	-	-	-	EBI_A23	TCC1_WO6	-	-	-	-	-	-	-	-	-
97	-	PC29	EIC_EXTINT13	-	-	-	EBI_NCS2	TCC1_WO7	-	-	-	-	-	-	-	-	-
98	-	PC16	EIC_EXTINT0	-	SERCOM2_PAD1	-	EBI_A5	TCC1_WO1	-	-	-	-	PCC_DATA9	ETH_TXER	-	PTC_DRV28	-
99	-	PC17	EIC_EXTINT1	-	SERCOM2_PAD2	-	EBI_A4	TCC1_WO2	-	-	-	-	PCC_DATA10	ETH_CRS	-	PTC_DRV29	-
100	-	PC18	EIC_EXTINT2	-	SERCOM2_PAD3	-	EBI_A3	TCC1_WO3	-	-	-	-	PCC_DATA11	ETH_COL	-	PTC_DRV30	-
101	-	PC10	EIC_EXTINT10	-	SERCOM0_PAD0	SERCOM2_PAD1	-	-	TCC1_WO5	-	-	-	-	ETH_RXDV	LUT1_IN0	PTC_DRV20	-
102	-	VDDIO	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
103	-	VSS	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
104	-	PC11	EIC_EXTINT11	-	SERCOM0_PAD1	SERCOM2_PAD0	-	-	TCC1_WO6	-	-	-	-	ETH_TXEN	LUT1_IN1	PTC_DRV21	-
105	-	PC12	EIC_EXTINT12	-	SERCOM0_PAD2	SERCOM2_PAD2	-	-	TCC1_WO7	-	-	-	-	ETH_TXD0	LUT1_IN2	PTC_DRV22	-
106	-	PC13	EIC_EXTINT13	-	SERCOM0_PAD3	SERCOM2_PAD3	-	-	-	-	-	-	-	ETH_TXD1	LUT1_OUT	PTC_DRV18	-
107	-	PC19	EIC_EXTINT3	-	-	-	EBI_A2	TCC1_WO4	-	-	-	-	PCC_DATA12	ETH_TXD2	-	PTC_DRV31	-
108	-	PC20	EIC_EXTINT4	-	-	-	EBI_NBS2_NWR2_A1	TCC1_WO5	-	-	-	-	PCC_DATA13	ETH_TXD3	-	PTC_DRV23	-
109	-	USB0PN	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
110	-	USB0PP	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
111	-	USB0ID	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
112	-	VUSB3V3	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Table 5-9. 144-Pin TQFP Package Pinout (continued)

Pin	Priority Function	MUXEN=0	MUXEN=1														External Oscillator I/O
			PMUX Values														
			0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7	0x8	0x9	0xa	0xb	0xc	0xd	
	RTC Security Tamper Input Loop(s)	Port	EIC	ADCN/ADC/AC	SERCOM	SERCOM Alt	EBI	TCC	TCC Alt/PDEC	SQI/CAN/USB/CM33 Trace	SDMMC	SPI_IXS	PCC	ETH	MISC	PTC/DEBUG	
113	-	USBVSSA	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
114	-	VBUS	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
115	-	USRBIAS	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
116	-	VDDREG_USB	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
117	-	VSS	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
118	-	VDDIO	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
119	-	PD00	EIC_EXTINT0	-	-	-	-	-	-	-	-	-	-	-	-	-	OSCCTRL_XIN
120	-	PD01	EIC_EXTINT1	-	-	-	-	-	-	-	-	-	-	-	-	-	OSCCTRL_XOUT
121	-	PD02	EIC_NMI	-	-	-	EBI_A19	-	-	USB_SOF_1KHZ	-	-	PCC_DEN2	ETH_TSUCOMP	-	-	-
122	-	-	-	-	-	-	-	-	-	USB_DM	-	-	-	-	-	-	-
123	-	-	-	-	-	-	-	-	-	USB_DP	-	-	-	-	-	-	-
124	-	PD13	EIC_EXTINT13	-	SERCOM1_PAD0	-	-	TCC2_WO6	-	-	-	SPI_IXS_FSYNC	-	-	-	-	-
125	-	PD14	EIC_EXTINT14	-	SERCOM1_PAD1	-	-	TCC2_WO7	-	-	-	SPI_IXS_SCK	-	-	-	-	-
126	-	PD15	EIC_EXTINT15	-	SERCOM1_PAD2	-	EBI_NCS3	-	-	-	-	-	-	-	-	-	-
127	-	PD16	EIC_EXTINT0	-	SERCOM1_PAD3	-	-	-	-	-	-	SPI_IXS_SDI	-	ETH_TSUCOMP	-	-	-
128	-	VSS	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
129	-	VDDIO	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
130	-	PD17	EIC_EXTINT1	-	SERCOM0_PAD0	-	-	-	-	-	-	SPI_IXS_SDO	-	-	LUT2_IN0	-	-
131	-	PD18	EIC_EXTINT2	-	SERCOM0_PAD1	-	-	-	-	CAN0_TX	-	-	-	-	LUT2_IN1	-	-
132	-	PD19	EIC_EXTINT3	-	SERCOM0_PAD2	-	-	-	-	CAN0_RX	-	-	PCC_DEN1	-	LUT2_IN2	-	-
133	-	PD20	EIC_EXTINT4	-	SERCOM0_PAD3	-	-	-	-	CAN1_TX	-	-	PCC_DEN2	-	LUT2_OUT	-	-
134	-	PD21	EIC_EXTINT5	-	-	-	-	-	-	CAN1_RX	-	-	-	-	-	-	-
135	-	PD12	EIC_EXTINT12	-	SERCOM0_PAD3	-	EBI_NCS0	TCC2_WO1	TCC2_WO7	SQI_CS3	SDMMC0_SDCD	-	-	-	GCLK_IO2	-	-
136	-	PD05	EIC_EXTINT5	-	SERCOM0_PAD2	-	EBI_NWE_NWR0	-	TCC2_WO0	SQI_CS1	SDMMC0_SDWP	-	-	-	GCLK_IO3	-	-

Table 5-9. 144-Pin TQFP Package Pinout (continued)

Pin	Priority Function	MUXEN=0	MUXEN=1														External Oscillator I/O
			PMUX Values														
			0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7	0x8	0x9	0xa	0xb	0xc	0xd	
	RTC Security Tamper Input Loop(s)	Port	EIC	ADCN/ADC/AC	SERCOM	SERCOM Alt	EBI	TCC	TCC Alt/PDEC	SQI/CAN/USB/CM33 Trace	SDMMC	SPI_IXS	PCC	ETH	MISC	PTC/DEBUG	
137	-	PD06	EIC_EXTINT6	-	SERCOM0_PAD0	-	EBI_A18	TCC2_WO4	TCC2_WO1	SQI_CS0	SDMMC0_SDCMD	SPI_IXS_FSYNC	-	ETH_MDIO	GCLK_IO4	-	-
138	-	PD07	EIC_EXTINT7	-	SERCOM0_PAD1	-	EBI_A17	TCC2_WO5	TCC2_WO2	SQI_DATA0	SDMMC0_SDDAT0	SPI_IXS_SDI	-	ETH_MDC	GCLK_IO5	-	-
139	-	VSS	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
140	-	VDDIO	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
141	-	PD08	EIC_EXTINT8	-	SERCOM1_PAD0	-	EBI_A16	-	TCC2_WO3	SQI_DATA2	SDMMC0_SDDAT2	SPI_IXS_SDO	-	-	LUT2_IN0	-	-
142	-	PD09	EIC_EXTINT9	-	SERCOM1_PAD1	-	EBI_NCS1	-	TCC2_WO4	SQI_DATA3	SDMMC0_SDDAT3	-	-	-	LUT2_IN1	-	-
143	-	PD10	EIC_EXTINT10	-	SERCOM1_PAD2	-	-	-	TCC2_WO5	SQI_CLK	SDMMC0_SDCK	SPI_IXS_SCK	-	-	LUT2_IN2	-	-
144	-	PD11	EIC_EXTINT11	-	SERCOM1_PAD3	-	EBI_NWAIT	TCC2_WO2	TCC2_WO6	SQI_DATA1	SDMMC0_SDDAT1	-	-	-	LUT2_OUT	-	-

Notes:
1. Pin column items displayed in **BOLD** are 5.5V tolerant pins.
2. PA02 and PA03 are dedicated SWDIO and SWCLK pins. They are not recommended for GPIO use.
3. VDDA and AVDD are used interchangeably.

5.4.2. 144-Pin TFBGA Package

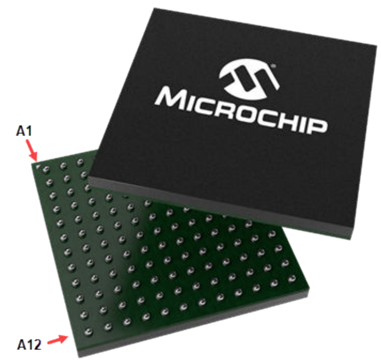
PIC32CK(1025/2051)GC01144 PIC32CK(1025/2051)SG01144	**Top View** 

Table 5-10. 144-pin TFBGA Thin Fine Pitch Ball Grid Array

Pin	SUPC	Priority Function	MUXEN=0	PMUX Values MUXEN=1														XOSC I/O
				0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7	0x8	0x9	0xA	0xB	0xC	0xD	
		RTC Security Tamper	PORT	EIC	ADCn/ADC/AC	SERCOM	SERCOM Alt	EBI	TCC	TCC Alt/PDEC	SQI/CAN/USB/CM33 Trace	SDMMC	SPI_IXS	PCC	ETH	MISC	PTC/DEBUG	
A1	-	-	VSS	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
A2	-	-	PC13	EIC_EXTINT13	-	SERCOM0_PAD3	SERCOM2_PAD3	-	-	-	-	-	-	-	ETH_TXD1	CCL_OUT1	PTC_DRV18	-
A3	-	-	PC12	EIC_EXTINT12	-	SERCOM0_PAD2	SERCOM2_PAD2	-	-	TCC1_WO7	-	-	-	-	ETH_TXD0	CCL_IN5	PTC_DRV22	-
A4	-	-	PC11	EIC_EXTINT11	-	SERCOM0_PAD1	SERCOM2_PAD0	-	-	TCC1_WO6	-	-	-	-	ETH_TXEN	CCL_IN4	PTC_DRV21	-
A5	-	-	PC18	EIC_EXTINT2	-	SERCOM2_PAD3	-	EBI_A3_	TCC1_WO3	-	-	-	-	PCC_DATA11	ETH_COL	-	PTC_DRV30	-
A6	-	-	PC17	EIC_EXTINT1	-	SERCOM2_PAD2	-	EBI_A4_	TCC1_WO2	-	-	-	-	PCC_DATA10	ETH_CRS	-	PTC_DRV29	-
A7	-	-	PC28	EIC_EXTINT12	-	-	-	EBI_A23_	TCC1_WO6	-	-	-	-	-	-	-	-	-
A8	-	-	PC15	EIC_EXTINT15	-	SERCOM2_PAD0	SERCOM3_PAD3	EBI_A6_	TCC1_WO0	TCC1_WO4	-	-	-	PCC_DATA8	ETH_RX_CLK	AC_CMP1	PTC_DRV27	-
A9	-	-	PC27	EIC_EXTINT11	-	-	-	EBI_A16_	-	-	-	-	-	-	-	CCL_OUT1	-	-
A10	-	-	PC26	EIC_EXTINT10	-	SERCOM2_PAD3	-	EBI_A17_	-	-	-	-	-	-	-	CCL_IN5	-	-
A11	-	-	PC24	EIC_EXTINT8	-	-	-	EBI_NWE_NWR0	-	-	-	-	-	-	-	CCL_IN3	-	-
A12	-	-	VSS	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
B1	-	-	USB0PN	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
B2	-	-	VUSB3V3	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
B3	-	-	PC19	EIC_EXTINT3	-	-	-	EBI_A2_	TCC1_WO4	-	-	-	-	PCC_DATA12	ETH_TXD2	-	PTC_DRV31	-
B4	-	-	PC20	EIC_EXTINT4	-	-	-	EBI_NBS2_NWR2_A1_	TCC1_WO5	-	-	-	-	PCC_DATA13	ETH_TXD3	-	PTC_DRV23	-
B5	-	-	PC10	EIC_EXTINT10	-	SERCOM0_PAD0	SERCOM2_PAD1	-	-	TCC1_WO5	-	-	-	-	ETH_RXDV	CCL_IN3	PTC_DRV20	-
B6	-	-	PC16	EIC_EXTINT0	-	SERCOM2_PAD1	-	EBI_A5_	TCC1_WO1	-	-	-	-	PCC_DATA9	ETH_TXER	-	PTC_DRV28	-
B7	-	-	PC29	EIC_EXTINT13	-	-	-	EBI_NCS2	TCC1_WO7	-	-	-	-	-	-	-	-	-
B8	-	-	PC09	EIC_EXTINT9	-	SERCOM2_PAD2	SERCOM3_PAD2	-	-	TCC1_WO3	-	-	-	-	ETH_RXER	AC_CMP0	PTC_DRV19	-
B9	-	-	PC05	EIC_EXTINT5	-	SERCOM3_PAD3	-	EBI_A7_	-	TCC1_WO0	-	SDMMC1_SDCK	-	PCC_CLK	ETH_RXD2	GCLK_IO7	PTC_DRV15	-

Table 5-10. 144-pin TFBGA Thin Fine Pitch Ball Grid Array (continued)

Pin	SUPC	Priority Function	MUXEN=0	PMUX Values														XOSC I/O
				MUXEN=1														
				0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7	0x8	0x9	0xA	0xB	0xC	0xD	
		RTC Security Tamper	PORT	EIC	ADCn/ADC/AC	SERCOM	SERCOM Alt	EBI	TCC	TCC Alt/PDEC	SQI/CAN/USB/CM33 Trace	SDMMC	SPI_IXS	PCC	ETH	MISC	PTC/DEBUG	
B10	-	-	PC25	EIC_ EXTINT9	-	-	-	EBI_NCS0	-	-	-	-	-	-	-	CCL_IN4	-	-
B11	-	-	PC04	EIC_ EXTINT4	-	SERCOM3_PAD2	-	EBI_A9_	TCC7_WO1	TCC0_WO5	-	SDMMC1_ SDCD	-	PCC_DEN1	ETH_MDIO	GCLK_IO6	PTC_DRV14	-
B12	-	-	PC03	EIC_ EXTINT3	-	SERCOM3_PAD1	-	EBI_A10_	TCC7_WO0	TCC0_WO4	CAN0_RX	SDMMC1_ SDWP	-	PCC_DATA2	ETH_MDC	GCLK_IO5	PTC_DRV13	-
C1	-	-	USB0PP	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
C2	-	-	USBID	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
C3	-	-	PD14	EIC_ EXTINT14	-	SERCOM1_PAD1	-	-	TCC2_WO7	-	-	-	SPI_IXS_SCK	-	-	-	-	-
C4	-	-	PC14	EIC_ EXTINT14	-	SERCOM2_PAD2	-	EBI_A8_	-	-	-	-	-	PCC_DEN2	ETH_RXD3	-	PTC_DRV26	-
C5	-	-	PC23	EIC_ EXTINT7	-	SERCOM2_PAD3	-	EBI_NCS1	-	-	-	-	-	-	-	-	-	-
C6	-	-	PC21	EIC_ EXTINT5	-	SERCOM2_PAD0	-	-	-	TCC0_WO6	-	-	-	PCC_DATA12	ETH_MDIO	AC_CMP0	-	-
C7	-	-	PC22	EIC_ EXTINT6	-	SERCOM2_PAD1	-	EBI_NWAIT	-	TCC0_WO7	-	-	-	PCC_DATA13	-	AC_CMP1	-	-
C8	-	RTC_IN7	PB24	EIC_ EXTINT8	-	SERCOM3_PAD3	-	EBI_A19_	TCC6_WO1	-	CAN1_RX	-	-	-	-	CCL_IN2	-	-
C9	-	RTC_IN6	PB22	EIC_ EXTINT6	-	SERCOM3_PAD1	-	EBI_A21_	TCC0_WO7	-	CAN0_RX	-	-	-	-	CCL_IN0	-	-
C10	-	RTC_OUT4	PB16	EIC_ EXTINT0	-	SERCOM3_PAD0	-	-	TCC6_WO0	-	CAN0_TX	-	-	-	-	-	PTC_DRV24	-
C11	-	-	PC01	EIC_ EXTINT1	-	SERCOM4_PAD3	-	EBI_A12_	TCC6_WO0	TCC0_WO2	USB_SOF_ 1KHZ	SDMMC1_ SDDAT2	-	PCC_DATA0	-	GCLK_IO3	PTC_DRV11	-
C12	-	-	PC02	EIC_ EXTINT2	-	SERCOM3_PAD0	-	EBI_A11_	TCC6_WO1	TCC0_WO3	CAN0_TX	SDMMC1_ SDDAT3	-	PCC_DATA1	ETH_MDIO	GCLK_IO4	PTC_DRV12	-
D1	-	-	VBUS	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
D2	-	-	USBRBIAS	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
D3	-	-	PD02	EIC_NMI	-	-	-	EBI_A19_	-	-	USB_SOF_ 1KHZ	-	-	PCC_DEN2	ETH_TSUCOMP	-	-	-
D4	-	-	USBVSSA	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
D5	-	-	VDDIO	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
D6	-	-	VDDIO	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
D7	-	-	PC07	EIC_ EXTINT7	AC_AIN3	SERCOM2_PAD1	SERCOM3_PAD0	-	-	TCC1_WO2	CAN1_RX	-	-	-	ETH_RXD0	-	PTC_DRV17	-

Table 5-10. 144-pin TFBGA Thin Fine Pitch Ball Grid Array (continued)

Pin	SUPC	Priority Function	MUXEN=0	PMUX Values MUXEN=1 0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7	0x8	0x9	0xA	0xB	0xC	0xD	XOSC I/O
		RTC Security Tamper	PORT	EIC	ADCn/ADC/AC	SERCOM	SERCOM Alt	EBI	TCC	TCC Alt/PDEC	SQI/CAN/USB/CM33 Trace	SDMMC	SPI_IXS	PCC	ETH	MISC	PTC/DEBUG	
D8	-	-	PC06	EIC_ EXTINT6	AC_AIN2	SERCOM2_PAD0	SERCOM3_PAD1	-	-	TCC1_WO1	CAN1_TX	-	-	-	ETH_RXD1	-	PTC_DRV16	-
D9	-	-	VDDIO	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
D10	-	RTC_OUT7	PB23	EIC_ EXTINT7	-	SERCOM3_PAD2	-	EBI_A20_	TCC6_WO0	-	CAN1_TX	-	-	-	-	CCL_IN1	-	-
D11	-	-	PC00	EIC_ EXTINT0	-	SERCOM4_PAD2	-	EBI_A13_	-	TCC0_WO1	-	SDMMC1_ SDDAT1	-	PCC_DATA7	ETH_TX_CLK/ ETH_REF_CLK	GCLK_IO2	PTC_DRV10	-
D12	-	-	PB10	EIC_ EXTINT10	-	SERCOM4_PAD1	SERCOM4_PAD0	EBI_A14_	-	TCC0_WO0	-	SDMMC1_ SDDAT0	-	PCC_DATA6	ETH_MDC	CCL_OUT0	PTC_DRV9	-
E1	-	-	PD00	EIC_ EXTINT0	-	-	-	-	-	-	-	-	-	-	-	-	-	OSCCTRL_ XIN
E2	-	-	PD16	EIC_ EXTINT0	-	SERCOM1_PAD3	-	-	-	-	-	-	SPI_IXS_SDI	-	ETH_TSUCOMP	-	-	-
E3	-	-	PD13	EIC_ EXTINT13	-	SERCOM1_PAD0	-	-	TCC2_WO6	-	-	-	SPI_IXS_FSYNC	-	-	-	-	-
E4	-	-	VDDREG_USB	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
E5	-	-	VSS	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
E6	-	-	VSS	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
E7	-	-	VDDREG	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
E8	-	-	VSS	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
E9	-	-	VDDIO	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
E10	-	RTC_OUT6	PB21	EIC_ EXTINT5	-	SERCOM3_PAD0	-	EBI_A22_	TCC0_WO6	-	CAN0_TX	-	-	-	-	-	-	-
E11	-	-	PB08	EIC_ EXTINT8	-	SERCOM3_PAD3	SERCOM4_PAD3	EBI_NBS0_ A0	TCC7_WO1	PDEC_QDI1	CAN0_TX	-	-	PCC_DATA4	-	CCL_IN1	PTC_DRV7	-
E12	-	-	PB09	EIC_ EXTINT9	-	SERCOM4_PAD0	SERCOM4_PAD1	EBI_A15_	-	PDEC_QDI2	CAN0_RX	-	-	PCC_DATA5	-	CCL_IN2	PTC_DRV8	-
F1	-	-	PD01	EIC_ EXTINT1	-	-	-	-	-	-	-	-	-	-	-	-	-	OSCCTRL_ XOUT
F2	-	-	PD17	EIC_ EXTINT 1	-	SERCOM0_PAD0	-	-	-	-	-	-	SPI_IXS_SDO	-	-	CCL_IN6	-	-
F3	-	-	PD15	EIC_ EXTINT15	-	SERCOM1_PAD2	-	EBI_NCS3	-	-	-	-	-	-	-	-	-	-
F4	-	-	VDDIO	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
F5	-	-	VSS	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
F6	-	-	AVSS	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
F7	-	-	VSS	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Table 5-10. 144-pin TFBGA Thin Fine Pitch Ball Grid Array (continued)

Pin	SUPC	Priority Function	MUXEN=0	PMUX Values MUXEN=1 0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7	0x8	0x9	0xA	0xB	0xC	0xD	XOSC I/O
		RTC Security Tamper	PORT	EIC	ADCn/ADC/AC	SERCOM	SERCOM Alt	EBI	TCC	TCC Alt/PDEC	SQI/CAN/USB/CM33 Trace	SDMMC	SPI_IXS	PCC	ETH	MISC	PTC/DEBUG	
F8	-	-	VDDIO	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
F9	-	-	VDDREG	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
F10	-	RTC_IN3	PB15	EIC_ EXTINT15	-	-	-	EBI_NRD	TCC0_WO5	-	-	-	-	-	-	-	-	-
F11	-	-	PB17	EIC_ EXTINT1	-	SERCOM3_PAD1	-	-	TCC6_WO1	-	CAN0_RX	-	-	-	-	-	PTC_DRV25	-
F12	-	-	PB07	EIC_ EXTINT7	-	SERCOM3_PAD2	SERCOM4_PAD2	EBI_NBS1_ NWR1	TCC7_WO0	PDEC_QDI0	-	SDMMC1_ SDCMD	-	PCC_DATA3	-	CCL_IN0	PTC_DRV6	-
G1	-	-	-	-	-	-	-	-	-	-	USB_DM	-	-	-	-	-	-	-
G2	-	-	PD19	EIC_ EXTINT3	-	SERCOM0_PAD2	-	-	-	-	CAN0_RX	-	-	PCC_DEN1	-	CCL_IN8	-	-
G3	-	-	PD18	EIC_ EXTINT2	-	SERCOM0_PAD1	-	-	-	-	CAN0_TX	-	-	-	-	CCL_IN7	-	-
G4	-	-	VDDIO	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
G5	-	-	VSS	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
G6	-	-	VSS	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
G7	-	-	VDDIO	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
G8	SUPC_OUT0	RTC_OUT0	PB05	EIC_ EXTINT5	-	SERCOM4_PAD1	-	EBI_D12_	TCC0_WO1	PDEC_QDI1	-	-	-	-	-	-	-	-
G9	-	RTC_IN2	PB14	EIC_ EXTINT14	-	-	-	EBI_D15_	TCC0_WO4	-	-	-	-	-	-	-	-	-
G10	SUPC_OUT1	RTC_IN4	PB13	EIC_ EXTINT13	-	SERCOM4_PAD2	-	EBI_D13_	TCC0_WO2	PDEC_QDI2	-	-	-	-	-	-	-	-
G11	-	-	PB25	EIC_ EXTINT9	-	-	-	EBI_A18_	TCC7_WO0	-	-	-	-	-	-	CCL_OUT0	-	-
G12	-	-	PB26	EIC_ EXTINT10	-	-	-	-	TCC7_WO1	-	-	-	-	PCC_DATA11	-	-	-	-
H1	-	-	-	-	-	-	-	-	-	-	USB_DP	-	-	-	-	-	-	-
H2	-	-	PD20	EIC_ EXTINT4	-	SERCOM0_PAD3	-	-	-	-	CAN1_TX	-	-	PCC_DEN2	-	CCL_OUT2	-	-
H3	-	-	PD08	EIC_ EXTINT8	-	SERCOM1_PAD0	-	EBI_A16_	-	TCC2_WO3	SQI_DATA2	SDMMC0_ SDDAT2	SPI_IXS_SDO	-	-	CCL_IN6	-	-
H4	-	-	VDDIO	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
H5	-	-	VSS	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
H6	-	-	VDDIO	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
H7	-	-	AVDD	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
H8	-	RTC_OUT5	PB18	EIC_ EXTINT2	-	SERCOM4_PAD1	SERCOM7_PAD0	-	TCC4_WO1	-	-	-	-	PCC_DATA8	-	-	-	-

Table 5-10. 144-pin TFBGA Thin Fine Pitch Ball Grid Array (continued)

Pin	SUPC	Priority Function	MUXEN=0	PMUX Values MUXEN=1														XOSC I/O
				0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7	0x8	0x9	0xA	0xB	0xC	0xD	
		RTC Security Tamper	PORT	EIC	ADCn/ADC/AC	SERCOM	SERCOM Alt	EBI	TCC	TCC Alt/PDEC	SQI/CAN/USB/CM33 Trace	SDMMC	SPI_IXS	PCC	ETH	MISC	PTC/DEBUG	
H9	-	RTC_IN0	PB06	EIC_ EXTINT6	-	SERCOM4_PAD3	-	EBI_D14_	TCC0_WO3	-	-	-	-	-	-	-	-	-
H10	-	RTC_IN5	PB19	EIC_ EXTINT3	-	SERCOM4_PAD2	SERCOM7_PAD2	-	TCC5_WO0	-	-	-	-	PCC_DATA9	-	-	-	-
H11	-	-	PB03	EIC_ EXTINT3	-	-	-	-	-	-	-	-	-	-	-	-	-	OSC32KCTRL_ XOUT32
H12	-	-	PB04	EIC_ EXTINT4	-	SERCOM4_PAD0	-	-	TCC0_WO0	PDEC_QDI0	-	-	-	-	-	-	-	-
J1	-	-	PD21	EIC_ EXTINT5	-	-	-	-	-	-	CAN1_RX	-	-	-	-	-	-	-
J2	-	-	PD12	EIC_ EXTINT12	-	SERCOM0_PAD3	-	EBI_NCS0	TCC2_WO1	TCC2_WO7	SQI_CS3	SDMMC0_ SDCD	-	-	-	GCLK_IO2	-	-
J3	-	-	PD11	EIC_ EXTINT11	-	SERCOM1_PAD3	-	EBI_NWAIT	TCC2_WO2	TCC2_WO6	SQI_DATA1	SDMMC0_ SDDAT1	-	-	-	CCL_OUT2	-	-
J4	-	-	PD09	EIC_ EXTINT9	-	SERCOM1_PAD1	-	EBI_NCS1	-	TCC2_WO4	SQI_DATA3	SDMMC0_ SDDAT3	-	-	-	CCL_IN7	-	-
J5	-	-	PD10	EIC_ EXTINT10	-	SERCOM1_PAD2	-	-	-	TCC2_WO5	SQI_CLK	SDMMC0_ SDCK	SPI_IXS_SCK	-	-	CCL_IN8	-	-
J6	-	-	PA16	EIC_ EXTINT0	-	SERCOM5_PAD3	-	EBI_D3_	TCC5_WO1	-	CPU0_ TRACECLK	-	-	-	-	CCL_OUT3	-	-
J7	-	-	VDDIO	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
J8	-	-	VDDIO	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
J9	-	RTC_OUT2	PB11	EIC_ EXTINT11	ADC_VIN P10	-	-	EBI_D10_	-	TCC3_WO6	-	-	-	-	-	-	-	-
J10	-	RTC_OUT3	PB12	EIC_ EXTINT12	ADC_VIN P11	SERCOM4_PAD0	SERCOM7_PAD1	EBI_D11_	TCC4_WO0	TCC3_WO7	-	-	-	-	-	-	-	-
J11	-	-	PB02	EIC_ EXTINT2	-	-	-	-	-	-	-	-	-	-	-	-	-	OSC32KCTRL_ XIN32
J12	-	-	PB20	EIC_ EXTINT4	-	SERCOM4_PAD3	SERCOM7_PAD3	-	TCC5_WO1	-	-	-	-	PCC_DATA10	-	-	-	-
K1	-	-	PD05	EIC_ EXTINT5	-	SERCOM0_PAD2	-	EBI_NWE_ NWR0	-	TCC2_WO0	SQI_CS1	SDMMC0_ SDWP	-	-	-	GCLK_IO3	-	-
K2	-	-	PD06	EIC_ EXTINT6	-	SERCOM0_PAD0	-	EBI_A18_	TCC2_WO4	TCC2_WO1	SQI_CS0	SDMMC0_ SDCMD	SPI_IXS_ FSYNC	-	ETH_MDIO	GCLK_IO4	-	-
K3	-	-	PA02[2]	EIC_ EXTINT2	-	-	SERCOM6_PAD2	-	-	-	-	-	-	-	-	-	BSDAP_TMS_ SWDIO	-

Table 5-10. 144-pin TFBGA Thin Fine Pitch Ball Grid Array (continued)

Pin	SUPC	Priority Function	MUXEN=0	PMUX Values MUXEN=1 0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7	0x8	0x9	0xA	0xB	0xC	0xD	XOSC I/O
		RTC Security Tamper	PORT	EIC	ADCn/ADC/AC	SERCOM	SERCOM Alt	EBI	TCC	TCC Alt/PDEC	SQI/CAN/USB/CM33 Trace	SDMMC	SPI_IXS	PCC	ETH	MISC	PTC/DEBUG	
K4	-	-	PA03[2]	EIC_EXTINT3	-	-	SERCOM6_PAD3	-	-	-	-	-	-	-	-	GCLK_IO0	BSDAP_TCK_SWCLK	-
K5	-	-	PA20	EIC_EXTINT4	-	SERCOM7_PAD1	-	-	TCC4_WO0	PDEC_QDI0	-	-	-	PCC_CLK	-	-	-	-
K6	-	-	PA15	EIC_EXTINT15	-	SERCOM5_PAD2	-	EBI_D2_	TCC5_WO0	-	CPU0_TRACEDATA1	-	-	-	-	CCL_IN11	-	-
K7	-	-	PA04	EIC_EXTINT4	ADC_VINP0	SERCOM7_PAD1	-	EBI_D5_	TCC3_WO0	-	-	-	-	-	-	CCL_IN9	-	-
K8	-	-	PA08	EIC_EXTINT8	ADC_VINP4	SERCOM6_PAD1	SERCOM5_PAD3	EBI_D9_	TCC3_WO4	-	-	-	-	-	-	-	-	-
K9	-	RTC_IN1	PB01	EIC_EXTINT1	ADC_VREFP0	SERCOM5_PAD3	SERCOM7_PAD3	-	TCC5_WO1	TCC3_WO5	CAN1_RX	-	-	-	-	-	PTC_DRV5	-
K10	-	RTC_OUT1	PB00	EIC_EXTINT0	ADC_VINP9	SERCOM5_PAD2	SERCOM7_PAD2	-	TCC5_WO0	TCC3_WO4	CAN1_TX	SDMMC0_SDWP	SPI_IXS_SDO	-	-	-	PTC_DRV4	-
K11	-	-	PA12	EIC_EXTINT12	ADC_VINP8	SERCOM5_PAD1	SERCOM7_PAD0	-	TCC4_WO1	TCC3_WO3	-	SDMMC0_SDCD	SPI_IXS_SDI	-	ETH_MDIO	AC_CMP1	PTC_DRV3	-
K12	-	-	PA11	EIC_EXTINT11	ADC_VINP7	SERCOM5_PAD0	SERCOM7_PAD1	-	TCC4_WO0	TCC3_WO2	-	-	SPI_IXS_SCK	-	ETH_MDC	AC_CMP0	PTC_DRV2	-
L1	-	-	PD07	EIC_EXTINT7	-	SERCOM0_PAD1	-	EBI_A17_	TCC2_WO5	TCC2_WO2	SQI_DATA0	SDMMC0_SDDAT0	SPI_IXS_SDI	-	ETH_MDC	GCLK_IO5	-	-
L2	-	-	**PA00**	EIC_EXTINT14	-	-	SERCOM6_PAD0	-	TCC2_WO0	-	-	-	-	-	-	-	BSDAP_TDI	-
L3	-	-	**PA01**	EIC_EXTINT1	-	SERCOM7_PAD0	SERCOM6_PAD1	-	TCC2_WO3	-	CPU0_SWO	-	-	-	-	GCLK_IO1	BSDAP_TDO	-
L4	-	-	RESET_N	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
L5	-	-	PA13	EIC_EXTINT13	-	SERCOM5_PAD0	-	EBI_D0_	TCC4_WO0	-	CPU0_TRACEDATA3	-	-	-	-	CCL_IN9	-	-
L6	-	-	PA24	EIC_EXTINT8	-	SERCOM5_PAD0	-	-	-	-	-	-	-	PCC_DATA3	-	GCLK_IO7	-	-
L7	-	-	PA26	EIC_EXTINT10	-	SERCOM5_PAD2	-	-	-	-	-	-	-	PCC_DATA5	-	-	-	-
L8	-	-	PA27	EIC_EXTINT11	-	SERCOM5_PAD3	-	-	-	-	-	-	-	PCC_DATA6	-	-	-	-
L9	-	-	PA28	EIC_EXTINT12	-	-	SERCOM6_PAD1	-	TCC3_WO6	-	-	-	-	PCC_DATA7	-	-	-	-
L10	-	-	PA19	EIC_EXTINT3	-	-	SERCOM6_PAD2	-	TCC3_WO7	-	-	-	-	-	-	-	PTC_ECI1	-

Table 5-10. 144-pin TFBGA Thin Fine Pitch Ball Grid Array (continued)

Pin	SUPC	Priority Function	MUXEN=0	PMUX Values														XOSC I/O
				MUXEN=1														
				0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7	0x8	0x9	0xA	0xB	0xC	0xD	
		RTC Security Tamper	PORT	EIC	ADCn/ADC/AC	SERCOM	SERCOM Alt	EBI	TCC	TCC Alt/PDEC	SQI/CAN/USB/CM33 Trace	SDMMC	SPI_IXS	PCC	ETH	MISC	PTC/DEBUG	
L11	-	-	PA10	EIC_EXTINT10	ADC_VINP6	SERCOM6_PAD3	-	EBI_NWAIT	-	TCC3_WO1	SQI_CS2	SDMMC0_SDCD	-	PCC_DEN1	-	GCLK_IO1	PTC_DRV1	-
L12	-	-	PA09	EIC_EXTINT9	ADC_VINP5	SERCOM6_PAD2	SERCOM6_PAD3	-	-	TCC3_WO0	-	-	SPI_IXS_FSYNC	-	-	GCLK_IO0	PTC_DRV0	-
M1	-	-	VSS	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
M2	-	-	PA21	EIC_EXTINT5	-	SERCOM7_PAD0	-	-	TCC4_WO1	PDEC_QDI1	-	-	-	PCC_DATA0	-	GCLK_IO3	-	-
M3	-	-	PA22	EIC_EXTINT6	-	SERCOM7_PAD2	-	-	TCC5_WO0	PDEC_QDI2	-	-	-	PCC_DATA1	-	GCLK_IO4	-	-
M4	-	-	PA23	EIC_EXTINT7	-	SERCOM7_PAD3	-	-	TCC5_WO1	-	-	-	-	PCC_DATA2	-	GCLK_IO5	-	-
M5	-	-	PA14	EIC_EXTINT14	-	SERCOM5_PAD1	-	EBI_D1_	TCC4_WO1	-	CPU0_TRACEDATA2	-	-	-	-	CCL_IN10	-	-
M6	-	-	PA17	EIC_EXTINT1	-	-	-	EBI_D4_	-	-	CPU0_TRACEDATA0	-	-	-	-	GCLK_IO6	-	-
M7	-	-	PA25	EIC_EXTINT9	-	SERCOM5_PAD1	-	-	-	-	-	-	-	PCC_DATA4	-	-	-	-
M8	-	-	PA05	EIC_EXTINT5	ADC_VINM0/ ADC_VINP1	SERCOM7_PAD2	SERCOM5_PAD0	EBI_D6_	TCC3_WO1	-	-	-	-	-	-	CCL_IN10	-	-
M9	-	-	PA06	EIC_EXTINT6	ADC_VINP2/ AC_AIN0	SERCOM7_PAD3	SERCOM5_PAD1	EBI_D7_	TCC3_WO2	-	-	-	-	-	-	CCL_IN11	-	-
M10	-	-	PA07	EIC_EXTINT7	ADC_VINM2/ ADC_VINP3/ AC_AIN1	SERCOM6_PAD0	SERCOM5_PAD2	EBI_D8_	TCC3_WO3	-	-	-	-	-	-	CCL_OUT3	-	-
M11	-	-	PA18	EIC_EXTINT2	ADC_VINM4	-	SERCOM6_PAD0	-	TCC3_WO5	-	-	-	-	-	-	-	PTC_ECI0	-
M12	-	-	VSS	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Notes:

1. Pin column items displayed in **BOLD** are 5.5V tolerant pins.
2. PA02 and PA03 are dedicated SWDIO and SWCLK pins. They are not recommended for GPIO use.
3. VDDA and AVDD are used interchangeably.

6. Signal Description

The following tables provide details on signal names classified by peripherals.

Note: IOSET Configurations – The PIC32CK device family has multiple peripheral instances, mapped to different I/O locations. Each peripheral I/O location is called IOSET and for a given peripheral, signals from different IOSET cannot be mixed.
For a given peripheral with two pads, PAD0 and PAD1:

- **Valid:** PAD0 and PAD1 in the same IOSETn
- **Invalid:** PAD0 in IOSETx and PAD1 in IOSETy

Notes:

1. Bolded pins are only available on PIC32CK1012 variant devices.
2. 64 TQFP package is identical to 64 VQFN package for signal description.

Table 6-1. Analog Comparator (AC) Signals

PIN NAME	PIN NUMBER					PIN TYPE	FUNCTION TYPE	DESCRIPTION
	48 VQFN	64 TQFP	100 TQFP	144 TQFP	144 TFBGA			
AC_AIN0	8/10	10/13	15/22	25	M9	I	Analog	Comparator 0 and 1 external analog inputs
AC_AIN1	9/11	11/14	16/23	26	M10	I	Analog	
AC_AIN2	33	41	60	85	D8	I	Analog	
AC_AIN3	34	42	61	86	D7	I	Analog	
AC_CMP0	12	15/44	24/63	35/80/94	B8/C6/K12	O	Analog	Comparator 0 Output
AC_CMP1	13	16	25/64	36/81/95	A8/C7/K11	O	Analog	Comparator 1 Output

Table 6-2. Analog-to-Digital Controller (ADC) Signals

PIN NAME	PIN NUMBER					PIN TYPE	FUNCTION TYPE	DESCRIPTION
	48 VQFN	64 TQFP	100 TQFP	144 TQFP	144 TFBGA			
ADC_VINM0	-	9	14	24	M8	I	Analog	ADC module analog inputs
ADC_VINM2	9	11	16	26	M10	I	Analog	
ADC_VINM4	-	-	18	28	M11	I	Analog	
ADC_VINP0	-	8	13	23	K7	I	Analog	
ADC_VINP1	-	9	14	24	M8	I	Analog	
ADC_VINP2	8	10	15	25	M9	I	Analog	
ADC_VINP3	9	11	16	26	M10	I	Analog	
ADC_VINP4	-	12	17	27	K8	I	Analog	
ADC_VINP5	10	13	22	33	L12	I	Analog	
ADC_VINP6	11	14	23	34	L11	I	Analog	
ADC_VINP7	12	15	24	35	K12	I	Analog	
ADC_VINP8	13	16	25	36	K11	I	Analog	
ADC_VINP9	14	17	26	37	K10	I	Analog	
ADC_VINP10	-	-	28	39	J9	I	Analog	
ADC_VINP11	-	-	29	40	J10	I	Analog	
ADC_VREFP0	15	18	27	38	K9	I	Analog	

Table 6-3. Controller Area Network (CAN) Signals

PIN NAME	PIN NUMBER					IOSET	PIN TYPE	FUNCTION TYPE	DESCRIPTION
	48 VQFN	64 TQFP	100 TQFP	144 TQFP	144 TFBGA				
CAN0_RX	-	31	49	71	E12	1	I	ST	CAN0_FD Receive
CAN0_RX	29/34	36/**42**	54/**61**	76	B12	2			
CAN0_RX	-	-	46	68	F11	3			
CAN0_RX	15	**18**	**27**	60	C9	4			
CAN0_RX	-	-	-	132	G2	5			
CAN0_TX	-	30	48	70	E11	1	O	CMOS	CAN0_FD Transmit
CAN0_TX	28/33	35/**41**	53/**60**	75	C12	2			
CAN0_TX	-	-	45	65	C10	3			
CAN0_TX	14	**17**	**26**	59	E10	4			
CAN0_TX	-	-	-	131	G3	5			
CAN1_RX	-	-	-	38	K9	1	I	ST	CAN1_FD Receive
CAN1_RX	-	42	61	86	D7	2			
CAN1_RX	-	-	-	62	C8	3			
CAN1_RX	-	-	-	134	J1	4			
CAN1_TX	-	17	26	37	K10	1	O	CMOS	CAN1_FD Transmit
CAN1_TX	-	41	60	85	D8	2			
CAN1_TX	-	-	-	61	D10	3			
CAN1_TX	-	-	-	133	H2	4			

Table 6-4. Configurable Custom Logic (CCL) Signals

PIN NAME	PIN NUMBER					IOSET	PIN TYPE	FUNCTION TYPE	DESCRIPTION
	48 VQFN	64 TQFP	100 TQFP	144 TQFP	144 TFBGA				
LUT0_IN0	17	**20**/29	**31**/47	69	F12	1	I	Digital Input	Lookup Table 0, Input 0
	-	-	-	60	C9	2			
LUT0_IN1	18	**21**/30	**32**/48	70	E11	1	I	Digital Input	Lookup Table 0, Input 1
	-	-	-	61	D10	2			
LUT0_IN2	42	31/**53**	49/**88**	71	E12	1	I	Digital Input	Lookup Table 0, Input 2
	-	-	-	62	C8	2			
LUT1_IN0	36	45	68	101	B5	1	I	Digital Input	Lookup Table 1, Input 0
	-	-	-	87	A11	2			
LUT1_IN1	37	46	71	104	A4	1	I	Digital Input	Lookup Table 1, Input 1
	-	-	-	88	B10	2			
LUT1_IN2	33	**41**/47	**60**/72	105	A3	1	I	Digital Input	Lookup Table 1, Input 2
	-	-	-	92	A10	2			
LUT2_IN0	14	**17**/61	**26**/97	141	H3	1	I	Digital Input	Lookup Table 2, Input 0
	-	-	-	130	F2	2			
LUT2_IN1	15	**18**/62	**27**/98	142	J4	1	I	Digital Input	Lookup Table 2, Input 1
	-	-	-	131	G3	2			
LUT2_IN2	20	**23**/63	**36**/99	143	J5	1	I	Digital Input	Lookup Table 2, Input 2
	-	-	-	132	G2	2			
LUT3_IN0	-	8	13	23	K7	1	I	Digital Input	Lookup Table 3, Input 0
	-	-	8	12	L5	2			
LUT3_IN1	-	9	14	24	M8	1	I	Digital Input	Lookup Table 3, Input 1
	-	-	9	13	M5	2			

Table 6-4. Configurable Custom Logic (CCL) Signals (continued)

PIN NAME	PIN NUMBER					IOSET	PIN TYPE	FUNCTION TYPE	DESCRIPTION
	48 VQFN	64 TQFP	100 TQFP	144 TQFP	144 TFBGA				
LUT3_IN2	8	10	15	25	M9	1	I	Digital Input	Lookup Table 3, Input 2
			10	14	K6	2			
LUT0_OUT	1/22	**1/25**/32	**1/39**/50	72	D12	1	O	Digital Output	Lookup Table 0 Output
	-	-	-	63	G11	2			
LUT1_OUT	-	48	73	106	A2	1	O	Digital Output	Lookup Table 1 Output
	-	-	-	93	A9	2			
LUT2_OUT	34	**42**/64	**61**/100	144	J3	1	O	Digital Output	Lookup Table 2 Output
	-	-	-	133	H2	2			
LUT3_OUT	9/21	11/**24**	16/**37**	26	M10	1	O	Digital Output	Lookup Table 3 Output
	-	-	11	15	J6	2			

Table 6-5. DEBUG – SW-2 wire, JTAG, Trace, and Programming/Debugging Signals

PIN NAME	PIN NUMBER					PIN TYPE	FUNCTION TYPE	DESCRIPTION
	48 VQFN	64 TQFP	100 TQFP	144 TQFP	144 TFBGA			
TDI	1	1	1	1	L2	I	ST	Debug/Scan JTAG Data Input
TDO	2	2	2	2	L3	O	CMOS	Debug/Scan JTAG Data Out
TMS	3	3	3	3	K3	I	ST	Debug/Scan JTAG Mode Select
TCK	4	4	4	4	K4	I	ST	Debug/Scan JTAG Clock
SWO	2	2	2	2	L3	O	CMOS	Debug Serial Wire Output Data
SWDIO	3	3	3	3	K3	I/O	ST/CMOS	Debug Serial Wire Data In/Out
SWCLK	4	4	4	4	K4	I	ST	Debug Serial Wire Input Clock
TRACECLK	-	-	11	15	J6	I	ST	Debug Trace Clock
TRACE_DATA0	-	-	12	16	M6	I/O	ST/CMOS	Debug Instruction Trace Data 0
TRACE_DATA1	-	-	10	14	K6			Debug Instruction Trace Data 1
TRACE_DATA2	-	-	9	13	M5			Debug Instruction Trace Data 2
TRACE_DATA3	-	-	8	12	L5			Debug Instruction Trace Data 3
RESET	5	5	5	5	L4	I	ST	MCU Reset input (Active Low)

Table 6-6. External Bus Interface (EBI) Signals

PIN NAME	PIN NUMBER			PIN TYPE	FUNCTION TYPE	DESCRIPTION
	100 TQFP	144 TQFP	144 TFBGA			
EBI_A2	74	107	B3	O	CMOS	EBI Address A[23:1} (External BUS interface)
EBI_A3	67	100	A5			
EBI_A4	66	99	A6			
EBI_A5	65	98	B6			
EBI_A6	64	95	A8			
EBI_A7	59	84	B9			
EBI_A8	58	82	C4			
EBI_A9	55	77	B11			
EBI_A10	54	76	B12			
EBI_A11	53	75	C12			
EBI_A12	52	74	C11			
EBI_A13	51	73	D11			
EBI_A14	50	72	D12			
EBI_A15	49	71	E12			
EBI_A16	97	93/141	A9/H3			
EBI_A17	94	92/138	A10/L1			
EBI_A18	93	63/137	G11/K2			
EBI_A19	88	62/121	C8/D3			
EBI_A20	-	61	D10			
EBI_A21	-	60	C9			
EBI_A22	-	59	E10			
EBI_A23	-	96	A7			
EBI_D0	8	12	L5	I/O	CMOS	EBI Data Bus [D15:0] (External BUS interface)
EBI_D1	9	13	M5			
EBI_D2	10	14	K6			
EBI_D3	11	15	J6			
EBI_D4	12	16	M6			
EBI_D5	13	23	K7			
EBI_D6	14	24	M8			
EBI_D7	15	25	M9			
EBI_D8	16	26	M10			
EBI_D9	17	27	K8			
EBI_D10	28	39	J9			
EBI_D11	29	40	J10			
EBI_D12	37	51	G8			
EBI_D13	38	52	G10			
EBI_D14	39	53	H9			
EBI_D15	40	54	G9			
EBI_NBS0_A0	48	70	E11	O	CMOS	EBI Address A0 or Byte Select 0
EBI_NBS1_NWR1	47	69	F12	O	CMOS	EBI Byte Select1 or Byte Write1 (Active Low)
EBI_NBS2_NWR2_A1	75	108	B4	O	CMOS	EBI Byte Select2 or Byte Write2 (Active Low) or Address A1

Table 6-6. External Bus Interface (EBI) Signals (continued)

PIN NAME	PIN NUMBER			PIN TYPE	FUNCTION TYPE	DESCRIPTION
	100 TQFP	144 TQFP	144 TFBGA			
EBI_NCS0	91	88/135	B10/J2	O	CMOS	EBI Chip Select 0 (Active Low)
EBI_NCS1	98	83/142	C5/J4			EBI Chip Select 1 (Active Low)
EBI_NCS2	-	97	B7			EBI Chip Select 2 (Active Low)
EBI_NCS3	-	126	F3			EBI Chip Select 3 (Active Low)
EBI_NRD	41	55	F10	O	CMOS	EBI Read (Active Low)
EBI_NWAIT	23/100	34/81/144	L11/C7/J3	O	CMOS	EBI WAIT output (Active Low)
EBI_NWE_NWR0	92	87/136	A11/K1	O	CMOS	EBI Write Enb or Write0 (Active Low)

Table 6-7. External Interrupt Controller (EIC) Signals

PIN NAME	PIN NUMBER					PIN TYPE	FUNCTION TYPE	DESCRIPTION
	48 VQFN	64 TQFP	100 TQFP	144 TQFP	144 TFBGA			
EIC_EXTINT0	14/26/40	17/33/51	11/26/45/51/65/86	15/37/65/73/98/119/127	B6/C10/D11/E1/E2/J6/K10	I	ST	External Interrupt 0 input
EIC_EXTINT1	2/15/27/41	2/**18**/34/52	2/12/**27**/46/52/66/87	2/16/68/74/99/120/130	A6/C11/F1/F2/F11/L3/M6	I	ST	External Interrupt 1 input
EIC_EXTINT2	3/17/28	3/20/35	3/18/31/53/67	3/28/41/45/75/100/131	A5/C12/G3/H8/J11/K3/M11	I	ST	External Interrupt 2 input
EIC_EXTINT3	4/18/29	4/21/36	4/**21**/32/54/74	4/42/46/76/107/132	B3/B12/G2/H10/H11/K4	I	ST	External Interrupt 3 input
EIC_EXTINT4	20	8/23/37	13/36/55/75	8/23/43/50/77/108/133	B4/B11/H2/H12/J12/K5/K7	I	ST	External Interrupt 4 input
EIC_EXTINT5	21/32	9/24/40/56	14/37/59/92	9/24/51/59/80/84/134/136	B9/C6/E10/G8/J1/K1/M2/M8	I	ST	External Interrupt 5 input
EIC_EXTINT6	8/22/33/45	10/25/41/57	15/39/60/93	10/25/53/60/85/81/137	C7/C9/D8/H9/K2/M3/M9	I	ST	External Interrupt 6 input
EIC_EXTINT7	9/34/46	11/29/42/58	16/47/61/94	11/26/61/69/83/86/138	C5/D7/D10/F12/L1/M4/M10	I	ST	External Interrupt 7 input
EIC_EXTINT8	-	12/30/61	17/48/97	17/27/62/70/87/141	A11/C8/E11/H3/K8/L6	I	ST	External Interrupt 8 input
EIC_EXTINT9	10	13/31/44/62	22/49/63/98	20/33/63/71/88/94/142	B8/B10/E12/G11/J4/L12/M7	I	ST	External Interrupt 9 input
EIC_EXTINT10	11/36	**14**/32/45/63	**23**/50/68/99	21/64/72/92/101/143	A10/B5/D12/G12/J5/L7	I	ST	External Interrupt 10 input
EIC_EXTINT11	12/37	15/46/64	24/28/71/100	22/35/39/93/104/144	A4/A9/J3/J9/K12/L8	I	ST	External Interrupt 11 input
EIC_EXTINT12	13	16/47	25/29/72/91	29/36/40/96/105/135	A3/A7/J2/J10/K11/L9	I	ST	External Interrupt 12 input
EIC_EXTINT13	-	48	8/38/73	12/52/97/106/124	A2/B7/E3/G10/L5	I	ST	External Interrupt 13 input

Table 6-7. External Interrupt Controller (EIC) Signals (continued)

PIN NAME	PIN NUMBER					PIN TYPE	FUNCTION TYPE	DESCRIPTION
	48 VQFN	64 TQFP	100 TQFP	144 TQFP	144 TFBGA			
EIC_EXTINT14	1	1	1/40/58/9	1/13/54/ 82/125	C3/C4/G9/ L2/M5	I	ST	External Interrupt 14 input
EIC_EXTINT15	-	-	10/41/64	14/55/ 95/126	A8/F3/ F10/K6	I	ST	External Interrupt 15 input
EIC_NMI	42	53	88	121	D3	I	ST	Non-Maskable Interrupt input

Table 6-8. Ethernet MAC Signals

PIN NAME	PIN NUMBER					PIN TYPE	FUNCTION TYPE	DESCRIPTION
	48 VQFN	64 TQFP	100 TQFP	144 TQFP	144 TFBGA			
ETH_COL	-	-	67	100	A5	I	ST/CMOS	Receiver Collision Detect
ETH_CRS	-	-	66	99	A6	I	ST	Receiver Carrier Sense
ETH_MDC	12/29/46	15/32/36/58	24/50/54/94	35/72/76/138	B12/D12 /K12/L1	O	CMOS	Management data clock
ETH_MDIO	13/28/45	16/35/37/57	25/53/55/93	36/75/77/80/137	B11/C6/ C12/K2/ K11	I/O	ST/CMOS	Management data
ETH_REF_CLK	-	33	51	73	D11	I	ST	Reference Clock
ETH_RXD0	34	42	61	86	D7	I	ST	Ethernet Media Access Controller Receive data Rx[3:0]
ETH_RXD1	33	41	60	85	D8			
ETH_RXD2	32	40	59	84	B9			
ETH_RXD3	-	-	58	82	C4			
ETH_RXDV	36	45	68	101	B5	I	ST	Receive Data Valid
ETH_RXER	-	44	63	94	B8	I	ST	Receive Data Error
ETH_RX_CLK	-	-	64	95	A8	I	ST	Receive Clock
ETH_TSUCOMP	42	53	88	121/127	D3/E2	O	CMOS	Time Stamp Unit Compensation
ETH_TXD0	-	47	72	105	A3	O	CMOS	Ethernet Media Access Controller Transmit data Tx[3:0]
ETH_TXD1	-	48	73	106	A2			
ETH_TXD2	-	-	74	107	B3			
ETH_TXD3	-	-	75	108	B4			
ETH_TXEN	37	46	71	104	A4	O	CMOS	Transmit Enable
ETH_TXER	-	-	65	98	B6	O	CMOS	Transmit Error
ETH_TX_CLK	26	33	51	73	D11	I	ST	Transmit Clock

Table 6-9. Generic Clock Controller (GCLK) Signals

PIN NAME	PIN NUMBER					PIN TYPE	FUNCTION TYPE	DESCRIPTION
	48 VQFN	64 TQFP	100 TQFP	144 TQFP	144 TFBGA			
GCLK_IO0	4/10	4/13	4/22	4/33	K4/L12	I	ST	External GCLK[7:0] pin Input clock source
GCLK_IO1	2/11	2/14	2/23	2/34	L3/L11			
GCLK_IO2	26	33	51/91	8/73/135	D11/J2/K5			
GCLK_IO3	27	34/56	52/92	9/74/136	C11/K1/M2			
GCLK_IO4	28/45	35/57	53/93	10/75/137	C12/K2/M3			
GCLK_IO5	29/46	36/58	54/94	11/76/138	B12/L1/M4			
GCLK_IO6	-	37	12/55	16/77	B11/M6			
GCLK_IO7	32	40	59	17/84	B9/L6			

Table 6-10. Inter-IC Sound Controller (I^2S) Signals

PIN NAME	PIN NUMBER					IOSET	PIN TYPE	FUNCTION TYPE	DESCRIPTION
	48 VQFN	64 TQFP	100 TQFP	144 TQFP	144 TFBGA				
SPI_IXS_FSYNC	45	57	93	137	K2	1	I/O	ST/CMOS	Frame Synchronization (LRCK)
	10	13	22	33	L12	2			
	-	-	-	124	E3	3			
SPI_IXS_SCK	-	63	99	143	J5	1	I/O	ST/CMOS	Serial Bit Clock
	12	15	24	35	K12	2			
	-	-	-	125	C3	3			
SPI_IXS_SDI	46	58	94	138	L1	1	I	ST	Serial Data Input for Receive Serializer
	13	16	25	36	K11	2			
	-	-	-	127	E2	3			
SPI_IXS_SDO	-	61	97	141	H3	1	O	CMOS	Serial Data Output for Transmit Serializer
	14	17	26	37	K10	2			
	-	-	-	130	F2	3			

Table 6-11. 32 kHz Oscillators Controller (OSC32KCTRL) Signals

PIN NAME	PIN NUMBER					PIN TYPE	FUNCTION TYPE	DESCRIPTION
	48 VQFN	64 TQFP	100 TQFP	144 TQFP	144 TFBGA			
OSC32KCTRL_XIN32	17	20	31	45	J11	I	Analog	32.768 kHz oscillator input
OSC32KCTRL_XOUT32	18	21	32	46	H11	O	Analog	32.768 kHz oscillator output

Table 6-12. Oscillator Controller (OSCCTRL) Signals

PIN NAME	PIN NUMBER					PIN TYPE	FUNCTION TYPE	DESCRIPTION
	48 VQFN	64 TQFP	100 TQFP	144 TQFP	144 TFBGA			
OSCCTRL_XIN	40	51	86	119	E1	I	Analog	Primary oscillator input
OSCCTRL_XOUT	41	52	87	120	F1	O	Analog	Primary oscillator output

Table 6-13. Parallel Capture Controller (PCC) Signals

PIN NAME	PIN NUMBER					IOSET	PIN TYPE	FUNCTION TYPE	DESCRIPTION
	48 VQFN	64 TQFP	100 TQFP	144 TQFP	144 TFBGA				
PCC_CLK	32	40	59	84	B9	1	I	ST	Parallel Capture Mode Clock
	-	-	-	8	K5	2			

Table 6-13. Parallel Capture Controller (PCC) Signals (continued)

PIN NAME	PIN NUMBER					IOSET	PIN TYPE	FUNCTION TYPE	DESCRIPTION
	48 VQFN	64 TQFP	100 TQFP	144 TQFP	144 TFBGA				
PCC_DATA0	27	34	52	74	C11	1	I/O	ST/CMOS	Parallel Capture Mode Data[13:0]
	-	-	-	9	M2	2			
PCC_DATA1	28	35	53	75	C12	1			
	-	-	-	10	M3	2			
PCC_DATA2	29	36	54	76	B12	1			
	-	-	-	11	M4	2			
PCC_DATA3	33	29/**41**	47/**60**	69	F12	1			
	-	-	-	17	L6	2			
PCC_DATA4	34	30/**42**	48/**61**	70	E11	1			
	-	-	-	20	M7	2			
PCC_DATA5	36	31/**45**	49/**68**	71	E12	1			
	-	-	-	21	L7	2			
PCC_DATA6	37	32/**46**	50/**71**	72	D12	1			
	-	-	-	22	L8	2			
PCC_DATA7	26	33	51	73	D11	1			
	-	-	-	29	L9	2			
PCC_DATA8	-	-	64	95	A8	1			
	-	-	-	41	H8	2			
PCC_DATA9	-	-	65	98	B6	1			
	-	-	-	42	H10	2			
PCC_DATA10	-	-	66	99	A6	1			
	-	-	-	43	J12	2			
PCC_DATA11	-	-	67	100	A5	1			
	-	-	-	64	G12	2			
PCC_DATA12	-	-	-	107	B3	1			
	-	-	74	80	C6	2			
PCC_DATA13	-	-	-	108	B4	1			
	-	-	75	81	C7	2			
PCC_DEN1	11	14/37	23/55	34/77	B11/L11	1	I	ST	Parallel Capture Mode Enable 1
	-	-	-	132	G2	2			
PCC_DEN2	42	53	58/88	82/121	C4/D3	1	I	ST	Parallel Capture Mode Enable 2
	-	-	-	133	H2	2			

Table 6-14. Position Decoder (PDEC) Signals

PIN NAME	PIN NUMBER					IOSET	PIN TYPE	FUNCTION TYPE	DESCRIPTION
	48 VQFN	64 TQFP	100 TQFP	144 TQFP	144 TFBGA				
PDEC_QDI0	20	23	36	50	H12	1	I	ST	PDEC Inputs[2:0]
	-	29	47	69	F12	2			
	20	**23**	**36**	8	K5	3			
PDEC_QDI1	21	24	37	51	G8	1			
	-	30	48	70	E11	2			
	21	**24**	**37**	9	M2	3			
PDEC_QDI2	-	31	49	71	E12	2			
	-	-	38	52	G10	1			
	22	**25**	**39**	10	M3	3			

Table 6-15. PORTA through PORTD Signals

PIN NAME	PIN NUMBER					PIN TYPE	FUNCTION TYPE	DESCRIPTION
	48 VQFN	64 TQFP	100 TQFP	144 TQFP	144 TFBGA			
PA00	1	1	1	1	L2	I/O	ST/CMOS	I/O Port A pins
PA01	2	2	2	2	L3			
PA02	3	3	3	3	K3			
PA03	4	4	4	4	K4			
PA04	-	8	13	23	K7			
PA05	-	9	14	24	M8			
PA06	8	10	15	25	M9			
PA07	9	11	16	26	M10			
PA08	-	12	17	27	K8			
PA09	10	13	22	33	L12			
PA10	11	14	23	34	L11			
PA11	12	15	24	35	K12			
PA12	13	16	25	36	K11			
PA13	-	-	8	12	L5			
PA14	-	-	9	13	M5			
PA15	-	-	10	14	K6			
PA16	-	-	11	15	J6			
PA17	-	-	12	16	M6			
PA18	-	-	18	28	M11			
PA19	-	-	21	32	L10			
PA20	-	-	-	8	K5			
PA21	-	-	-	9	M2			
PA22	-	-	-	10	M3			
PA23	-	-	-	11	M4			
PA24	-	-	-	17	L6			
PA25	-	-	-	20	M7			
PA26	-	-	-	21	L7			
PA27	-	-	-	22	L8			
PA28	-	-	-	29	L9			

Table 6-15. PORTA through PORTD Signals (continued)

PIN NAME	PIN NUMBER					PIN TYPE	FUNCTION TYPE	DESCRIPTION
	48 VQFN	64 TQFP	100 TQFP	144 TQFP	144 TFBGA			
PB00	14	17	26	37	K10	I/O	ST/CMOS	I/O Port B pins
PB01	15	18	27	38	K9			
PB02	17	20	31	45	J11			
PB03	18	21	32	46	H11			
PB04	20	23	36	50	H12			
PB05	21	24	37	51	G8			
PB06	22	25	39	53	H9			
PB07	-	29	47	69	F12			
PB08	-	30	48	70	E11			
PB09	-	31	49	71	E12			
PB10	-	32	50	72	D12			
PB11	-	-	28	39	J9			
PB12	-	-	29	40	J10			
PB13	-	-	38	52	G10			
PB14	-	-	40	54	G9			
PB15	-	-	41	55	F10			
PB16	-	-	45	65	C10			
PB17	-	-	46	68	F11			
PB18	-	-	-	41	H8			
PB19	-	-	-	42	H10			
PB20	-	-	-	43	J12			
PB21	-	-	-	59	E10			
PB22	-	-	-	60	C9			
PB23	-	-	-	61	D10			
PB24	-	-	-	62	C8			
PB25	-	-	-	63	G11			
PB26	-	-	-	64	G12			

Table 6-15. PORTA through PORTD Signals (continued)

PIN NAME	PIN NUMBER					PIN TYPE	FUNCTION TYPE	DESCRIPTION
	48 VQFN	64 TQFP	100 TQFP	144 TQFP	144 TFBGA			
PC00	26	33	51	73	D11	I/O	ST/CMOS	I/O Port C pins
PC01	27	34	52	74	C11			
PC02	28	35	53	75	C12			
PC03	29	36	54	76	B12			
PC04	-	37	55	77	B11			
PC05	32	40	59	84	B9			
PC06	33	41	60	85	D8			
PC07	34	42	61	86	D7			
PC09	-	44	63	94	B8			
PC10	36	45	68	101	B5			
PC11	37	46	71	104	A4			
PC12	-	47	72	105	A3			
PC13	-	48	73	106	A2			
PC14	-	-	58	82	C4			
PC15	-	-	64	95	A8			
PC16	-	-	65	98	B6			
PC17	-	-	66	99	A6			
PC18	-	-	67	100	A5			
PC19	-	-	74	107	B3			
PC20	-	-	75	108	B4			
PC21	-	-	-	80	C6			
PC22	-	-	-	81	C7			
PC23	-	-	-	83	C5			
PC24	-	-	-	87	A11			
PC25	-	-	-	88	B10			
PC26	-	-	-	92	A10			
PC27	-	-	-	93	A9			
PC28	-	-	-	96	A7			
PC29	-	-	-	97	B7			

Table 6-15. PORTA through PORTD Signals (continued)

PIN NAME	PIN NUMBER					PIN TYPE	FUNCTION TYPE	DESCRIPTION
	48 VQFN	64 TQFP	100 TQFP	144 TQFP	144 TFBGA			
PD00	40	51	86	119	E1	I/O	ST/CMOS	I/O Port D pins
PD01	41	52	87	120	F1			
PD02	42	53	88	121	D3			
PD05		56	92	136	K1			
PD06	45	57	93	137	K2			
PD07	46	58	94	138	L1			
PD08	-	61	97	141	H3			
PD09	-	62	98	142	J4			
PD10	-	63	99	143	J5			
PD11	-	64	100	144	J3			
PD12	-	-	91	135	J2			
PD13	-	-	-	124	E3			
PD14	-	-	-	125	C3			
PD15	-	-	-	126	F3			
PD16	-	-	-	127	E2			
PD17	-	-	-	130	F2			
PD18	-	-	-	131	G3			
PD19	-	-	-	132	G2			
PD20	-	-	-	133	H2			
PD21	-	-	-	134	J1			

Table 6-16. Power (PWR) and Ground Signals

PIN NAME	PIN NUMBER					PIN TYPE	FUNCTION TYPE	DESCRIPTION
	48 VQFN	64 TQFP	100 TQFP	144 TQFP	144 TFBGA			
AVDD	19	22	33	47	H7	PWR	Power	Positive supply for analog modules.
AVSS	16	19	30	44	F6	PWR	Power	Ground for analog modules.
VDDIO	7/23/30/39/48	7/26/38/50/60	7/20/34/42/ 56/69/85/96	7/19/31//48/ 56/66/78/ 90/102/118/ 129/140	D5/D6/D9/ E9/F4/F8/ G4/G7/H4/ H6/J7/J8	PWR	Power	Positive supply for I/O logic
VDDREG	24/35	28/43	44/62/83	58/89/116	E4/E7/F9	PWR	Power	Positive supply for CORE logic
VSS	6/25/31/38/47	6/27/39/49/59	6/19/35/43/57/ 70/80/84/95	6/18/30/44/ 49/57/67/79/ 91/113/117/ 128/139/204	A1/A12/D4 / E5/E6/E8/F 5/ F7/G5/G6/ H5/M1/M1 2	PWR	Power	Ground for CORE, peripherals and I/O

Table 6-17. Peripheral Touch Controller (PTC) Signals

PIN NAME	PIN NUMBER					PIN TYPE	FUNCTION TYPE	DESCRIPTION
	48 VQFN	64 TQFP	100 TQFP	144 TQFP	144 TFBGA			
PTC_DRV0	10	13	22	33	L12	I/O	ST/CMOS	Digital Output (X-line) & Analog I/O (Y-line)
PTC_DRV1	11	14	23	34	L11			
PTC_DRV2	12	15	24	35	K12			
PTC_DRV3	13	16	25	36	K11			
PTC_DRV4	14	17	26	37	K10			
PTC_DRV5	15	18	27	38	K9			
PTC_DRV6	-	29	47	69	F12			
PTC_DRV7	-	30	48	70	E11			
PTC_DRV8	-	31	49	71	E12			
PTC_DRV9	-	32	50	72	D12			
PTC_DRV10	26	33	51	73	D11			
PTC_DRV11	27	34	52	74	C11			
PTC_DRV12	28	35	53	75	C12			
PTC_DRV13	29	36	54	76	B12			
PTC_DRV14	-	37	55	77	B11			
PTC_DRV15	32	40	59	84	B9			
PTC_DRV16	33	41	60	85	D8			
PTC_DRV17	34	42	61	86	D7			
PTC_DRV18	-	48	73	106	A2			
PTC_DRV19	-	44	63	94	B8			
PTC_DRV20	36	45	68	101	B5			
PTC_DRV21	37	46	71	104	A4			
PTC_DRV22	-	47	72	105	A3			
PTC_DRV23	-	-	75	108	B4			
PTC_DRV24	-	-	45	65	C10			
PTC_DRV25	-	-	46	68	F11			
PTC_DRV26	-	-	58	82	C4			
PTC_DRV27	-	-	64	95	A8			
PTC_DRV28	-	-	65	98	B6			
PTC_DRV29	-	-	66	99	A6			
PTC_DRV30	-	-	67	100	A5			
PTC_DRV31	-	-	74	107	B3			
PTC_ECI0	-	-	18	28	M11	I	Analog	External Capacitor Analog Input 0
PTC_ECI1	-	-	21	32	L10	I	Analog	External Capacitor Analog Input 1

Table 6-18. Real-Time Counter (RTC) Signals

PIN NAME	PIN NUMBER					PIN TYPE	FUNCTION TYPE	DESCRIPTION
	48 VQFN	64 TQFP	100 TQFP	144 TQFP	144 TFBGA			
RTC_IN0	22	25	39	53	H9	I	ST	RTC security tamper input loops
RTC_IN1	15	18	27	38	K9			
RTC_IN2	-	-	40	54	G9			
RTC_IN3	-	-	41	55	F10			
RTC_IN4	-	-	38	52	G10			
RTC_IN5	-	-	-	42	H10			
RTC_IN6	-	-	-	60	C9			
RTC_IN7	-	-	-	62	C8			
RTC_OUT0	21	24	37	51	G8	O	CMOS	RTC security tamper output loops
RTC_OUT1	14	17	26	37	K10			
RTC_OUT2	-	-	28	39	J9			
RTC_OUT3	-	-	29	40	J10			
RTC_OUT4	-	-	45	65	C10			
RTC_OUT5	-	-	-	41	H8			
RTC_OUT6	-	-	-	59	E10			
RTC_OUT7	-	-	-	61	D10			

Table 6-19. Host Controller (SDMMC0 and SDMMC1) Signals

PIN NAME	PIN NUMBER					PIN TYPE	FUNCTION TYPE	DESCRIPTION
	48 VQFN	64 TQFP	100 TQFP	144 TQFP	144 TFBGA			
SDMMC0_SDCD	11/13/36	14/16/37/45	23/25/55/68/91	34/36/135	L11/K11/J2	I	ST	SD Host Controller 0 Card Detect (10K-100K Pull-up required)
SDMMC0_SDCK	32	40/63	59/99	143	J5	O	CMOS	SD Host Controller 0 Clock
SDMMC0_SDCMD	33/45	29/41/57	47/60/93	137	K2	I/O	ST/CMOS	SD Host Controller 0 Command (10K-100K Pull-up required)
SDMMC0_SDDAT0	34/46	32/42/58	50/61/94	138	L1	I/O	ST/CMOS	SD Host Controller 0 DATA0 (10K-100K Pullup required)
SDMMC0_SDDAT1	26	33/64	51/100	144	J3			SD Host Controller 0 DATA1 (10K-100K Pullup required)
SDMMC0_SDDAT2	27	34/61	52/97	141	H3			SD Host Controller 0 DATA2 (10K-100K Pullup required)
SDMMC0_SDDAT3	28	35/62	53/98	142	J4			SD Host Controller 0 DATA3 (10K-100K Pullup required)
SDMMC0_SDWP	14/29	17/36/56	26/54/92	37/136	K10/K1	I	ST	SD Host Controller 0 Write Protect (10K-100K Pull-up required)
SDMMC1_SDCD	-	37	55	77	B11	I	ST	SD Host Controller 1 Card Detect (10K-100K Pull-up required)

Table 6-19. Host Controller (SDMMC0 and SDMMC1) Signals (continued)

PIN NAME	PIN NUMBER					PIN TYPE	FUNCTION TYPE	DESCRIPTION
	48 VQFN	64 TQFP	100 TQFP	144 TQFP	144 TFBGA			
SDMMC1_SDCK	-	40	59	84	B9	O	CMOS	SD Host Controller 1 Clock
SDMMC1_SDCMD	-	29	47	69	F12	I/O	ST/CMOS	SD Host Controller 1 Command (10K-100K Pull-up required)
SDMMC1_SDDAT0	-	32	50	72	D12	I/O	ST/CMOS	SD Host Controller 1 DATA0 (10K-100K Pullup required)
SDMMC1_SDDAT1	-	33	51	73	D11			SD Host Controller 1 DATA1 (10K-100K Pullup required)
SDMMC1_SDDAT2	-	34	52	74	C11			SD Host Controller 1 DATA2 (10K-100K Pullup required)
SDMMC1_SDDAT3	-	35	53	75	C12			SD Host Controller 1 DATA3 (10K-100K Pullup required)
SDMMC1_SDWP	-	36	54	76	B12	I	ST	SD Host Controller 1 Write Protect (10K-100K Pull-up required)

Table 6-20. Serial Communication Interface (SERCOM) Signals

PIN NAME	PIN NUMBER					IOSET	PIN TYPE	FUNCTION TYPE	DESCRIPTION
	48 TQFP	64 TQFP	100 TQFP	144 TQFP	144 TFBGA				
SERCOM0_PAD0	36	45	68	101	B5	1	I/O	ST/CMOS	UART0_RxD, TxD / SPI0_DI,DO / I2C0_SDA
	-	-	-	130	F2	2			
	45	57	93	137	K2	3			
SERCOM0_PAD1	37	46	71	104	A4	1			UART0_RxD,XCK / SPI0_DI, SCK / I2C0_SCL
	-	-	-	131	G3	2			
	46	58	94	138	L1	3			
SERCOM0_PAD2	-	47	72	105	A3	1			UART0_RxD, RTS, TE / SPI0_DI, $\overline{SS}$ / I2C0_SDA_OUT
	-	-	-	132	G2	2			
	1	56	92	136	K1	3			
SERCOM0_PAD3	-	48	73	106	A2	1			UART0_RxD, CTS / SPI0_DI,DO / I2C0_SCL_OUT
	-	-	-	133	H2	2			
	2	-	91	135	J2	3			
SERCOM1_PAD0	-	61	97	141	H3	1	I/O	ST/CMOS	UART1_RxD, TxD / SPI1_DI,DO / I2C1_SDA
	1	-	-	124	E3	2			
SERCOM1_PAD1	-	62	98	142	J4	1			UART1_RxD,XCK / SPI1_DI, SCK / I2C1_SCL
	2	-	-	125	C3	2			
SERCOM1_PAD2	-	63	99	143	J5	1			UART1_RxD, RTS,TE / SPI1_DI, $\overline{SS}$ / I2C1_SDA_OUT
	45	-	-	126	F3	2			
SERCOM1_PAD3	-	64	100	144	J3	1			UART1_RxD, CTS / SPI1_DI,DO / I2C1_SCL_OUT
	46	-	-	127	E2	2			

Table 6-20. Serial Communication Interface (SERCOM) Signals (continued)

PIN NAME	PIN NUMBER					IOSET	PIN TYPE	FUNCTION TYPE	DESCRIPTION
	48 TQFP	64 TQFP	100 TQFP	144 TQFP	144 TFBGA				
SERCOM2_PAD0	37	46	71	104	A4	1	I/O	ST/CMOS	UART2_RxD, TxD / SPI2_DI,DO / I2C2_SDA
	-	-	64	95	A8	2			
	37	46	71	80	C6	3			
	33	41	60	85	D8	4			
SERCOM2_PAD1	36	45	68	101	B5	1			UART2_RxD,XCK / SPI2_DI, SCK / I2C2_SCL
	-	-	65	98	B6	2			
	36	45	68	81	C7	3			
	34	42	61	86	D7	4			
SERCOM2_PAD2	-	47	72	105	A3	1			UART2_RxD, RTS, TE / SPI2_DI, $\overline{SS}$ / I2C2_SDA_OUT
	-	-	66	99	A6	2			
	32	40	58	82	C4	3			
	-	44	63	94	B8	4			
SERCOM2_PAD3	-	48	73	106	A2	1			UART2_RxD, CTS / SPI2_DI,DO / I2C2_SCL_OUT
	-	-	67	100	A5	2			
	28	-	-	83	C5	3			
	-	-	-	92	A10	4			
SERCOM3_PAD0	28	35	53	75	C12	1	I/O	ST/CMOS	UART3_RxD, TxD / SPI3_DI,DO / I2C3_SDA
	-	-	45	65	C10	2			
	-	-	-	59	E10	3			
	34	42	61	86	D7	4			
SERCOM3_PAD1	29	36	54	76	B12	1			UART3_RxD,XCK / SPI3_DI, SCK / I2C3_SCL
	-	-	46	68	F11	2			
	-	-	-	60	C9	3			
	33	41	60	85	D8	4			
SERCOM3_PAD2	-	37	55	77	B11	1			UART3_RxD, RTS,TE / SPI3_DI, $\overline{SS}$ / I2C3_SDA_OUT
	-	29	47	69	F12	2			
	-	-	-	61	D10	3			
	27	34/44	52/63	94	B8	4			
SERCOM3_PAD3	32	40	59	84	B9	1			UART3_RxD, CTS / SPI3_DI,DO / I2C3_SCL_OUT
	-	30	48	70	E11	2			
	-	-	-	62	C8	3			
	26	33	51/64	95	A8	4			

Table 6-20. Serial Communication Interface (SERCOM) Signals (continued)

PIN NAME	PIN NUMBER					IOSET	PIN TYPE	FUNCTION TYPE	DESCRIPTION
	48 TQFP	64 TQFP	100 TQFP	144 TQFP	144 TFBGA				
SERCOM4_PAD0	-	31	49	71	E12	1	I/O	ST/CMOS	UART4_RxD, TxD / SPI4_DI,DO / I2C4_SDA
	20	-	-	50	H12	2			
	-	-	29	40	J10	3			
	-	32	50	72	D12	4			
SERCOM4_PAD1	-	32	50	72	D12	1			UART4_RxD,XCK / SPI4_DI, SCK / I2C4_SCL
	21	-	-	51	G8	2			
	-	-	-	41	H8	3			
	-	31	49	71	E12	4			
SERCOM4_PAD2	26	33	51	73	D11	1			UART4_RxD, RTS,TE / SPI4_DI, $\overline{SS}$ / I2C4_SDA_OUT
	11	-	38	52	G10	2			
	-	-	-	42	H10	3			
	-	29	47	69	F12	4			
SERCOM4_PAD3	27	34	52	74	C11	1			UART4_RxD, CTS / SPI4_DI,DO / I2C4_SCL_OUT
	22	-	-	53	H9	2			
	-	-	-	43	J12	3			
	-	30	48	70	E11	4			
SERCOM5_PAD0	12	15	24	35	K12	1	I/O	ST/CMOS	UART5_RxD, TxD / SPI5_DI,DO / I2C5_SDA
	-	-	8	12	L5	2			
	-	-	-	17	L6	3			
	-	9	14	24	M8	4			
SERCOM5_PAD1	13	16	25	36	K11	1			UART5_RxD,XCK / SPI5_DI, SCK / I2C5_SCL
	-	-	9	13	M5	2			
	-	-	-	20	M7	3			
	8	10	15	25	M9	4			
SERCOM5_PAD2	14	17	26	37	K10	1			UART5_RxD, RTS,TE / SPI5_DI, $\overline{SS}$ / I2C5_SDA_OUT
	-	-	10	14	K6	2			
	-	-	-	21	L7	3			
	9	11	16	26	M10	4			
SERCOM5_PAD3	15	18	27	38	K9	1			UART5_RxD, CTS / SPI5_DI,DO / I2C5_SCL_OUT
	-	-	11	15	J6	2			
	-	-	-	22	L8	3			
	-	12	17	27	K8	4			

Table 6-20. Serial Communication Interface (SERCOM) Signals (continued)

PIN NAME	PIN NUMBER					IOSET	PIN TYPE	FUNCTION TYPE	DESCRIPTION
	48 TQFP	64 TQFP	100 TQFP	144 TQFP	144 TFBGA				
SERCOM6_PAD0	-	11	16	26	M10	1	I/O	ST/CMOS	UART6_RxD, TxD / SPI6_DI,DO / I2C6_SDA
	-	1	1	1	L2	2			
	-	-	18	28	M11	3			
SERCOM6_PAD1	-	12	17	27	K8	1			UART6_RxD,XCK / SPI6_DI, SCK / I2C6_SCL
	-	2	2	2	L3	2			
	-	-	-	29	L9	3			
SERCOM6_PAD2	-	13	22	33	L12	1			UART6_RxD, RTS,TE / SPI6_DI, $\overline{SS}$ / I2C6_SDA_OUT
	-	3	3	3	K3	2			
	-	-	21	32	L10	3			
SERCOM6_PAD3	-	14	23	34	L11	1			UART6_RxD, CTS / SPI6_DI,DO / I2C6_SCL_OUT
	-	4	4	4	K4	2			
	-	13	22	33	L12	3			
SERCOM7_PAD0	-	2	2	2	L3	1	I/O	ST/CMOS	UART7_RxD, TxD / SPI7_DI,DO / I2C7_SDA
	-	-	-	9	M2	2			
	-	16	25	36	K11	3			
	-	-	-	41	H8	4			
SERCOM7_PAD1	-	8	13	23	K7	1			UART7_RxD,XCK / SPI7_DI, SCK / I2C7_SCL
	-	-	-	8	K5	2			
	-	15	24	35	K12	3			
	-	-	29	40	J10	4			
SERCOM7_PAD2	-	9	14	24	M8	1			UART7_RxD, RTS,TE / SPI7_DI, $\overline{SS}$ / I2C7_SDA_OUT
	-	-	-	10	M3	2			
	-	17	26	37	K10	3			
	-	-	-	42	H10	4			
SERCOM7_PAD3	-	10	15	25	M9	1			UART7_RxD, CTS / SPI7_DI,DO / I2C7_SCL_OUT
	-	-	-	11	M4	2			
	-	18	27	38	K9	3			
	-	-	-	43	J12	4			

Table 6-21. Quad Serial Peripheral Interface (QSPI/SQI) Signals

PIN NAME	PIN NUMBER					PIN TYPE	FUNCTION TYPE	DESCRIPTION
	48 VQFN	64 TQFP	100 TQFP	144 TQFP	144 TFBGA			
SQI_CLK	33	**41**/63	**60**/99	143	J5	O	CMOS	Quad SPI Clock
SQI_CS0	27/45	**34**/57	**52**/93	137	K2	O	CMOS	Quad SPI Chip Select 0
SQI_CS1	26	**33**/56	**51**/92	136	K1			Quad SPI Chip Select 1
SQI_CS2	11	14	23	34	L11			Quad SPI Chip Select 2
SQI_CS3	-	-	91	135	J2			Quad SPI Chip Select 3
SQI_DATA0	28/34/46	**35/42**/58	**52/61**/94	138	L1	I/O	ST/CMOS	Quad SPI Data 0 I/O
SQI_DATA1	-	64	100	144	J3			Quad SPI Data 1 I/O
SQI_DATA2	29	**36**/61	**54**/97	141	H3			Quad SPI Data 2 I/O
SQI_DATA3	32	**40**/62	**59**/98	142	J4			Quad SPI Data 3 I/O

Table 6-22. Supply Controller (SUPC) Signals

PIN NAME	PIN NUMBER					PIN TYPE	FUNCTION TYPE	DESCRIPTION
	48 VQFN	64 TQFP	100 TQFP	144 TQFP	144 TFBGA			
SUPC_BKOUT0	21	24	37	51	G8	O	CMOS	SUPC Backup Output 0
SUPC_BKOUT1	-	-	38	52	G10	O	CMOS	SUPC Backup Output 1

Table 6-23. Timer Counter for Control (TCC) Signals

PIN NAME	PIN NUMBER					IOSET	PIN TYPE	FUNCTION TYPE	DESCRIPTION
	48 VQFN	64 TQFP	100 TQFP	144 TQFP	144 TFBGA				
TCC0_WO0	20	23	36	50	H12	1	I/O	ST/CMOS	TCC0 Waveform Outputs/ Input Counter/Capture
	-	32	50	72	D12	2			
TCC0_WO1	21	24	37	51	G8	1			
	26	33	51	73	D11	2			
TCC0_WO2	-	-	38	52	G10	1			
	27	34	52	74	C11	2			
TCC0_WO3	22	25	39	53	H9	1			
	28	35	53	75	C12	2			
TCC0_WO4	-	-	40	54	G9	1			
	29	36	54	76	B12	2			
TCC0_WO5	-	-	41	55	F10	1			
	-	37	55	77	B11	2			
TCC0_WO6	-	-	-	59	E10	1			
	-	-	-	80	C6	2			
TCC0_WO7	-	-	-	60	C9	1			
	-	-	-	81	C7	2			
TCC1_WO0	-	-	64	95	A8	1	I/O	ST/CMOS	TCC1 Waveform Outputs/ Input Counter/Capture
	32	40	59	84	B9	2			
TCC1_WO1	-		65	98	B6	1			
	33	41	60	85	D8	2			
TCC1_WO2	-	-	66	99	A6	1			
	34	42	61	86	D7	2			
TCC1_WO3	-	-	67	100	A5	1			
	-	44	63	94	B8	2			
TCC1_WO4	-	-	74	107	B3	1			
	-	-	64	95	A8	2			
TCC1_WO5	-	-	75	108	B4	1			
	36	45	68	101	B5	2			
TCC1_WO6	-	-		96	A7	1			
	37	46	71	104	A4	2			
TCC1_WO7	-	-		97	B7	1			
	-	47	72	105	A3	2			

Table 6-23. Timer Counter for Control (TCC) Signals (continued)

PIN NAME	PIN NUMBER					IOSET	PIN TYPE	FUNCTION TYPE	DESCRIPTION
	48 VQFN	64 TQFP	100 TQFP	144 TQFP	144 TFBGA				
TCC2_WO0	1	1	1	1	L2	1	I/O	ST/CMOS	TCC2 Waveform Outputs/ Input Counter/Capture
	1	**1**/56	**1**/92	136	K1	2			
TCC2_WO1	-		91	135	J2	1			
	37/45	**46**/57	**71**/93	137	K2	2			
TCC2_WO2	-	64	100	144	J3	1			
	36/46	**45**/58	**68**/94	138	L1	2			
TCC2_WO3	2	2	2	2	L3	1			
	2	**2**/61	**2**/97	141	H3	2			
TCC2_WO4	45	57	93	137	K2	1			
	45	**57**/62	**93**/98	142	J4	2			
TCC2_WO5	46	58	94	138	L1	1			
	46	**58**/63	**94**/99	143	J5	2			
TCC2_WO6	-	-	-	124	E3	1			
	-	64	100	144	J3	2			
TCC2_WO7	-	-	-	125	C3	1			
	-	-	91	135	J2	2			
TCC3_WO0	-	8	13	23	K7	1	I/O	ST/CMOS	TCC3 Waveform Outputs/ Input Counter/Capture
	10	13	22	33	L12	2			
	10	**13**	**22**	-	-	3			
TCC3_WO1	-	9	14	24	M8	1			
	-	14	23	34	L11	2			
	11	**14**	**23**	-	-	3			
TCC3_WO2	-	10	15	25	M9	1			
	12	15	24	35	K12	2			
	12	**15**	**24**	-	-	3			
TCC3_WO3	-	11	16	26	M10	1			
	13	16	25	36	K11	2			
	13	**16**	**25**	-	-	3			
TCC3_WO4	-	12	17	27	K8	1			
	14	17	26	37	K10	2			
	14	**17**	**26**	-	-	3			
TCC3_WO5	-	-	18	28	M11	1			
	15	18	27	38	K9	2			
	15	**18**	**27**	-	-	3			
TCC3_WO6	-	-		29	L9	1			
	-	-	28	39	J9	2			
	8	**10**	**15**	-	-	3			
TCC3_WO7	-	-	21	32	L10	1			
	-	-	29	40	J10	2			
	9	**11**	**16**	-	-	3			

Table 6-23. Timer Counter for Control (TCC) Signals (continued)

PIN NAME	PIN NUMBER					IOSET	PIN TYPE	FUNCTION TYPE	DESCRIPTION
	48 VQFN	64 TQFP	100 TQFP	144 TQFP	144 TFBGA				
TCC4_WO0	12	15	24	35	K12	1	I/O	ST/CMOS	TCC4 Waveform Outputs/ Input Counter/Capture
	-	-	8	12	L5	2			
	-	-		8	K5	3			
	-	-	29	40	J10	4			
TCC4_WO1	13	16	25	36	K11	1			
	-	-	9	13	M5	2			
	-	-	-	9	M2	3			
	-	-	-	41	H8	4			
TCC5_WO0	14	17	26	37	K10	1	I/O	ST/CMOS	TCC5 Waveform Outputs/ Input Counter/Capture
	-	-	10	14	K6	2			
	-	-	-	10	M3	3			
	-	-	-	42	H10	4			
TCC5_WO1	15	18	27	38	K9	1			
	-	-	11	15	J6	2			
	-	-	-	11	M4	3			
	-	-	-	43	J12	4			
TCC6_WO0	-	34	52	74	C11	1	I/O	ST/CMOS	TCC6 Waveform Outputs/ Input Counter/Capture
	-	-	45	65	C10	2			
	-	-	-	61	D10	3			
TCC6_WO1	-	35	53	75	C12	1			
	-	-	46	68	F11	2			
	-	-	-	62	C8	3			
TCC7_WO0	-	29	47	69	F12	1	I/O	ST/CMOS	TCC7 Waveform Outputs/ Input Counter/Capture
	-	36	54	76	B12	2			
	-	-	-	63	G11	3			
TCC7_WO1	-	30	48	70	E11	1			
	-	37	55	77	B11	2			
	-	-	-	64	G12	3			

Table 6-24. Full-Speed Universal Serial Bus (USB) Signals

PIN NAME	PIN NUMBER					IOSET	PIN TYPE	FUNCTION TYPE	DESCRIPTION
	48 VQFN	64 TQFP	100 TQFP	144 TQFP	144 TFBGA				
USB_DM	43	54	89	122	G1	1	I/O	Analog	USB FS D- differential data line
USB_DP	44	55	90	123	H1	1	I/O	Analog	USB FS D+ differential data line
USB_SOF_1KHZ	42	53	88	121	D3	1	O	CMOS	USB FS Start of Frame
	27	34	52	74	C11	2			

Table 6-25. High-Speed Universal Serial Bus (USBHS) Signals

PIN NAME	PIN NUMBER			PIN TYPE	FUNCTION TYPE	DESCRIPTION
	100 TQFP	144 TQFP	144 TFBGA			
USBHS_USB0ID	78	111	C2	O	CMOS	USB HS ID Detect
USBHS_USB0PN	76	109	B1	I/O	Analog	USB HS D- differential data line
USBHS_USB0PP	77	110	C1	I/O	Analog	USB HS D+ differential data line
USBHS_USBRBIAS	82	115	D2	I	Analog	270Ω ±1% resistor to AGND
USBHS_VBUS	81	114	D1	I	Analog	USB HS Bus Power Monitor

7. Power Supplies and Startup Considerations

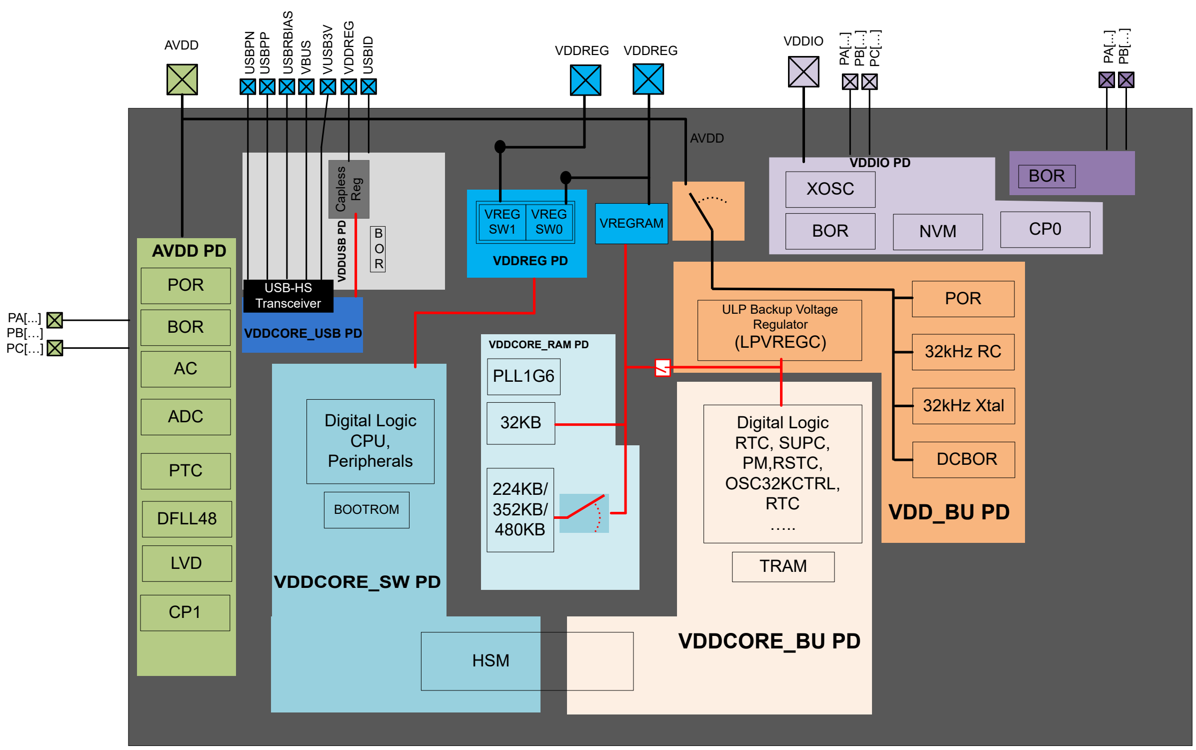
Table 7-1. External Power Supply Pins

Name	Associated Ground	Powers
AVDD	AVSS	Supplies power to the pSWITCH (in the DS SMOR), SMOR, ADCCOREn, ACDACn, OPAMPCOREn, ACCOREn, PLL0, PLL1, XOSC, DFLL48M and analog I/Os. (Nominal range: 1.71V - 3.63V, must be within 300mv of VDDIO.)
VDDIO	VSSIO	Powers I/Os. (Nominal range: 1.71 V - 3.63V, wrt VSSIO.)
VDDREG	VSSCORE	Powers all internal regulators except for the LPVREGC. (Nominal range: 1.71V - 3.63V, cannot exceed VDDIO).
VDDF	VSSF	Supplies power to the Flash panel. This pin can be tied to VDDIO outside of the device (on-board). (Nominal range: 1.71V - 3.63V, cannot exceed VDDIO).

7.1. Power Domain Overview

The following figure illustrates the block diagram of the power domain.

Figure 7-1. Power Domain Block Diagram



7.2. Power Domain Constraints

The PIC32CK SG/GC power domains are not independent of each other:

- Refer to the *Power, Ground, and Voltage Reference Pinout I/O Descriptions* to understand which pins are tied together.
- VDDIO and VDDREG share VDD. VSSIO and VSSCORE share GND. Ideally, they can be connected together outside of this device.

- VDDREG can be used as a low-voltage input to the main regulators, such as 1.8v < VDDREG < AVDD.
- AVDD has its own power and ground, AVSS.
- VDDF has its own power and ground, VSSF.
- VDDREG powers internal regulators that, in turn, power internal domains, such as VDDCORE_SW, VDDCORE_RAM, and VDDCORE_BU.

The maximum supply falling and rising rates of the different power supplies must not exceed the values described in the “Power Supply Characteristics” section of the Electrical Characteristics (85°C) chapter.

7.2.1. Power Supplies

The PIC32CK SG/GC has the following power supply pins:

- AVDD – Powers the backup power domain as well as the ADCs, Analog Comparator (AC), and the Peripheral Touch Controller (PTC).
- VDDREG – Powers internal regulators for the VDDCORE_SW, VDDCORE_RAM, and VDDCORE_USB power domains.
- VDDIO – Powers I/O lines, an External Crystal Oscillator (XOSC), the 48 MHz Digital Frequency Locked Loop (DFLL48), and two charge pumps which support the Analog Comparator (AC) and the ADCs on the device (ADC0-3 and PTC ADC). Most VDDIO pins also power the device's Flash panel through double bonding.
- VDDUSB3V – Powers USB ports on the device.

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

7.2.2. Power-on Reset (POR) and Brown-out Reset (BOR)

The PIC32CK SG/GC embeds three features to monitor, warn, and reset the device:

- A Power-on Reset on V_{DD} (AVDD):
  - Monitoring is always activated, including during device startup or during any sleep modes
  - Having V_{DD} below a fixed threshold voltage will reset the whole device

  Note: Refer to the Electrical Characteristics (85°C) for the rising and falling threshold voltages.
- A Brown-out Reset (VDDBOR) on V_{DD} (VDDREG, AVDD, VDDIO, VDDUSB3V):
  - The VDDBOR monitors VDD continuously
  - dcBOR in (continuous mode) or periodically (sampled mode) with a programmable sample frequency in Backup mode
  - A programmable threshold loaded from the USER CFG page is used to trigger an interrupt and/or reset the whole device
- PORCORE a Brown-out Detector on VDDCORE
  Note: PORCORE is calibrated in production and its calibration parameters are stored in the USER CFG page. This data must not be changed to ensure correct device behavior.

7.2.3. Voltage Regulator

The PIC32CK SG/GC internal Main Voltage Regulators have three different modes, controlled by the Power Manager (PM).

- **Enabled mode:** This is the default mode when CPU and peripherals are running.
- **Low-Power mode:** This is the mode when CPU and peripherals are in standby.
- **Disabled mode:** When the chip is in backup mode, the internal regulator is off, the VDDCORE_SW core power domain is OFF. The VDDCORE_BU backup domain is powered by the backup regulator (LPVREGC).

7.2.4. Internal Power Domains

A variety of internal regulated power domains (nominal 1.2v) are as follows:

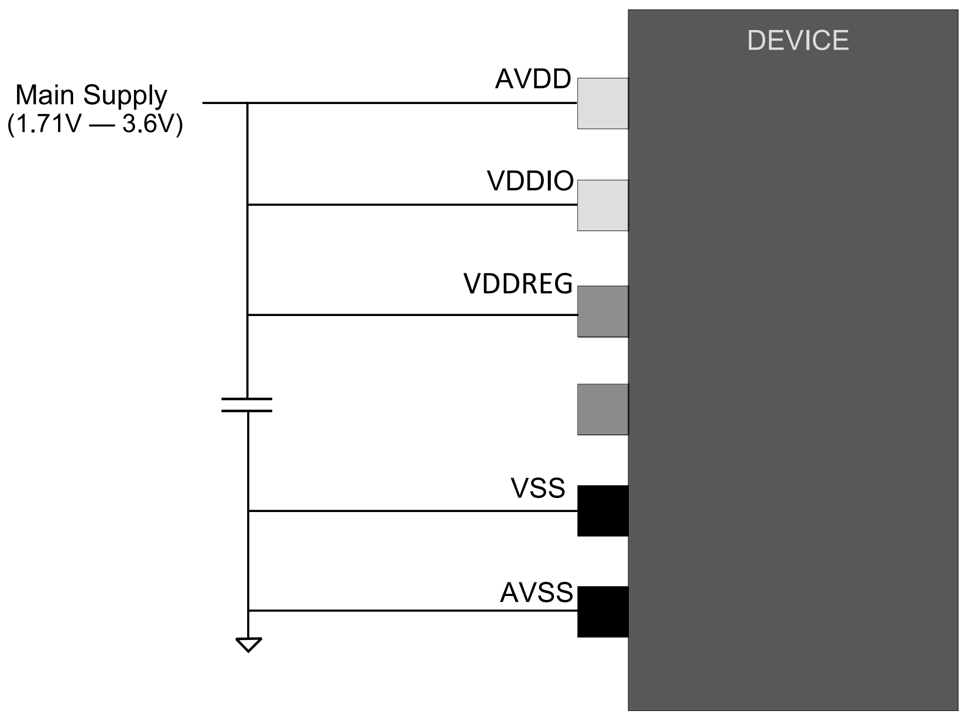
- **VDDCORE_BU**: This domain powers the backup domain.
- **VDDCORE_SW**: The main voltage domain for the CPU, bus, and most peripherals.
- **VDDCORE_USB**: This domain powers the USB PHY.
- **VDDCORE_RAM**: This domain is used to retain the System SRAM and the PLL.

7.2.5. Typical Powering Schematic

The PIC32CK SG/GC uses a single supply from 1.71V to 3.6V.

The following figure shows the recommended power supply connection.

Figure 7-2. Power Supply Connection for Enabled Mode Only



7.2.6. Power Up Sequence

7.2.6.1. Supply Order

The VDDIO and AVDD must have the same supply sequence. Ideally, they must be connected together outside of this device.

VDDF supplies power to the Flash panel. This pin can also be tied to VDDIO outside of this device.

VDDREG can be used as a low-voltage input to the main regulators, such as 1.8v < VDDREG < AVDD.

7.2.6.2. Minimum Rise Rate

The integrated Power-on Reset (POR) circuits monitoring AVDD requires a minimum rise rate.

7.2.6.3. Maximum Rise Rate

The rise rate of the power supplies must not exceed the values described in the “Electrical Characteristics” chapter.

7.3. Power-Up

This section summarizes the power-up sequence of the PIC32CK SG/GC. The behavior after power-up is controlled by the Power Manager (RPMU/PM).

7.3.1. Starting of Internal Regulator

After power-up of the VDDREG, VDDIO and AVDD supplies (VDDx > POR voltage), the device is set to its initial state and kept in Reset, until the band-gap and internal regulator power has stabilized throughout the device.

The internal regulator provides the internal VDDCORE_SW once the external voltage VDDIO/AVDD and the internal VDDCORE_xx domains reach a stable value, the internal System Reset is released.

7.3.2. Starting of Clocks

Once the power has stabilized and the internal Reset is released, the device will use a 48 MHz clock by default. The clock source for this clock signal is DFLL48M, which is enabled after a reset by default. This is also the default time base for Generic Clock Generator 0. In turn, Generator 0 provides the main clock GCLK_MAIN which is used by the Main Clock module (MCLK).

Some synchronous system clocks are active after start-up, allowing software execution. Refer to the *Clock Mask Registers* section in the *Main Clock (MCLK)* chapter for a list of clocks that are running by default. Synchronous system clocks that are running receive the 48 MHz clock from Generic Clock Generator 0. Other generic clocks are disabled.

7.3.3. I/O Pins

After power-up, the I/O pins are tri-stated except RESETn, which is pull-up enabled and configured as an input.

7.3.4. Fetching of Initial Instructions

After Reset has been released, the CPU starts fetching PC and SP values from the Reset address, 0x0400_0000. This points to the Boot ROM firmware. The code read from the internal Flash can be used to configure the clock system and clock sources. See the related peripheral documentation for details. Refer to the Arm “Architecture Reference Manual” for more information on CPU startup (www.arm.com).

7.4. Power-on Reset and Brown-out Reset

The PIC32CK SG/GC family of devices embed the following features to monitor, warn, and reset the device:

- **POR**: Power-on Reset on the main supply VDD (AVDD)
- **dcBOR**: Duty cycled Brown-out Reset on VDDREG/AVDD/VDDIO
- **VDDBORxx**: Brown-out Reset on VDDREG/AVDD/VDDIO/VDDUSB3V
- **PORCORE**: Brown-out Detector internal to the VDDCORE_SW power domain. The PORCORE is calibrated in production and its calibration parameters are stored in the USER CFG page. This data should not be changed if the User Row is written to assure correct behavior.
  Note: The output of the USB PHY regulators is monitored by the SUPC.

7.4.1. Power-on Reset (POR) on the Main Supply VDD (AVDD)

The main supply VDD (AVDD) is monitored by POR. Monitoring is always activated, including start up and all sleep modes. If VDD goes below the threshold voltage, all I/Os supplied by VDDIO are reset.

7.4.2. Brown-out Detector on VDDCORE

Once the device has started up, PORCORE monitors the internal VDDCORE_* power domains.

7.5. Analog Peripherals Considerations

This chapter provides a global view of the analog system, which is composed of the following analog peripherals: AC and ADC.

The analog peripherals can be connected to each other as illustrated in the block diagrams for the individual analog modules.

Important:
When an analog peripheral is enabled, each analog output of the peripheral will be prevented from using the alternative functions of the output pads. This is also true even when the peripheral is used for internal purposes.

Analog inputs do not interfere with alternate pad functions.

7.6. Device Startup

This section summarizes the PIC32CK SG/GC device startup sequence which starts after device power-up. After power-up, the device is kept in reset until the power has stabilized throughout the device. Once VDDIO/AVDD and VDDCORE voltages reach a stable value, the internal reset is released.

7.6.1. Clocks Startup

The device selects the OSC48M oscillator which is enabled by default after reset and configured at 48 MHz.

This 48 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 GLCK_MAIN clock.
Note: Other generic clocks are disabled to optimize power consumption.

Some synchronous clocks also require to be active after startup.

Note: These active synchronous clocks also receive the 48 MHz clock from Generic Clock Generator 0.

Refer to the *Clock Mask Register* section in the *Main Clock (MCLK)* chapter to obtain the list of clocks that are running by default.

7.6.2. Initial Instructions Fetching

After reset is released, the CPU starts fetching from the Boot Flash (BFM).

Unless a debugger is connected and places the Boot Flash (BFM) 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. Before jumping to the Flash, the Boot Flash (BFM) resets the two first 2 kB of SRAM. The Clocks remain unchanged.
Note: The PIC32CK SG/GC Boot Interactive mode allows a debugger to perform several actions on the device, such as NVM areas integrity check, chip erase, and so on. Refer to the “Boot Flash (BFM)” section for more information.

In addition, the PIC32CK SG/GC Boot Flash (BFM) has extra security features, such as device integrity checks, memories and peripherals security attributions, and secure boot that can be executed before jumping to the Flash in Secure state.

7.6.3. I/O Pins

After reset, the I/O pins are tri-stated except the PA02 and PA03 pins (configured as an input with pull-up enabled). These pins, by default, are assigned to TMS_SWDIO for the PA02 pin, and to the SWCLK peripheral function for the PA03 pin, which allows for debugger probe detection.

8. Product Mapping

8.1. Code Address Space

Table 8-1. Code Address Space

Start Adress	End Address	Size	Device Configuration:		IDAU		
Code Address Space			2 MB Flash, 512 KB RAM	1 MB Flash, 256 KB RAM	Disabled	Enabled	ID
0x0000_0000	0x0000_0FFF	4KB	TCM DATARAM		EXEMPT	MR_SEC	1
0x0000_1000	0x0000_13FF	1KB	TCM TAGRAM		EXEMPT	MR_SEC	2
0x0000_1400	0x0000_2FFF	RSVD	RSVD		INVALID	INVALID	0
0x0000_3000	0x0000_33FF	1KB	TCM VALIDRAM		EXEMPT	MR_SEC	3
0x0000_3400	0x07FF_FFFF	RSVD	RSVD		INVALID	INVALID	0
0x0400_0000	0x0400_7FFF	32KB	PRM (Host Boot ROM)		SEC	MR_SEC	4
0x0400_8000	0x07FF_FFFF	RSVD	RSVD		INVALID	INVALID	0
0x0800_0000	0x0801_FFFF	128KB	Boot Flash Memory (BFM)		EXEMPT	MR_SEC_NSC_W	5/6
0x0802_0000	0x09FF_FFFF	RSVD	RSVD		INVALID	INVALID	0
0x0A00_0000	0x0A00_FFFF	64KB	Configuration Flash Memory (CFM)		EXEMPT	MR_SEC	7
0x0A01_0000	0x0BFF_FFFF	RSVD	RSVD		INVALID	INVALID	0
0x0C00_0000	0x0C07_FFFF	512KB	PFM0		EXEMPT	MR_SEC_NSC_NONSEC_W	8/9/10
0x0C08_0000	0x0C0F_FFFF	512KB	PFM1				
0x0C10_0000	0x0C1F_FFFF	1MB	PFM2	RSVD			
0x0C20_0000	0x0FFF_EFFF	RSVD	RSVD		INVALID	INVALID	0
0x0FFF_F000	0x0FFF_FFFF	4KB	SQI (SFR)		EXEMPT	MR_SECnNONSEC	11
0x1000_0000	0x1FFF_FFFF	256MB	SQI (XIP)		EXEMPT	MR_SECnNONSEC_L	12

8.2. SRAM Address Space

Table 8-2. SRAM Address Space

Start Adress	End Address	Size	Device Configuration:		IDAU		
SRAM Address Space			2 MB Flash, 512 KB RAM	1 MB Flash, 256 KB RAM	Disabled	Enabled	ID
0x2000_0000	0x2000_7FFF	32KB	FLEXRAM		EXEMPT	MR_SEC_NONSEC_W	13/14
0x2000_8000	0x2000_FFFF	32KB	FLEXRAM				
0x2001_0000	0x2001_FFFF	64KB	FLEXRAM				
0x2002_0000	0x2003_FFFF	128KB	FLEXRAM				
0x2004_0000	0x2007_FFFF	256KB	FLEXRAM	RSVD			
0x2008_0000	0x3FFF_FFFF	RSVD	RSVD		INVALID	INVALID	0

8.3. Peripheral Address Space

Table 8-3. Peripheral Address Space

Start Address	End Address	Size	Device Configuration:		IDAU		
Peripheral Address Space			2 MB Flash, 512 KB RAM	1 MB Flash, 256 KB RAM	Disabled	Enabled	ID
0x4000_0000	0x43FF_FFFF	RSVD	RSVD		INVALID	INVALID	0
0x4400_0000	0x4403_FFFF	256KB	APB A		MR_EXEMPT	MR_EXEMPT	15
0x4404_0000	0x447F_FFFF	RSVD	RSVD		INVALID	INVALID	0
0x4480_0000	0x4483_FFFF	256KB	APB B		MR_EXEMPT	MR_EXEMPT	16
0x4484_0000	0x44FF_FFFF	RSVD	RSVD		INVALID	INVALID	0
0x4500_0000	0x4503_FFFF	256KB	APB C		MR_EXEMPT	MR_EXEMPT	17

Table 8-3. Peripheral Address Space (continued)

Start Address	End Address	Size	Device Configuration:		IDAU		
Peripheral Address Space			2 MB Flash, 512 KB RAM	1 MB Flash, 256 KB RAM	Disabled	Enabled	ID
0x4580_0000	0x45FF_FFFF	RSVD	RSVD		INVALID	INVALID	0
0x4600_0000	0x4600_0FFF	8KB	USBHS		EXEMPT	MR_SECnNONSEC	18
0x4600_1000	0x47FF_FFFF	RSVD	RSVD		INVALID	INVALID	0
0x4800_0000	0x4800_03FF	1KB	HSM MAILBOX		EXEMPT	MR_SECnNONSEC	19
0x4800_0400	0x5FFF_FFFF	RSVD	RSVD		INVALID	INVALID	0

8.4. External RAM Address Space

Table 8-4. External RAM Address Space

Start Adress	End Address	Size	Device Configuration:		IDAU		
Ext. RAM Space			2 MB Flash, 512 KB RAM	1 MB Flash, 256 KB RAM	Disabled	Enabled	ID
0x6000_0000	0x60FF_FFFF	16MB	EBI (CS0)		MR_EXEMPT	MR_SECnNONSEC_L	20
0x6100_0000	0x61FF_FFFF	16MB	EBI (CS1)		MR_EXEMPT	MR_SECnNONSEC_L	21
0x6200_0000	0x62FF_FFFF	16MB	EBI (CS2)		MR_EXEMPT	MR_SECnNONSEC_L	22
0x6300_0000	0x63FF_FFFF	16MB	EBI (CS3)		MR_EXEMPT	MR_SECnNONSEC_L	23
0x6400_0000	0x9FFF_FFFF	RSVD	RSVD		INVALID	INVALID	0

8.5. Peripheral Bus A Address Map

Table 8-5. Peripheral Bus A Address Map

Peripheral Bus A Address Map		
Peripheral on APB	Start Address	End Address
DSU	0x4400_0000	0x4400_1FFF
FCR	0x4400_2000	0x4400_3FFF
FCW	0x4400_4000	0x4400_5FFF
PM	0x4400_6000	0x4400_FFFF
SUPC	0x4400_8000	0x4401_1FFF
RSTC	0x4400_A000	0x4401_FFFF
OSCCTRL	0x4400_C000	0x4402_1FFF
OSC32KCTRL	0x4400_E000	0x4402_FFFF
GCLK	0x4401_0000	0x4403_1FFF
MCLK	0x4401_2000	0x4403_FFFF
FREQM	0x4401_4000	0x4404_1FFF
WDT	0x4401_6000	0x4404_3FFF
RTC	0x4401_8000	0x4404_FFFF
EIC	0x4401_A000	0x4405_1FFF
PAC	0x4401_C000	0x4405_3FFF
TRAM	0x4401_E000	0x4405_FFFF

8.6. Peripheral Bus B Address Map

Table 8-6. Peripheral Bus B Address Map

Peripheral Bus B Address Map		
Peripheral on APB	Start Address	End Address
PORT	0x4480_0000	0x4480_1FFF
DMA0	0x4480_2000	0x4480_FFFF

Table 8-6. Peripheral Bus B Address Map (continued)

Peripheral Bus B Address Map		
Peripheral on APB	Start Address	End Address
DMA1	0x4480_4000	0x4481_1FFF
HMATRIX	0x4480_6000	0x4481_FFFF
CMCC	0x4480_8000	0x4482_1FFF
PRM	0x4480_A000	0x4482_3FFF
IDAU	0x4480_C000	0x4482_7FFF
EVSYS	0x4480_E000	0x4483_FFFF
SERCOM0	0x4481_0000	0x4484_1FFF
SERCOM1	0x4481_2000	0x4484_FFFF
SERCOM2	0x4481_4000	0x4485_1FFF
SERCOM3	0x4481_6000	0x4485_FFFF
TCC0	0x4481_8000	0x4486_1FFF
TCC1	0x4481_A000	0x4486_FFFF
TCC2	0x4481_C000	0x4487_1FFF
TCC3	0x4481_E000	0x448E_FFFF

8.7. Peripheral Bus C Address Map

Table 8-7. Peripheral Bus C Address Map

Peripheral Bus C Address Map		
Peripheral on APB	Start Address	End Address
SERCOM4	0x4500_0000	0x4500_1FFF
SERCOM5	0x4500_2000	0x4500_3FFF
SERCOM6	0x4500_4000	0x4500_5FFF
SERCOM7	0x4500_6000	0x4500_FFFF
TCC4	0x4500_8000	0x4501_1FFF
TCC5	0x4500_A000	0x4501_3FFF
TCC6	0x4500_C000	0x4501_5FFF
TCC7	0x4500_E000	0x4501_FFFF
ADC	0x4501_0000	0x4502_FFFF
AC	0x4501_2000	0x4503_1FFF
PTC	0x4501_4000	0x4505_FFFF
I2S	0x4501_6000	0x4506_1FFF
PCC	0x4501_8000	0x4506_3FFF
CCL	0x4501_A000	0x4506_FFFF
PDEC	0x4501_C000	0x4507_1FFF
CAN0	0x4501_E000	0x450F_FFFF
CAN1	0x4502_0000	0x4502_1FFF
ETH	0x4502_2000	0x4502_3FFF
TRNG	0x4502_4000	0x4502_5FFF
SDMMC0	0x4502_6000	0x4502_7FFF
SDMMC1	0x4502_8000	0x4502_9FFF
USBHS	0x4502_A000	0x4502_BFFF
EBI	0x4502_C000	0x4502_CFFF

8.8. Peripheral Bus AHB Address Map

Table 8-8. Peripheral Bus AHB Address Map

Peripheral Bus AHB Address Map		
Peripheral on AHB	Start Address	End Address
RSVD	0x4F00_0000	0x4F00_7FFF
SQI0	0x4F00_8000	0x4F00_8FFF
SQI1	0x4F00_9000	0x4F00_9FFF
HSUSB 0	0x4F01_0000	0x4F01_1FFF
HSUSB 1	0x4F01_2000	0x4F01_3FFF

8.9. Flash CFM Configuration Address Map

Table 8-9. Flash CFM Configuration Address Map

FLASH CFM CONFIGURATION ADDRESS MAP				
Start Address	End Address	Size	Panel	Contents
0x0A00_0000	0x0A00_0FFF	4 KB	1	User CFG-1
0x0A00_1000	0x0A00_1FFF	4 KB	1	User NVMOTP-1
0x0A00_2000	0x0A00_2FFF	4 KB	1	BOOT CFG-1
0x0A00_3000	0x0A00_3FFF	4 KB	1	RSVD
0x0A00_4000	0x0A00_4FFF	4 KB	1	DAL CFG
0x0A00_5000	0x0A00_5FFF	4 KB	1	RSVD
0x0A00_6000	0x0A00_6FFF	4 KB	1	CAL-Backup
0x0A00_7000	0x0A00_7FFF	4 KB	1	CAL-OTP, CAL, Variant, FFF, F2RR, F1RR
0x0A00_8000	0x0A00_8FFF	4 KB	2	User CFG-2
0x0A00_9000	0x0A00_9FFF	4 KB	2	User NVMOTP-2
0x0A00_A000	0x0A00_AFFF	4 KB	2	BOOT CFG-2
0x0A00_B000	0x0A00_BFFF	4 KB	2	RSVD
0x0A00_C000	0x0A00_CFFF	4 KB	2	RSVD
0x0A00_D000	0x0A00_DFFF	4 KB	2	RSVD
0x0A00_E000	0x0A00_EFFF	4 KB	2	RSVD
0x0A00_F000	0x0A00_FFFF	4 KB	2	RSVD

9. Peripherals

9.1. Register Description

9.1.1. 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). The 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.*

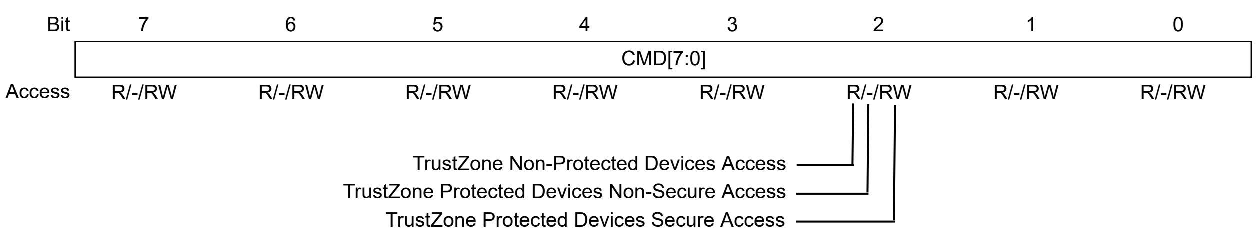
Enable-Protected Register Property:

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:

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.

In the following register descriptions, the access permissions are specified.



For additional information, refer to Implementation Defined Attribution Unit (IDAU).

Bitfield Access Properties:

The access properties of bit fields within a register are defined as follows:

Table 9-1. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

9.1.2. Registers Access Permissions

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: PORT, EIC
- 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 (PORT, EIC):

- 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 behaves like a Mix-Secure peripheral

For additional information, refer to Implementation Defined Attribution Unit (IDAU).

9.1.3. Register Synchronization

9.1.3.1. Overview

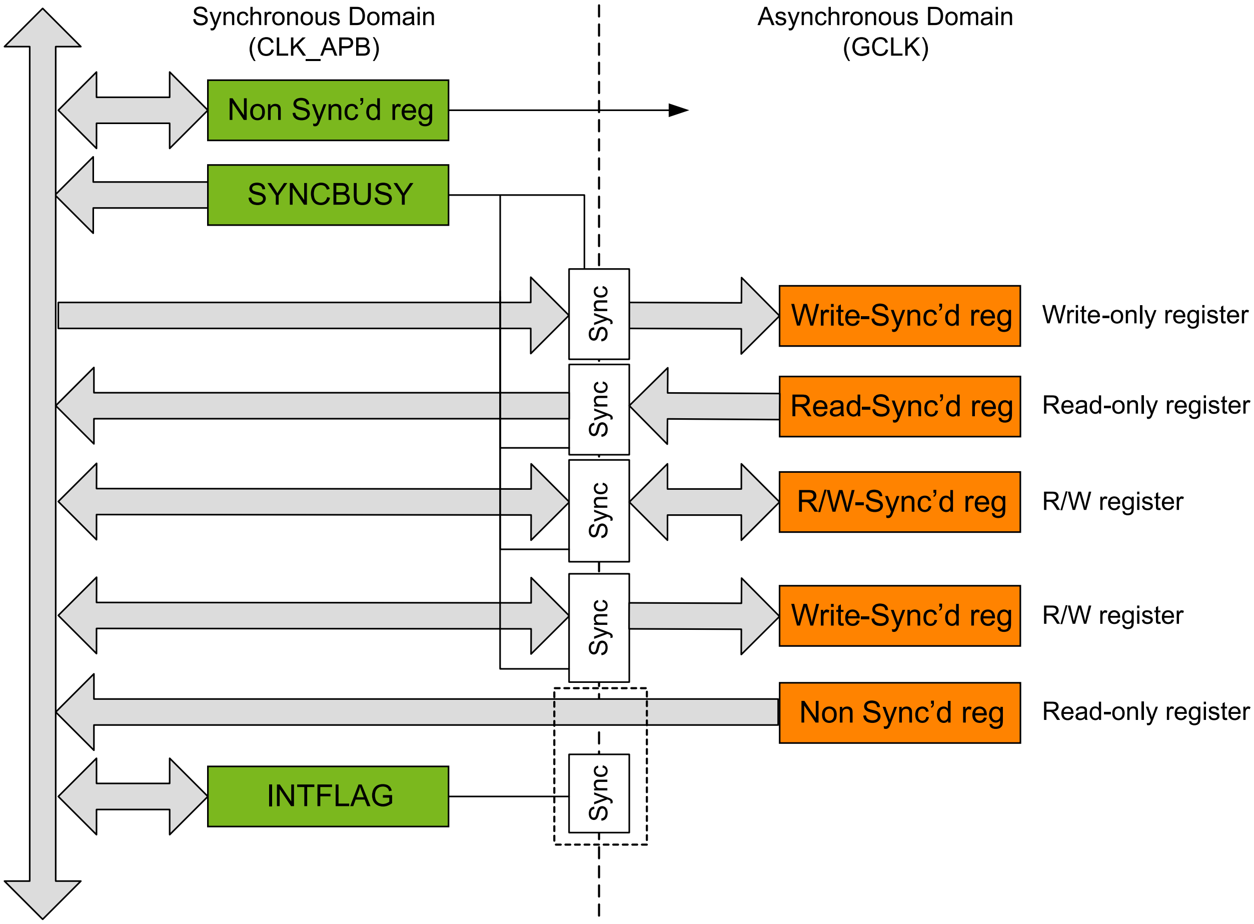
Most of the peripherals are composed of one digital bus interface connected to the APB or AHB bus and running from a corresponding clock in the Main Clock domain, and one peripheral core running from the peripheral Generic Clock (GCLK).

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 read-synchronization and write-synchronization, the corresponding bit in the SYNCBUSY register is shared.

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

Figure 9-1. Register Synchronization Overview



9.1.3.2. 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 *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)*.

For example, the REGA, REGB are 8-bit core registers, and the REGC is a 16-bit core register.

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

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

The REGC (16-bit access) can be written without affecting the REGA or REGB registers. If the REGC register 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. The REGA, REGB, and REGC registers can be updated at different times because of independent write synchronization.

9.1.3.3. General Read Synchronization

Read-synchronized registers are synchronized each time the register value is updated but the corresponding SYNCBUSY bits are not set. Reading a read-synchronized register does not start a new synchronization, it returns the last synchronized value.

Note: The corresponding bits in SYNCBUSY will automatically be set when the device wakes up from sleep because read-synchronized registers need to be synchronized. Therefore reading a read-synchronized register before its corresponding SYNCBUSY bit is cleared will return the last synchronized value before sleep mode.
Moreover, if a register is also write-synchronized, any write access while the SYNCBUSY bit is set will be discarded and generate an error.

9.1.3.4. 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'.

9.1.3.5. Synchronization Delay

The synchronization will delay read and writer 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. Processor and Architecture

Refer to Arm reference documents *ARM Cortex-M33 Processor Data Sheet* and *ARM Cortex-M33 Technical Reference Manual*, available for download at the following location: www.arm.com.

10.1. Overview

The following table provides the configuration for the Arm Cortex-M33 processor in the PIC32CK SG/GC devices.

Table 10-1. Arm Cortex-M33 Configuration

Features	Configuration
Debug	
Comparator set	Full comparator set: 4 DWT and 8 FPB comparators
ETM support	Instruction ETM interface
Internal Trace support (ITM)	ITM and DWT trace functionality implemented
CTI and WIC	Not embedded
TCM	
ITCM maximum size	128 KB
DTCM maximum size	128 KB
Cache	
Cache size	16 KB for instruction cache, 16 KB for data cache
Number of sets	256 for instruction cache, 128 for data cache
Number of ways	2 for instruction cache, 4 for data cache
Number of words per cache line	8 words (32 bytes)
ECC on Cache	Embedded
NVIC	
IRQ number	158
IRQ priority levels	8
MPU	
Number of regions	8
FPU	
FPU precision	Single precision

10.2. Nested Vectored Interrupt Controller

10.2.1. Overview

The Nested Vectored Interrupt Controller (NVIC) provides the user interface to control the external and on-chip peripheral interrupts. The NVIC provides up to 158 interrupt sources (unused sources are reserved for future use) that can be programmed with eight different priority levels. The NVIC is included as part of the CPU. For more details, refer to the “Cortex-M33 Technical Reference Manual”, which is available for download at (www.arm.com).

10.2.2. NVIC Interrupt Line Mapping

Each of the interrupt lines is connected to a single 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 (INTEN- CLR) register.

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

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

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

Table 10-2. NVIC Interrupt Mapping

NVIC Interrupt Mapping	
Module	**Interrupt Index: Source**
FCR - PFM READ Controller	0: ECERR
	1 : FAULT or CRCERR FLT, CRC
FCW - PFM WRITE Controller	2 : FCW MAIN
PM - Power Manager	3 : SLEEPRDY
SUPC - Supply Controller	4 : LVDET, LVDRDY, BORVDDUSB_0, ADDVREGRDY_0
OSCCTRL - Oscillators Control	5 : XOSCFAIL, XOSCRDY, CLKFAIL
	6 : DFLLRDY, DFLLLOCK, DFLLOVF, DFLLUNF, DFLLRCS, DFLLFAIL
	7 : PLLLOCKR
OSC32KCTRL- 32kHz Oscillators Control	8 : XOSC32KRDY, XOSC32KFAIL
MCLK - Main Clock	9 : CKRDY
FREQM - Frequency Meter	10 : DONE, WINMON
WDT - Watchdog Timer	11 : EW
RTC - Real-Time Counter	12 : PER_0, PER_1, PER_2, PER_3, PER_4, PER_5, PER_6, PER_7, CMP_0, CMP_1, CMP_2, CMP_3, TAMPER, OVF
EIC - External Interrupt Controller	NMI : EXTINT_NMI
	13 : EXTINT_0
	14 : EXTINT_1
	15 : EXTINT_2
	16 : EXTINT_3
	17 : EXTINT_4
	18 : EXTINT_5
	19 : EXTINT_6
	20 : EXTINT_7
	21 : EXTINT_8
	22 : EXTINT_9
	23 : EXTINT_10
	24 : EXTINT_11
	25 : EXTINT_12
	26 : EXTINT_13
	27 : EXTINT_14
	28 : EXTINT_15
	29 : NSCHK
PAC - Peripheral Access Controller	30 : ERR
TRAM - Trust RAM	31 : ERR, DRP
PORT - Port A/B/C/D	32 : NSCHK_0, NSCHK_1, NSCHK_2, NSCHK_3

Table 10-2. NVIC Interrupt Mapping (continued)

NVIC Interrupt Mapping	
Module	**Interrupt Index: Source**
DMA0 - Direct Memory Access Controller 0	33 : Priority 0
	34 : Priority 1
	35 : Priority 2
DMA1 - Direct Memory Access Controller 1	36 : Priority 0
	37 : Priority 1
HMATRIX	38 : CMCC, PRM HMATRIX
EVSYS - Event System Interface	39 : EVD_0, OVR_0
	40 : EVD_1, OVR_1
	41 : EVD_2, OVR_2
	42 : EVD_3, OVR_3
	43 : EVD_4, OVR_4, EVD_5, OVR_5, EVD_6, OVR_6, EVD_7, OVR_7, EVD_8, OVR_8, EVD_9, OVR_9, EVD_10, OVR_10, EVD_11, OVR_11, NSCHK
SERCOM0 - Serial Communication Interface 0	44 : ERROR, RXBRK
	45 : DRE/PREC
	46 : TXC/AMATCH
	47 : RXC/DRDY
	48: RXS/SSL/TXFE
	49 : CTSIC/RXFF
SERCOM1 - Serial Communication Interface 1	50 : ERROR, RXBRK
	51 : DRE/PREC
	52 : TXC/AMATCH
	53 : RXC/DRDY
	54 : RXS/SSL/TXFE
	55 : CTSIC/RXFF
SERCOM2 - Serial Communication Interface 2	56 : ERROR, RXBRK
	57 : DRE/PREC
	58 : TXC/AMATCH
	59 : RXC/DRDY
	60 : RXS/SSL/TXFE
	61 : CTSIC/RXFF
SERCOM3 - Serial Communication Interface 3	62 : ERROR, RXBRK
	63 : DRE/PREC
	64 : TXC/AMATCH
	65 : RXC/DRDY
	66 : RXS/SSL/TXFE
	67 : CTSIC/RXFF
TCC0 - Timer Counter Control 0	68 : CNT, DFS, ERR, FAULTA, FAULTB, FAULT0, FAULT1, OVF, TRG, UFS
	69 : MC_0
	70 : MC_1
	71 : MC_2
	72 : MC_3
	73 : MC_4
	74 : MC_5

Table 10-2. NVIC Interrupt Mapping (continued)

NVIC Interrupt Mapping	
Module	**Interrupt Index: Source**
TCC1 - Timer Counter Control 1	75 : CNT, DFS, ERR, FAULTA, FAULTB, FAULT0, FAULT1, OVF, TRG, UFS
	76 : MC_0
	77 : MC_1
	78 : MC_2
	79 : MC_3
	80 : MC_4
	81 : MC_5
TCC2 - Timer Counter Control 2	82 : CNT, DFS, ERR, FAULTA, FAULTB, FAULT0, FAULT1, OVF, TRG, UFS
	83 : MC_0
	84 : MC_1
	85 : MC_2
	86 : MC_3
	87 : MC_4
	88 : MC_5
TCC3 - Timer Counter Control 3	89 : CNT, DFS, ERR, FAULTA, FAULTB, FAULT0, FAULT1, OVF, TRG, UFS
	90 : MC_0
	91 : MC_1
	92 : MC_2
	93 : MC_3
	94 : MC_4
	95 : MC_5
SERCOM4 - Serial Communication Interface 4	96 : ERROR, RXBRK
	97 : DRE/PREC
	98 : TXC/AMATCH
	99 : RXC/DRDY
	100 : RXS/SSL/TXFE
	101 : CTSIC/RXFF
SERCOM5 - Serial Communication Interface 5	102 : ERROR, RXBRK
	103 : DRE/PREC
	104 : TXC/AMATCH
	105 : RXC/DRDY
	106 : RXS/SSL/TXFE
	107 : CTSIC/RXFF
SERCOM6 - Serial Communication Interface 6	108 : ERROR, RXBRK
	109 : DRE/PREC
	110 : TXC/AMATCH
	111 : RXC/DRDY
	112 : RXS/SSL/TXFE
	113 : CTSIC/RXFF

Table 10-2. NVIC Interrupt Mapping (continued)

NVIC Interrupt Mapping	
Module	**Interrupt Index: Source**
SERCOM7 - Serial Communication Interface 7	114 : ERROR, RXBRK
	115 : DRE/PREC
	116 : TXC/AMATCH
	117 : RXC/DRDY
	118 : RXS/SSL/TXFE
	119 : CTSIC/RXFF
TCC4 - Timer Counter Control 4	120 : CNT, DFS, ERR, FAULTA, FAULTB, FAULT0, FAULT1, OVF, TRG, UFS
	121 : MC_0
	122 : MC_1
TCC5 - Timer Counter Control 5	123 : CNT, DFS, ERR, FAULTA, FAULTB, FAULT0, FAULT1, OVF, TRG, UFS
	124 : MC_0
	125 : MC_1
TCC6 - Timer Counter Control 6	126 : CNT, DFS, ERR, FAULTA, FAULTB, FAULT0, FAULT1, OVF, TRG, UFS
	127 : MC_0
	128 : MC_1
TCC7 - Timer Counter Control 7	129 : CNT, DFS, ERR, FAULTA, FAULTB, FAULT0, FAULT1, OVF, TRG, UFS
	130 : MC_0
	131 : MC_1
ADC - Analog Digital Controller	132 : ADC Global Interrupt, 133 : ADC Core 0 Interrupt, 134-137 : Reserved
AC - Analog Comparator	138 : COMP_0, COMP_1, WIN_0
PTC - Peripheral Touch	139 : EOC, WCOMP, ACRRDY
SPI_IXS - AUDIO SPI IxS	140 : SPI_IXS
PCC - Parallal Capture Controller	141 : PCC
PDEC	142 : DIR, ERR, MC_0, MC_1, OVF, VLC
CAN0 - Control Area Network	143 : LINE_0, LINE_1, ERROR
CAN1 - Control Area Network	144 : LINE_0, LINE_1, ERROR
ETH - Ethernet MAC	145 : Q_0
SQI - Quad SPI interface	146 : SQI
TRNG - True Random Generator	147 : IS0
SDMMC0- SD/MMC Host Controller 0	148 : LINE, TIMER
SDMMC1 - SD/MMC Host Controller 1	149 : LINE, TIMER
USBFS - Full-Speed Universal Serial Bus	150 : EORSM_DNRSM, EORST_RST, LPM_DCONN, LPMSUSP_DDISC, MSOF, RAMACER, RXSTP_TXSTP[0:7], STALL0_STALL[0:7], STALL1[0:7], SUSPEND, TRFAIL0_TRFAIL[0:7], TRFAIL1_PERR[0:7], UPRSM, WAKEUP
	151 : SOF_HSOF
	152 : TRCPT0[0:7]
	153 : TRCPT1[0:7]
USBHS - High Speed Universal Serial Bus	154 : USBHS
HSM	155 : ERROR, TAMPER
	156 : TXINT
	157 : RXINT

Note: NVIC interrupt lines for modules which are not present on a specific part will be non-functional.

10.3. High-Speed Bus

10.3.1. Overview

The Bus Matrix implements a multi-layer AHB, based on the AHB-Lite protocol, that enables parallel access paths between multiple AHB Hosts and Clients in a system, therefore increasing the overall bandwidth. The normal latency to connect a Host to a Client is one cycle except for the default Host of the accessed Client which is connected directly (zero cycle latency).

10.3.2. Bus Matrix Connectivity

The following figure shows the connectivity between Initiators and Targets.

There are three APB shared buses (APB A-C) which provide access to all peripheral SFR registers. This is a shared connection as opposed to a dedicated connection provided by crossbars. Therefore, only one Host may communicate to one particular APB shared bus at a time. During this time, no other Host may communicate to an APB target on that particular APB shared bus. However, there are no restrictions for two initiators to communicate to two different APB shared buses at the same time.

Access to Data RAM Memory (DRM) is supported by five AHB target read/write ports (numbered 0-4) on the Multi-Channel RAM Controller (MCRAMC). The number shown for each MCRAMC initiator indicates which port is used.

Table 10-3. Bus Matrix Connectivity

			High Speed Bus Targets													
												MCRAMC Channels				
			FCR	ABH-APB Bridge A	ABH-APB Bridge B	ABH-APB Bridge C	HSM Mailbox	SQI	EBI	USBHS	BROMC	RAM (512/256/128 KB)				
												MCRAMC Ports				
												0	1	2	3	4
			0	1	2	3	4	5	6	7	8	9	10	11	12	
High Speed Bus Initiators	CM33 Sys	0		x	x	x	x		x	x		x				
	CMCC	1	x					x			x					
	DMA0 RD	2	x	x	x	x			x					x		
	DMA0 WR	3		x	x	x			x						x	
	DMA1 RD	4	x	x	x	x			x					x		
	DMA1 WR	5		x	x	x			x						x	
	SDMMC0	6											x			
	SDMMC1	7											x			
	CAN0	8											x			
	CAN1	9											x			
	ETH	10											x			
	DSU	11	x	x	x	x		x	x	x	x		x			
	SQI DMA	12											x			
	FCW	13											x			
	USBHS	14	x										x			
	HSM AUX HSM DMA	#	x	x	x	x		x	x	x			x			
	USBFS															x

Notes:
1. The FCR Controller supports one AHB target port.
2. Data RAM (DRM) access occurs through the Multi-Channel RAM Controller (MCRAMC), which provides five AHB target ports numbered 0-4. The number shown indicates which port is used.

11. Memories

11.1. Embedded Memories

- Internal high-speed Flash with Read-While-Write (RWW - Live Update) capability, (i.e., Erase/Write on one Flash panel while executing code from other Flash panel, Flash divided equally into two panels.)
- Internal high-speed RAM, single-cycle access at full speed
- Internal backup RAM, single-cycle access at full speed

11.2. Physical Memory Map

The high-speed bus is implemented as a bus matrix. All high-speed bus addresses are fixed and are never remapped in any way, even during boot. The 32-bit physical address space is mapped as follows:

Table 11-1. Physical Memory Map

Memory	Start Address	Size in KB (unless otherwise stated)		
		PIC32CK2051xx	PIC32CK1025xx	PIC32CK1012xx
Tightly Coupled Memory (TCM)	0x0000_0000	4	4	4
Boot Flash Memory	0x0800_0000	128	128	128
Configuration Flash Memory	0x0A00_0000	64	64	64
Program Flash Memory	0x0C00_0000	2048	1024	1024
Embedded SRAM	0x2000_0000	512	256	128
Peripheral Bridge A	0x4400_0000	256	256	256
Peripheral Bridge B	0x4480_0000	256	256	256
Peripheral Bridge C	0x4500_0000	256	256	256
Independent Peripherals	0x4600_0000	9	9	1

11.2.1. Flash Memory Parameters

A single page contains 4K Bytes which is applicable to all the device part numbers listed in the “Configuration Summary”.

Number of pages available in a device part number will vary based on available maximum Flash memory size.

Equation 11-1.

$$NumberofPages = \frac{FlashSize(Bytes)}{4Kbytes}$$

11.3. SRAM Memory Configuration

Retention

Depending on the application and power budget needs, system memory can be retained in Standby or Hibernate sleep modes. The amount of the SRAM retained in this mode is software selectable, by writing the RAMCFG bits in the Power Manager Standby Configuration register and Hibernate Configuration register respectively (PM.STDBYCFG.RAMCFG and PM.HIBCFG.RAMCFG).

11.4. PFM Error Correction

For safety applications, the PIC32CK SG/GC family of devices embed error correction codes (ECC) to detect and correct single bit errors, or to enable dual-error detection for the system memory. The ECC is software selectable through the ECCDIS bit in the NVM User Configuration. For additional information, refer to the "NVM User Page Mapping - Dedicated Entries".

11.5. Configuration Flash Memory (CFM)

There are eight dedicated pages of CFM Flash used for various configuration control. There is a separate set of 8 pages in each Flash panel. The following table provides the CFM pages and their system usage.

Table 11-2. Configuration Flash Memory (CFM) Address Map

Start Address	End Address	Size	Flash Panel	Contents
0x0A00_0000	0x0A00_0FFF	4 KB	1	User CFG-1
0x0A00_1000	0x0A00_1FFF	4 KB	1	User OTP-1
0x0A00_2000	0x0A00_2FFF	4 KB	1	BOOT CFG-1
0x0A00_3000	0x0A00_3FFF	4 KB	1	RSVD
0x0A00_4000	0x0A00_4FFF	4 KB	1	RSVD
0x0A00_5000	0x0A00_5FFF	4 KB	1	RSVD
0x0A00_6000	0x0A00_6FFF	4 KB	1	RSVD
0x0A00_7000	0x0A00_7FFF	4 KB	1	CAL-OTP
0x0A00_8000	0x0A00_8FFF	4 KB	2	User CFG-2
0x0A00_9000	0x0A00_9FFF	4 KB	2	User OTP-2
0x0A00_A000	0x0A00_AFFF	4 KB	2	BOOT CFG-2
0x0A00_B000	0x0A00_BFFF	4 KB	2	RSVD
0x0A00_C000	0x0A00_CFFF	4 KB	2	RSVD
0x0A00_D000	0x0A00_DFFF	4 KB	2	RSVD
0x0A00_E000	0x0A00_EFFF	4 KB	2	RSVD
0x0A00_F000	0x0A00_FFFF	4 KB	2	RSVD

Note: CAL-OTP is locked at manufacturing and cannot be changed.

Flash Panel One contains the following configuration pages:

- **USER CFG (UCFG1)**: User configuration. Contains user specific configuration. such as Panel Sequence Number, WDT configuration, Clock configuration, BOR configuration and so on.
- **USER OTP - (UOTP1)**: User OTP. One-time-programmable User OTP configuration. Contains user serial number, User defined keys, custom analog calibration values, and so on.
- **BOOT CFG (BTCFG1)**: Boot configuration. Contains Chip Erase keys, Boot Options, Secure boot configuration, and so on.
- **CAL-OTP (CCFG)**: Calibration mainly used by analog macros and device variant information. Calibration data written by Factory Test Program. This page is programmed and locked at manufacture and cannot be changed.
- **RSVD**: Reserved page for future use or features available under NDA.

Flash Panel Two contains the following configuration pages:

- **USER CFG2 (UCFG2)**: Second user configuration. Format same as Panel 1 USER CFG
- **USER OTP2 (UOTP2)**: Second user OTP. Format same as Panel 1 USER OPT
- **BOOT CFG2 (BTCFG2)**: Second boot configuration. Format same as Panel 1 BOOT CFG
- **RSVD**: Reserved page for future use or features available under NDA

Note: The USER OTPn Flash rows are write protectable by user software through the FCW.UOWP register. The user OTP Flash cannot be erased using a page or chip erase. The user OTP Flash is intended for the storage of the user system calibration data that must survive any Flash erase.

11.5.1. User Configuration

The User Configuration values exist in the USER CFG page of CFM Flash Memory. Because this Flash memory uses ECC the Sequence Number the User Configuration must exist in separate Flash words. This is because they are often not written during the same programming sequence but ECC requires a 256-bit Flash word to be written once only. The user configuration register names correspond to Flash addresses as follows:

Table 11-3. User Configuration Map

Address Offset	Name	User	Description
0x0000	FSEQ0	BootROM	Sequence Number
0x0004	FSEQ1	BootROM	RSVD -Seq Num to determine boot Flash panel
0x0008	FSEQ2	BootROM	RSVD -Seq Num to determine boot Flash panel
0x000C	FSEQ3	BootROM	RSVD -Seq Num to determine boot Flash panel
0x0010	FSEQ4	BootROM	RSVD -Seq Num to determine boot Flash panel
0x0014	FSEQ5	BootROM	RSVD -Seq Num to determine boot Flash panel
0x0018	FSEQ6	BootROM	RSVD -Seq Num to determine boot Flash panel
0x001C	FSEQ7	BootROM	RSVD -Seq Num to determine boot Flash panel
0x0020	AFSEQ0	BootROM	Reserved for alternate sequence number
0x0024	AFSEQ1	BootROM	Reserved for a second sequence number
0x0028	AFSEQ2	BootROM	Reserved for a second sequence number
0x002C	AFSEQ3	BootROM	Reserved for a second sequence number
0x0030	AFSEQ4	BootROM	Reserved for a second sequence number
0x0034	AFSEQ5	BootROM	Reserved for a second sequence number
0x0038	AFSEQ6	BootROM	Reserved for a second sequence number
0x003C	AFSEQ7	BootROM	Reserved for a second sequence number
0x0040	FUCFG0	BootROM=WDT	WDT_CFG[31:0] (matches WDT SFR0-3)
0x0044	FUCFG1	Reserved	Reserved
0x0048	FUCFG2	BootROM=SUPC	DS Supply Monitor Overseer and Reference Configuration
0x004C	FUCFG3	BootROM=SUPC	Supply Monitor Overseer and Reference BOR Configuration
0x0050	FUCFG4	BootROM=SUPC	Supply Monitor Overseer and Reference CKT Configuration
0x0054	FUCFG5	Reserved	Reserved
0x0058	FUCFG6	(FCR)BootROM=FCR.ECCCTL	NOTE : Matches FCR SFR ECCCTL bit order.
0x005C	FUCFG7	(FCR)BootROM=FCR.CTRLB	NOTE : Matches FCR SFR CTRLB bit order.
0x0060	FUCFG8	(FCW)BootROM=FCW.CWP	NOTE : Matches FCW.CWP SFR bit order.
0x0064	FUCFG9	BOOTROM Control	BISR_RST_EN
0x0068	FUCFG10	BOOTROM	Peripheral Bus A security attribution
0x006C	FUCFG11	BOOTROM	Peripheral Bus B security attribution
0x0070	FUCFG12	BOOTROM	Peripheral Bus C security attribution
0x0074	FUCFG13	BOOTROM	AHB Peripheral security attribution
0x0078	FUCFG14	BOOTROM	PFM IDAU Configuration
0x007C	FUCFG15	BOOTROM	RAM IDAU Configuration
0x0080-0x00BC	FUCFG16-FUCFG31	RSVD	
0x00FC	RSVD	RSVD	For FACTORY Undefined UCFG
0x0100	RSVD	User SW Internal Use	Reserved for USER Defined CFG
....	RSVD	RSVD	Reserved for USER Defined CFG
0x0FFC	RSVD	N/A	Reserved for USER Defined CFG

11.5.1.1. FSEQ0

Name: FSEQ0

Offset: 0x0000

Value after chip erase: 0xFFFF_0001

Bits 31:16 SEQBAR[15:0] – Ones complement of sequence number

Bits 15:0 SEQNUM[15:0] – Sequence number

11.5.1.2. AFSEQ0

Name: AFSEQ0

Offset: 0x0020

Value after chip erase: 0XFFFF_FFFF

Bits 31:16 ASEQBAR[15:0] – Ones complement of alternate sequence number

Bits 15:0 ASEQNUM[15:0] – Alternate sequence number

11.5.1.3. FUCFG0 - User Configuration Register 0 – WDT Configuration

Name: FUCFG0

Offset: 0x0040, Value after chip erase: 0x0

Bits 23-16 EWOFFSET[3:0] : Configuration setting for WDT.EWCTRL.EWOFFSET[3:0]

Bits 15-12 WDT_WIN[3:0] : Configuration setting for WDT.CONFIG.WINDOW[3:0]

Bits 11-8 PER[3:0] : Configuration setting for WDT.CONFIG.PER[3:0]

Bit 7 ALWAYSON : Configuration setting for WDT.CTRLA.ALWAYSON

Bit 6 RUNSTDBY : Configuration setting for WDT.CTRLA.RUNSTDBY

Bit 2 WEN : Configuration setting for WDT.CTRLA.WEN

Bit 1 ENABLE : Configuration setting for WDT.CTRLA.ENABLE

11.5.1.4. FUCFG1 - User Configuration Register 1 – Reserved

Name: FUCFG1

Offset: 0x0044, Value after chip erase: 0XFFFF_FFFF

11.5.1.5. FUCFG2 - User Configuration Register 2 – SUPC DS Supply Monitor Overseer and Reference (SMOR) Configuration

Name: FUCFG2

Offset: 0x0048, Value after chip erase: 0x00000079

Bits 6-3 CKT[3:0] : Reserved

Bits 2:1 BOR_TRIP[1:0] : BOR Trip Value
Note: See DCBOR Electrical Specifications.

Bit 0 BOR_HYS : BOR Hysteresis
Note: See DCBOR Hysteresis Electrical Specifications.

11.5.1.6. FUCFG3 - User Configuration Register 3 – SUPC Supply Monitor Overseer and Reference (SMOR) BOR Configuration

Name: FUCFG3

Offset: 0x004C, Value after chip erase: 0x00000449

Bit 10 HYST_BOR_VDDREG

Value	Description
0	20mV
1	40mV

Bits 8-7 BOR_TRIP_VDDA[1:0]

Bit 6 HYST_BOR_VDDA: AVDD_BOR (Brown-out Reset), Hysteresis Select

BOR_TRIP_AVDD		HYST_BOR_AVDD	
Value	Description	Value	Description
0x00	1.67-1.71V	0	20.0 mV
		1	40.0 mV
0x01	2.19-2.22V	0	62.5 mV
		1	125 mV
0x02	2.63-2.67V	0	100 mV
		1	200 mV
0x03	2.96-2.96V	0	125 mV
		1	300 mV

Bits 2-1 BOR_TRIP_VDDIO[1:0]

Bit 0 HYST_BOR_VDDIO

BOR_TRIP_VDDIO		HYST_BOR_VDDIO	
Value	Description	Value	Description
0x00	1.67-1.71V	0	20.0 mV
		1	40.0 mV
0x01	2.19-2.22V	0	62.5 mV
		1	125 mV
0x02	2.63-2.67V	0	100 mV
		1	200 mV
0x03	2.96-2.96V	0	125 mV
		1	300 mV

Notes:
1. See BOR Electrical Specifications.
2. See BOR Hysteresis Electrical Specifications.
3. VDDA and AVDD are used interchangeably.

11.5.1.7. FUCFG4 - User Configuration Register 4 – Reserved

Name: FUCFG4

Offset: 0x0050, Value after chip erase: 0xFFFF_FFFF

Bits 7-0: Reserved

11.5.1.8. FUCFG5 - User Configuration Register 5 – Reserved

Name: FUCFG5

Offset: 0x0054

Value after chip erase: 0xFFFF_FFFF
Note: The user must always leave as 0xFFFFFFFF.

11.5.1.9. FUCFG6 - User Configuration Register 6 – FCR.ECCCTRL Configuration

Name: FUCFG6

Offset: 0x0058, Value after chip erase: 0x0000_0070

Bits 15-8 SECCNT[7:0] : Single Error Control Configuration Bits

Bit 6 ECCUNLCK: NVM ECC Mode Control Unlock configuration bits

Bits 5-4 ECCCTRL[1:0] : NVM ECC Mode Control configuration bits

11.5.1.10. FUCFG7 - User Configuration Register 7 – Reserved

Name: FUCFG7

Offset: 0x005C, Value after chip erase: 0xFFFF_FFFF

11.5.1.11. FUCFG8 - User Configuration Register 8 – FCW.CWP Configuration

Name: FUCFG8

Offset: 0x0060, Value after chip erase: 0x0000_0000

Bit 24 UC2WPLOCK: User Configuration Page 2 Write Protect Lock Bit

Bit 16 UC1WPLOCK: User Configuration Page 1 Write Protect Lock Bit

Bit 8 UC2WP: User Configuration Page 2 Write Protect Bit

Bit 0 UC1WP: Boot Configuration Page 1 Write Protect Bit

11.5.1.12. FUCFG9 - User Configuration Register 9 – RESERVED

Name: FUCFG9

Offset: 0x0064, Value after chip erase: 0xFFFF_FFFF

11.5.1.13. FUCFG10 - User Configuration Register 10 – Peripheral Bus A security attribution (BOOTROM_NONSECA)

Name: FUCFG10
Offset: 0x0068
Value after chip erase: 0x0000_0000
Bit 14: NONSECA_PAC
Bit 13: NONSECA_EIC
Bit 12: NONSECA_RTC
Bit 11: NONSECA_WDT
Bit 10: NONSECA_FREQM
Bit 9: NONSECA_MCLK
Bit 8: NONSECA_GCLK
Bit 7: NONSECA_OSC32KCTRL
Bit 6: NONSECA_OSCCTRL
Bit 5: NONSECA_RSTC
Bit 4: NONSECA_SUPC
Bit 3: NONSECA_PM

11.5.1.14. FUCFG11 - User Configuration Register 11 – Peripheral Bus B security attribution (BOOTROM_NONSECB)

Name: FUCFG11
Offset: 0x006C
Value after chip erase: 0x0000_0000
Bit 16: NONSECB_TCC3
Bit 15: NONSECB_TCC2
Bit 14: NONSECB_TCC1

Bit 13: NONSECB_TCC0
Bit 12: NONSECB_SERCOM3
Bit 11: NONSECB_SERCOM2
Bit 10: NONSECB_SERCOM1
Bit 9: NONSECB_SERCOM0
Bit 8: NONSECB_EVSYS
Bit 6: NONSECB_PRM
Bit 5: NONSECB_CMCC
Bit 4: NONSECB_HMATRIX
Bit 3: NONSECB_DMA1
Bit 2: NONSECB_DMA0
Bit 1: NONSECB_PORT

11.5.1.15. FUCFG12 - User Configuration Register 12 – Peripheral Bus C security attribution (BOOTROM_NONSECC)

Name: FUCFG12
Offset: 0x0070
Value after chip erase: 0x0000_0000
Bit 23: NONSECC_EBI
Bit 22: NONSECC_USB
Bit 21: NONSECC_SDMMC1
Bit 20: NONSECC_SDMMC0
Bit 19: NONSECC_TRNG
Bit 18: NONSECC_ETH
Bit 17: NONSECC_CAN1
Bit 16: NONSECC_CAN0
Bit 15: NONSECC_PDEC
Bit 14: NONSECC_CCL
Bit 13: NONSECC_PCC
Bit 12: NONSECC_SPI_IXS
Bit 11: NONSECC_PTC
Bit 10: NONSECC_AC
Bit 9: NONSECC_ADC
Bit 8: NONSECC_TCC7
Bit 7: NONSECC_TCC6
Bit 6: NONSECC_TCC5
Bit 5: NONSECC_TCC4
Bit 4: NONSECC_SERCOM7
Bit 3: NONSECC_SERCOM6
Bit 2: NONSECC_SERCOM5
Bit 1: NONSECC_SERCOM4

11.5.1.16. FUCFG13 - User Configuration Register 13 – AHB peripheral security attribution (BOOTROM_NONSECAHB)

Name: FUCFG13
Offset: 0x0074
Value after chip erase: 0x0000_0000
Bit 3: NONSEC_AHB_SQI
Bit 2: NONSEC_AHB_USBHS

Bit 1: NONSEC_AHB_HSM

11.5.1.17. FUCFG14 - User Configuration Register 14 – PFM IDAU Configuration

Name: FUCFG14

Offset: 0x0078

Value after chip erase: 0x0000_0000

Bit 16: IDAU_ANSC : Non-secure Flash application size

Bits 8-0: IDAU_ANS: Non-secure callable Flash application size

11.5.1.18. FUCFG15 - User Configuration Register 15 – RAM IDAU Configuration

Name: FUCFG15

Offset: 0x007C

Value after chip erase: 0x0000_0000

Bit 6-0: IDAU_RNS : Non-Secure Ram size

11.5.1.19. FUCFG16-31 - User Configuration Registers 16-31 – Reserved

Name: FUCFG16-31

Offset: 0x0080-0x00BC

Reserved

Note: FUCFG16 and FUCFG24 contain HSM_PTRMETA0 and HSM_PTRMETA1 respectively, and are relevant only on variants with HSM enabled.

11.5.2. User OTPn, n = 1,2

Each page in Flash is reserved for the user specific information, providing 4 Kbytes of emulated the user OTP Memory. The four Flash rows of each page are write protectable by the user software through the FCW.UOWP.UOnWP[3:1] bits. The user OTP Flash cannot be erased using a page or chip erase. The user OTP Flash is intended for the storage of users system calibration data that must survive any Flash erase.

11.5.3. Boot Configuration

The Boot configuration page contains security information used by the Boot ROM. Information such as the Boot ROM CRC-32 value will be stored within this reserved space. Both BootCFG1 and BootCFG2 have the same mapping, the table below provides BootCFG mapping details.

Table 11-4. Bootcfg Mapping

<table>
<tr><th>Address</th><th>Register</th><th>Bit Field</th><th>Start Bit</th><th>Size (bits)</th><th>Value after chip erase</th><th>Description</th></tr>
<tr><td rowspan="7">0A002000</td><td rowspan="7">BLDRCFG</td><td><See Below:></td><td></td><td></td><td>0xC0000000</td><td>Bootloader configurations</td></tr>
<tr><td>PFM_BCRP</td><td>0</td><td>1</td><td>0x0</td><td>Boot Config Page read protect bit</td></tr>
<tr><td>PFM_BCWP</td><td>1</td><td>1</td><td>0x0</td><td>Boot Config Page write protect bit</td></tr>
<tr><td>BROM_EN_BFMCHK</td><td>28</td><td>1</td><td>0x0</td><td>BFM CRC check enable</td></tr>
<tr><td>BROM_EN_PLL</td><td>29</td><td>1</td><td>0x0</td><td>PLL config enable for BFM processings</td></tr>
<tr><td>BROM_EN_DALUN</td><td>30</td><td>1</td><td>0x1</td><td>Allow BootRom to apply DALUN</td></tr>
<tr><td>BROM_EN_ RWLOCKS</td><td>31</td><td>1</td><td>0x1</td><td>Allow BootRom to apply Read & Write locks</td></tr>
<tr><td rowspan="3">0A002004</td><td rowspan="3">BROM_BSEQ</td><td><See Below:></td><td></td><td></td><td>0xFFFE0001</td><td>Sequence # for BootCfg page</td></tr>
<tr><td>SEQNUM</td><td>0</td><td>16</td><td>0x0001</td><td>Boot ROM Sequence Number</td></tr>
<tr><td>SEQBAR</td><td>16</td><td>16</td><td>0xFFFE</td><td>1’s Complement of Seq. #</td></tr>
</table>

Table 11-4. Bootcfg Mapping (continued)

Address	Register	Bit Field	Start Bit	Size (bits)	Value after chip erase	Description
0A002008	BFM_CHK_ TABLEPTR		0	32	0xFFFFFFFF	BFM CRC table pointer
0A002020 - 0A00207F	Reserved					
0A002080	KEYVAL_ TZ0_CE_ALL		0	128	0xFFFF...FF	Chip erase 3 (TZ0_ALL) access key
0A002090	CELOCK_ TZ0_CE_ALL		0	32	0xFFFFFFF	Chip erase lock for TZ0_ALL
0A002094	Reserved		0	96		
0A0020A0	KEYVAL_ CRCCMD		0	128	0xFFFF...FF	CRC access key
0A0020B0	KEYCONFIG_ CRCCMD		0	128	0xFFFF...FF	Config for CRC access key
0A0020C0	KEYVAL_ HOSTDALELEV		0	128	0xFFFF...FF	Host-authenticated DAL elevation key
0A0020D0	KEYCONFIG_ HOSTDALELEV		0	128	0xFFFF...FF	Config for Host-authenticated DAL elevation key
0A0020E0	ROM_CTRLA		0	32	0xFFFFFFFF	ROM.CTRLA setting (if BROM_PLL_EN=1)
0A0020E4	FCR_CTRLA		0	32	0xFFFFFFFF	FCR.CTRLA setting (if BROM_PLL_EN=1)
0A0020E8	RPMU_ VREGCTRL		0	32	0xFFFFFFFF	RPMU.VREGCTRL setting (if BROM_PLL_EN=1)
0A0020EC	PLL0_CTRL		0	32	0xFFFFFFFF	PLL0.CTRL setting (if BROM_PLL_EN=1)
0A0020F0	PLL0_FBDIV		0	32	0xFFFFFFFF	PLL0.FBDIV setting (if BROM_PLL_EN=1)
0A0020F4	PLL0_REFDIV		0	32	0xFFFFFFFF	PLL0.REFDIV setting (if BROM_PLL_EN=1)
0A0020F8	PLL0_ POSTDIVA		0	32	0xFFFFFFFF	PLL0.POSTDIVA setting (if BROM_PLL_EN=1)
0A0020FC	MCLK_ CLKDIV1		0	32	0xFFFFFFFF	MCLK.CLKDIV1 setting (if BROM_PLL_EN=1)
0A002100	GCLK_ GENCTRL0		0	32	0xFFFFFFFF	GCLK.GENCTRL0 setting (if BROM_PLL_EN=1)
0A002104	Reserved		0	96		
0A002110	BROM_ BOOTCFGCRC		0	128	0xFFFFFFFF	CRC table for bootcfg page
0A002120	BROM_ PAGEEND		0	0	0xFFFFFFFF	Marker for end of verified data

Notes:
- PFM BootCfg read and write protects are disabled (BCWP = 0, BCRP = 0)
- BROM_EN_BFMCHK = 0, BFM CRC check disabled
- BROM_EN_PLL = 0, PLL config disabled
- BROM_EN_DALUN is set (DAL setting will be applied at Boot ROM exit)
- BROM_EN_RWLOCKS is set (Locks will be applied at Boot ROM exit)

11.6. BFM Integrity Check

The Boot ROM can optionally perform an integrity check of the Boot Flash Memory (BFM). If BROM_EN_BFMCHK = 1, the Boot ROM calculates a CRC over the selected portion of the current BFM and compares the calculated value to the expected value in a CRC table. If the CRC values do not match, the Boot ROM enters interactive mode. The BFM_CHK_TABLEPTR word specifies the location of the CRC table. Typically, the CRC table is stored in BFM. The default setting is BROM_EN_BFMCHK = 0.

When the CRC check fails for any reason (incorrect CRC table, ECC error, or CRC mismatch), the STATUS_ERR_BFMCRC is placed on the Boot communication channel for use by a debugger and interactive mode is called.

Table 11-5. CRC Table Format

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

Notes:
1. The start address must be a multiple of 4 (Only ADDR[31:2] are used).
2. The size must be a multiple of 4 (Only SIZE[31:2] are used).
3. The expected value is the computed CRC32 value of the memory target.

11.7. CAL OTP Configuration

The following values are programmed at the Factory and cannot be changed.

Table 11-6. CAL OTP Map

Address	Register	Module	Bitfield
Base + 0x080	FCCFG0	SUPC	CFG_SMOR0[31:0]
Base + 0x080	FCCFG1	SUPC	CFG_SMOR0[31:0]
Base + 0x0A0	FCCFG8	SUPC/OSC32KCTRL	[15:0] - CFG_DSSMOR_REF[15:0] [31:16] - CFG_DSSMOR_CLK[15:0]
Base + 0x0C0	FCCFG16	OSCCTRL	CFG_DFLL48M[31:0]
Base + 0x0C4	FCCFG17	OSCCTRL	[15:0] - CFG_XTAL[15:0] [31:16] - RSVD
Base + 0x0E0	FCCFG24	SUPC	[15:0] – CFG CALVREGSW0[15:0] (99ma) [31:16] – CFG CALVREGSW1[15:0] (99ma)
Base + 0x0E8	FCCFG26	SUPC	[15:0] – CFG CALADDVREG0[15:0] (50 ma) [31:16] – RSVD
Base + 0x0EC	FCCFG27	SUPC	[15:0] – RSVD [31:16] - CFG_CALVREGRAM[15:0] (50 ma)

Table 11-6. CAL OTP Map (continued)

Address	Register	Module	Bitfield
Base + 0x0F0	FCCFG28	SUPC	[3:0] - CALCP[3:0] [7:4] - RSVD [15:8] - CALSUPC[7:0] [31:16] - RSVD
Base + 0x0F4	FCCFG29	SUPC	[7:0] - CALUSERLDO[7:0] [31:8] - RSVD
Base + 0x100	FCCFG32	DSU[DID] DSU[DCFG0]	[7:0] - RSVD [3:0] , PFM_SZ[3:0] , [15:8] - DRM_SZ [7:0] [23:16] - RSVD[3:0] ,FPACKAGE[3:0] [31:24] - DID_DEVSEL[7:0]
Base + 0x104	FCCFG33	DSU[DCFG1]	[7:0] – ME_HSM, DPA, RSVD, ME_SPI_IXS, RSVD, ME_SQI, ME_SDMMC1, SDMMC0 [15:8] – ME_TCC7, ME_TCC6, ME_TCC5, ME_TCC4, ME_TCC3, ME_TCC2, ME_TCC1, ME_TCC0 [23:16] – ME_SERCOM5, ME_SERCOM4, ME_SER-COM3, ME_SERCOM2, ME_SERCOM1, ME_SER-COM0, RSVD[1:0] [31 :24] – ME_EBI, ME_PCC, ME_ETH, ME_PTC, RSVD[1:0], ME_SERCOM7, ME_SERCOM6
Base + 0x108	FCCFG34	DSU[DCFG2]	[7:0] – RSVD[5:0], ME_CAN1, ME_CAN0 [15:8] - ME_DMA0CH[5:0], ME_USBFS, ME_USBHS0 [23:16] - ME_DMA1CH[5:0], RSVD, ME_PDEC [31:24] – RSVD (for ME_ACPAIR_SEL[7:0])
Base + 0x144	FCCFG49	MBISTINTF	MEM_MARGIN0[31:0]
Base + 0x148	FCCFG50	MBISTINTF	MEM_MARGIN1[31:0]
Base + 0x14C	FCCFG51	MBISTINTF	MEM_MARGIN2[31:0]
Base + 0x160	FCCFG56	BOOTROM -> RPMU [BRRPMU_CAL]	BRRPMU_CAL[31:0]
Base + 0x164	FCCFG57	BOOTROM -> RPMU [BRCFGPWSAKDLY]	BRCFGPWSAKDLY [31:0]
Base + 0x168	FCCFG58	BOOTROM -> RPMU [BRCFGRINGOSC]	BRCFGRINGOSC [31:0]
Base + 0x16C	FCCFG59	BOOTROM	Change IDAU settings when TrustZone is disabled
Base + 0x180	FCCFG64	User SW => PTC	[15:0] - PTCCFG0[15:0] [31:16] - RSVD[15:0]
Base + 0x184	FCCFG65	User SW => ADC	ADCCFG0[31:0]
Base + 0x188	FCCFG66	User SW => AC (CTRLC)	[7:0] – RSVD [7:4], CONFIG[3:0] [31:8] - RSVD
Base + 0x18C	FCCFG67	User SW => USBHS Squelch	[3:0] – CFG_USB_PHY[3:0] [31:4] - RSVD

Table 11-6. CAL OTP Map (continued)

Address	Register	Module	Bitfield
Base + 0x190	FCCFG68	User SW => USBFS.PAD-CAL	[4:0]: TRANSP[4:0] [5]: RSVD [10:6]: TRANSN[4:0] [11]: RSVD [14:12] TRIM[2:0] [31:15]: RSVD
Base + 0x1A0	FCCFG72	User SW => ETH	FMAC[31:0]
Base + 0x1A4	FCCFG73	User SW => ETH	[15:0] - FMAC[47:32] [31:16] - RSVD[15:0]

11.8. Unique ID (UID)

The PIC32CK SG/GC provides a 128 bit (UNIQIDW*n*, *n* = 0,1,2,3) which conforms to the version 4 UUID format. See the following table for the location in Flash memory. The unique ID is programmed onto the device at manufacture.

For more information on the standard, refer to the RFC4122 at tools.ietf.org/html/rfc4122.

The version 4 UUID is a 128-bit value which includes 122-bit random value + 2-bit variant code + 4-bit version code. The random value can be generated using the TRNG on the device or with external equipment during test. This form of UID is referred to as a statistically guaranteed unique value because the possibility of a collision is infinitesimally small. It has the advantage that UID values are unique across vendors without the need for a central authority to manage the number space.

Table 11-7. Unique ID Map

Address Offset Base=CAL OTP	Name	User	Description
Base + 0x01E0	UNIQIDW0	User SW	UID[31:0] - Unique ID Word 0
Base + 0x01E4	UNIQIDW1	User SW	UID[63:32] - Unique ID Word 1
Base + 0x01E8	UNIQIDW2	User SW	UID[95:64] - Unique ID Word 2
Base + 0x01EC	UNIQIDW3	User SW	UID[127:96] - Unique ID Word 3

12. Hardware Security Module (HSM)

12.1. Features

The following are key features of the HSM module:

- User programmable Arm M0+CPU
- Hardware security partition with 128 kB local RAM
- Standard firmware for turnkey operation
- High performance cryptographic accelerators
- AES, TDES, ChaCha20, SHA-2, SHA-1, Poly1305, RSA, ECC
- Secure non-volatile key storage
- Secure boot
- Secure debug
- True Random Number Generator(TRNG)
- Real Time Clock (RTC)
- Tamper response module
- Backup registers
- AHB DMA Interface
- AHB Host Interface

12.2. Performance

Table 12-1. HSM Performance (3)

Algorithm	Speed	Units
Authenticated Encryption with Associated Data Algorithms		
AES-GCM	130	Mbps
Digital Signature		
ECDSA P-256 Sign	3.5	ms
ECDSA P-256 Verify	3.8	ms
ECDSA P-384 Sign	9.0	ms
ECDSA P-384 Verify	10.5	ms
Hash Algorithms		
SHA1	600	Mbps
SHA2-224	710	Mbps
SHA2-256	710	Mbps
SHA2-384	890	Mbps
SHA2-512	890	Mbps
Key Agreement Scheme (KAS) Algorithms		
KAS SECP256R1	3.7	ms
KAS SECP384R1	9.1	ms
Random Number Generator (RNG) Algorithms		
TRNG (1)	50	Mbps
Symmetric Algorithms (2)		
AES 128	670	Mbps
AES 192	620	Mbps

Table 12-1. HSM Performance [3] (continued)

Algorithm	Speed	Units
AES 256	570	Mbps

Notes:
1. 100 KB output.
2. ECB, CBC and CTR modes.
3. Main clock and HSM clock are configured for 120MHz.

Note: Contact a local Microchip Sales Office for more information on this module available under a non disclosure agreement.

13. Multi-Channel RAM Controller (MCRAMC)

13.1. Overview

The MCRAMC provides memory controller functions as well as ECC and fault injection for the System RAM (DRM).

The memory controller provides up to 8 AHB ports which provide access the system RAM. Each port is guaranteed 100% bandwidth to the entire system RAM. However, there may be a latency penalties if more than one port attempts to access the same memory bank. The CPU master has exclusive access to AHB port 0 which is configured to a higher priority than other AHB ports. This configuration provides the CPU with the lowest overall latency to the system memory.

The memory controller also provides ECC and fault injection. The user can optional enable/disable ECC correction of read data. ECC values are always calculated and stored for writes. Both SEC and DED errors are logged and each event can generate an interrupt. On a SEC, the controller returns corrected data. On a DED, the controller generates a slave bus error. The MCRAMC implements up to four memory banks or channels. RAM must be distributed evenly across all channels.

Note: This module can only be configured by PIC32CK1012 devices.

13.2. Features

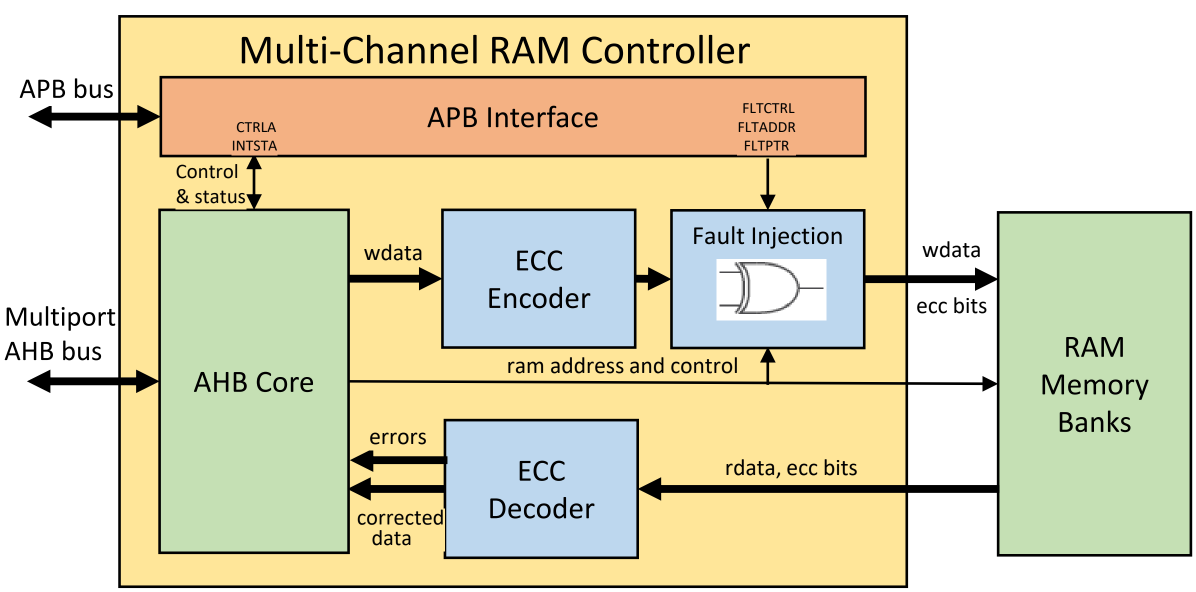
Embedded Characteristics:

- ECC Single Error Correction on the fly
- ECC correction automatic write-back to RAM
- ECC synchronous abort exception for recoverable Double Error capability
- ECC test with fault injection
- ECC error status report and interrupt signaling

13.3. Block Diagram

The following figure illustrates the block diagram of the MCRAMC.

Figure 13-1. MCRAMC Block Diagram



13.4. Functional Description

13.4.1. ECC SECDED - ECC Functionality

The ECC decoding is enabled or disabled with the CTRLA.ENABLE bit. After reset the Rom code will copy the BOOTCFG1.BOOT_FLAG.RAM_INIT_ENB fuse in the CTRLA.ENABLE bit.

The ECC feature can then be disabled and re-enabled using the same bit (for testing the ECC feature, for example, as detailed in the “ECC testing” chapter).

This disabling or enabling only applies to ECC decoding upon reads. The ECC encoding upon writes is always enabled and cannot be disabled.

Each 32-bit data word in SRAM is completed by 7 additional ECC bits, which are not accessible by the user.

Upon any 8-bit, 16-bit, or 32-bit write in the memory, the 7 ECC bits are computed and stored along with the data (The 8-bit or 16-bit writes are actually composed of an atomic read 32 bits, modify, write 32 bits).

Upon any 8-bit, 16-bit, or 32-bit read in the memory, if the ECC feature is disabled, then single or double errors are not detected, and not corrected. If the ECC feature is enabled, the ECC syndrome is computed on the related 32 data bits and 7 ECC bits.

Single-bit error read correction is performed on the fly with no penalty.

The MCRAMC writes back any corrected data into the RAM.

Simply reading the sensitive RAM content on a regular basis, for example from a DMA Host, prevents bit error accumulation. This is known as ECC memory scrubbing.

Double-bit error read detection triggers a bus error response from the MCRAMC, typically leading to a synchronous abort exception at the bus Host. This enables stopping of the bus Host access sequence precisely at the faulty address.

Upon a bus error response, the faulty address is registered in Error Capture Address Register.

If the corrupted RAM word laying at the 32-bit aligned address was originally provided from a non volatile memory, that is, CPU instruction, it can be safely copied again into the RAM before returning from the abort exception handler. If the abort exception is immediately re-entered, there are ‘stuck-at’ bits in the corrupted memory word, which requires applying self-patching code technics prior returning from the abort exception handler.

Another example is if the corrupted RAM word was part of a DMA buffer from a communication peripheral, the communication might be retried if the DMA master has support for transfer error response.

13.4.2. SRAM Initialization

After a reset, the SRAM content (data and ECC) is random and the ECC feature may be enabled depending on the BOOTCFG1.BOOT_FLAG.RAM_INIT_ENB fuse value.

If the ECC feature is enabled, any 32-bit write will initialize the data and the related ECC bits. However, 8-bit or 16-bit writes (which imply an internal read32/modify/write32) will probably trigger a single or double error on the internal 32-bit read (depending on the randomness of the 39 bits in memory). Consequently, the SRAM content must be initialized before it can be used.

The simplest option is to program a basic FOR loop filling the whole memory or to program a DMA transfer. The written data can take any value. However, care must be taken to only perform 32-bit writes in SRAM to access variables or DMA descriptors, and to not overwrite these data during the memory fill. The ECC bits will then be computed for each write and the memory will then be available for normal use.

13.4.3. ECC Testing

For ECC testing purpose, single-bit or double-bit faults can be injected during writes at a specific address. The address needs to be programmed in the MCRAMC Fault Injection Address Register (MCRAMC.FLTADDR), the bits to be flipped need to be programmed in the MCRAMC Fault Injection Pointer Register (MCRAMC.FLTPTR) and the fault injection needs to be enabled in the MCRAMC Fault Injection Control Register (MCRAMC.FLTCTRL).

Then the MCRAMC FLTCTRL Register has to be read prior to the first write access to the RAM.

After fault injections, if ECC decoding is disabled in the MCRAMC Control Enable A Register, reading at a faulty address directly shows the faulty data bits, if any. Then ECC decoding can be enabled again and fault injection disabled, to read and check single error correction or double error detection. Then ECC decoding can be disabled again to read and check if a single error has been automatically corrected in the RAM array.

When ECC decoding is enabled in the MCRAMC Control Enable A Register, single-bit and double-bit ECC errors can be flagged in the MCRAMC INTSTA Interrupt Status register and can trigger an interrupt if enabled in the MCRAMC.INTENx Interrupt Enable register.

Note: This interrupt is an asynchronous CPU exception. It generally comes too late for system safe state recovery in case of a double-bit error.

The characteristics of a single-bit error are captured into the MCRAMC Error Capture registers and held stable until either the MCRAMC INTSTA.SER status bit is cleared by the software or a Double-bit error occurs.

The characteristics of a double-bit error are captured into the MCRAMC Error Capture registers and held stable until the MCRAMC INTSTA.DER status bit is cleared by the software.

The following constraints must be observed during the ECC testing process:

- After one or more writes to the MCRAMC user interface to change its configuration, a single read needs to be done from the MCRAMC FLTCTRL Register, prior to performing any access to the RAM.
- When both ECC decoding and fault injection are enabled, no single-bit fault RAM word must be read at the RAM fault injection address, because memory correction write-back would inject a fault again.
- When both ECC decoding and fault injection are enabled, a double-bit fault RAM word at the RAM fault injection address must be overwritten only with a 32-bit wide access.
- When fault injection is enabled, the data bits to be flipped, as programmed in the MCRAMC Fault Injection Pointer Register, must always be part of the bytes modified by the write access to the RAM fault injection address. A simple way to ensure this is to restrict the write accesses to the RAM fault injection address to be 32-bit wide only.

Note: A double error condition will generate a hard fault which will supersede the double error interrupt. If identification and soft recovery is desired, this condition must be detected in the hard fault handler, allowing the function to exit and the ISR for the double error to be executed.

13.4.4. Throughput Optimization

The throughput of the MCRAMC can be optimized when the system bus clock frequency is in lower range by removing any wait-state in the access to the RAM.

At higher frequency a wait-state must be inserted at the start of a burst.

When the MCRAMC_CTRLA.WS = 1, a wait state is inserted.

Refer to the *Electrical Characteristics* chapter for the configuration of WS.

13.4.5. Privileged Access

When the MCRAMC_CTRLA.PRIV = 1, the user interface access is granted only if the read or write access is performed only if the privileged attribute is set.

13.5. Register Summary

For descriptions and definitions of both register and bitfield properties, refer to the *Register Properties.*

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRLA	31:24								
		23:16								
		15:8								
		7:0							ENABLE	SWRST
0x04	SYNCBUSY	31:24								
		23:16								
		15:8								
		7:0						ENABLE	FLTEN	
0x08	INTENCLR	31:24								
		23:16								
		15:8								
		7:0							DERREN	SERREN
0x0C	INTENSET	31:24								
		23:16								
		15:8								
		7:0							DERREN	SERREN
0x10	INTSTA	31:24								
		23:16								
		15:8								
		7:0							DERR	SERR
0x14	FLTCTRL	31:24								
		23:16								
		15:8			FLTMD[1:0]					
		7:0							FLTEN	
0x18	FLTPTR	31:24								
		23:16	FLT2PTR[7:0]							
		15:8								
		7:0	FLT1PTR[7:0]							
0x1C	FLTADR	31:24								
		23:16	FLTADR[23:16]							
		15:8	FLTADR[15:8]							
		7:0	FLTADR[7:0]							
0x20	ERRCADR	31:24								
		23:16	ERCADR[23:16]							
		15:8	ERCADR[15:8]							
		7:0	ERCADR[7:0]							
0x24	ERRCPAR	31:24								
		23:16								
		15:8								
		7:0	ERCPAR[7:0]							
0x28	ERRCSYN	31:24								
		23:16								
		15:8	ERR2	ERR1						
		7:0	ERCSYN[7:0]							

13.5.1. Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00000002
Property: PAC Write-Protection

Note: When protected by PAC, any write attempt to this register will fail and return a bus error.

Table 13-1. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

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

Bit 1 – ENABLE ECC Decoder Enable

Value	Description
0	ECC decoding is disabled.
1	ECC decoding is enabled.

Bit 0 – SWRST Software Reset

Value	Description
0	No effect.
1	Reset the MCRAMC. A software-triggered hardware reset of the MCRAMC user interface is performed.

13.5.2. Synchronization Busy Register

Name: SYNCBUSY
Offset: 0x04
Reset: 0x00000000
Property: Read-Only

Table 13-2. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						ENABLE	FLTEN	
Access						R/HS/HC	R/HS/HC	
Reset						0	0	

Bit 2 – ENABLE ECC Decoder Enable Busy Bit

Bit 1 – FLTEN Fault Injection Enabled Busy Bit

13.5.3. Interrupt Enable Clear Register

Name: INTENCLR
Offset: 0x08
Reset: 0x00000000
Property: PAC Write-Protected

Note: When protected by PAC, any write attempt to this register will fail and return a bus error.

Table 13-3. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

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

Bit 1 – DERREN Double Bit Error Interrupt Enable Clear
A read returns the Interrupt Enable bit DERREN for the Interrupt Status bit DERR.
Writing a ‘0’ has no effect.
Writing a ‘1’ clears the Interrupt Enable bit DERREN.

Bit 0 – SERREN Single Bit Error Interrupt Enable Clear
A read returns the Interrupt Enable bit SERREN for the Interrupt Status bit SERR.
Writing a ‘0’ has no effect.
Writing a ‘1’ clears the Interrupt Enable bit SERREN.

13.5.4. Interrupt Enable Set Register

Name: INTENSET
Offset: 0x0C
Reset: 0x00000000
Property: PAC Write-Protection

Note: When protected by PAC, any write attempt to this register will fail and return a bus error.

Table 13-4. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

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

Bit 1 – DERREN Double Bit Error Interrupt Enable Set
A read returns the Interrupt Enable bit DERREN for the Interrupt Status bit DERR.
Writing a ‘0’ has no effect.
Writing a ‘1’ sets the Interrupt Enable bit DERREN.

Bit 0 – SERREN Single Bit Error Interrupt Enable Set
A read returns the Interrupt Enable bit SERREN for the Interrupt Status bit SERR.
Writing a ‘0’ has no effect.
Writing a ‘1’ sets the Interrupt Enable bit SERREN.

13.5.5. Interrupt Status Register

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

Note: Subsequent to an interrupt flag being cleared, the flag must be read back to verify the clear before exiting the ISR. Failure to do this can result in duplicate interrupts.

Table 13-5. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

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

Bit 1 – DERR Double Bit Error
Reading a '0' means no double bit error has occurred since the last clearing of this bit.
Reading a '1' means at least one double bit error has occurred since the last clearing of this bit.
Writing a '0' has no effect.
Writing a '1' clears this Interrupt Status bit.

Bit 0 – SERR Single Bit Error
Reading a '0' means no single bit error has occurred since the last clearing of this bit.
Reading a '1' means at least one single bit error has occurred since the last clearing of this bit.
Writing a '0' has no effect.
Writing a '1' clears this Interrupt Status bit.

13.5.6. Fault Injection Control Register

Name: FLTCTRL
Offset: 0x14
Reset: 0x00000000
Property: PAC Write-Protection

Note: When protected by PAC, any write attempt to this register will fail and return a bus error.

Table 13-6. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
			FLTMD[1:0]					
Access			R/W	R/W				
Reset			0	0				

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

Bits 13:12 – FLTMD[1:0] Fault Injection Mode
Note: When FLTEN has previously been written to 1, any write attempt to this field will fail and return a bus error.

Value	Name	Description
00	DISABLE	Fault Injection Disabled.
01	SINGLE	Single Fault Injection at bit selected by MCRAMC_FLTPTR.FLT1PTR.
10	DOUBLE	Double Fault Injection at the MCRAMC_FLTPTR.FLT1PTR and MCRAMC_FLTPTR.FLT2PTR bits.
11	RESERVED	Reserved.

Bit 1 – FLTEN Fault Injection Enabled

Value	Description
0	Disables fault injection.
1	Enables fault injection at FLTADR address offset as selected by FLTMD and FLTxPTR.

13.5.7. Fault Injection Pointer Register

Name: FLTPTR
Offset: 0x18
Reset: 0x00000000
Property: PAC Write-Protection

Note: When protected by PAC or when MCRAMC_FLTCTRL.FLTEN = 1, any write attempt to this register will fail and return a bus error.

Table 13-7. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	FLT2PTR[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
	FLT1PTR[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 – FLT2PTR[7:0] Double Fault Injection Bit Pointer
Index of the data bit to be flipped during RAM write access at MCRAMC address offset FLTADR for double bit error. Valid values range from 0 to 38.

Bits 7:0 – FLT1PTR[7:0] Single Fault Injection Bit Pointer
Index of the data bit to be flipped during RAM write access at MCRAMC address offset FLTADR for single and double bit error.
Valid values range from 0 to 38.

13.5.8. Fault Injection Address Register

Name: FLTADR
Offset: 0x1C
Reset: 0x00000000
Property: PAC Write-Protection

Note: When protected by PAC or when MCRAMC_FLTCTRL.FLTEN = 1, any write attempt to this register will fail and return a bus error.

Table 13-8. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	FLTADR[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
	FLTADR[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
	FLTADR[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 – FLTADR[23:0] Fault Address Offset
MCRAMC address offset of the RAM data word where the fault injection will occur when written at. Valid values range from 0 to 0x1FFFF8.
The MCRAMC system bus base address should be added to this offset to know the corresponding system bus address to be corrupted.

13.5.9. Error Capture Address Register

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

Table 13-9. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	ERCADR[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
	ERCADR[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
	ERCADR[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 23:0 – ERCADR[23:0] ECC SECDED Error Capture Address

MCRAMC address offset whose reading caused the ECC Error as reported in the Error Capture Syndrome register. The MCRAMC system bus base address should be added to this offset to know the corresponding system bus address.

13.5.10. Error Capture Parity Register

Name: ERRCPAR
Offset: 0x24
Reset: 0x00000000
Property: -

Table 13-10. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

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

Bits 7:0 – ERCPAR[7:0] ECC SECDED Error Capture Parity
The ECC decoder output Parity bits read at the ERCADR address offset from the MCRAMC system bus base address.

13.5.11. Error Capture Syndrome Register

Name: ERRCSYN
Offset: 0x28
Reset: 0x00000000
Property: -

Table 13-11. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
	ERR2	ERR1						
Access	R	R						
Reset	0	0						

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

Bit 15 – ERR2 ECC Double Bit Error

Value	Description
0	Not a Double bit error.
1	Double bit error.

Bit 14 – ERR1 ECC Single Bit Error

Value	Description
0	Not a Single bit error.
1	Single bit error.

Bits 7:0 – ERCSYN[7:0] ECC SECDED Error Capture Syndrome
ECC SECDED Syndrome bits read at the ERCADR address offset from the MCRAMC system bus base address.

14. Cortex-M Cache Controller (CMCC)

14.1. Overview

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

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

14.2. Features

The following are key features of the Cortex-M Cache Controller:

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

14.3. Block Diagram

Figure 14-1. CMCC Block Diagram

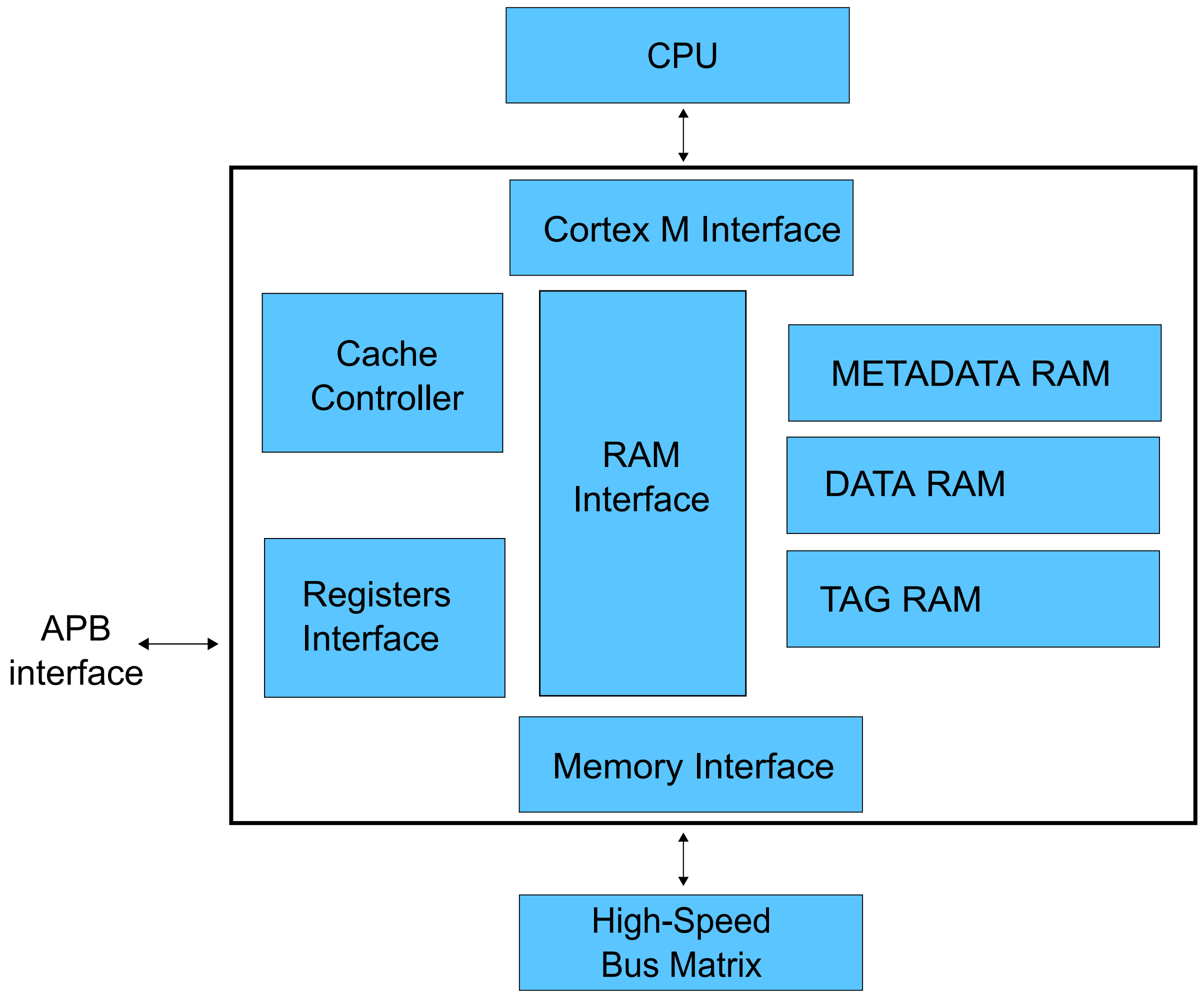
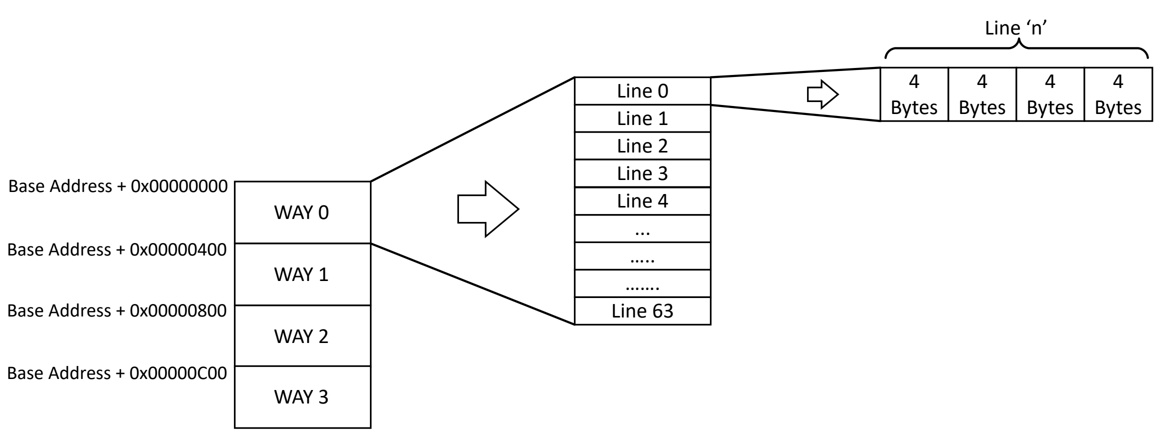


Figure 14-2. CMCC Organization



14.4. Peripheral Dependencies

Peripheral Name	Base Address	PAC Peripheral Identifier (PAC.WRCTRL.PERIDx)	Power Domain
CMCC	0x4480_8000	24	VDDREG

14.4.1. Power Management

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

14.4.2. Debug Operation

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

14.5. Functional Description

14.5.1. Principle of Operation

14.5.2. Initialization and Normal Operation

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

Follow these steps to enable the cache controller:

1. Verify that the CMCC is disabled by reading the SR.CSTS value.
2. Enable the CMCC by writing '1' in CTRL.CEN. The module is disabled by writing a '0' in CTRL.CEN.

14.5.3. Change Cache Size

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

Use the following sequence to change the cache size:

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

14.5.4. Data Cache Disable

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

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

4. Enable the CMCC by writing CTRL.CEN = 1.

14.5.5. Instruction Cache Disable

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

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

14.5.6. Cache Load and Lock

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

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

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

14.5.7. Tightly Coupled Memory

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

TCM size = maximum Cache size - configured Cache size.

The TCM start address can be obtained from the product memory mapping. The cache memory starts first from the address followed by the TCM memory. Size of the Way is fixed and the number of ways varies according to the available size for the cache memory. For additional information, refer to the section Product Mapping.

Table 14-1. TCM Sizes

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

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

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

14.5.8. Cache Maintenance

14.5.8.1. Cache Invalidate by Line Operation

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

1. Disable the cache controller by writing a zero to the Cache Controller Enable bit in the Cache Control register (CTRL.CEN).
2. Check SR.CSTS to verify that the CMCC is successfully disabled.
3. Perform an invalidate by line by writing the INDEX and WAY bit fields in the Cache Maintenance 1 register (MAINT1.INDEX, MAINT1.WAY).
4. Enable the CMCC by writing a '1' to CTRL.CEN.

14.5.8.2. Cache Invalidate All Operation

Use the following sequence to invalidate all cache entries.

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

14.5.9. Cache Performance Monitoring

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

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

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

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

Use the following sequence to activate the counter:

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

4. Check the value of the monitor counter by reading the MSR.EVENT_CNT bit field.

14.6. DEBUG Mode

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

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

The TAG, METADATA and DATA RAM blocks' R/W properties are summarized in RAM Properties.

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

- Disable the cache controller by writing a zero to the Cache Controller Enable bit in the Cache Control register (CTRL.CEN).
- Check the Cache Controller Status bit in the Cache Status register (SR.CSTS) to verify that the CMCC is successfully disabled.
- Perform a read or write access through Debugger:
  - CMCC_AHB_ADDR for DATA RAM,
  - CMCC_AHB_ADDR_TAG for TAG RAM,
  - CMCC_AHB_ADDR_MTDATA for METADATA RAM.
- If a write access has been performed in the TAG, METADATA, or DATA RAM in the cache section, an invalid operation must be performed before re-enabling the CMCC.

14.7. RAM Properties

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

Table 14-2. Access to RAM

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

Note:

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

14.8. Register Summary

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

14.8.1. Cache Type

Name: TYPE
Offset: 0x00
Reset: 0x000012D2
Property: Read Only

Table 14-3. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
				CLSIZE[2:0]			CSIZE[2:0]	
Access			R	R	R	R	R	R
Reset			0	1	0	0	1	0

Bit	7	6	5	4	3	2	1	0
	LCKDOWN	WAYNUM[1:0]		RRP			GCLK	
Access	R	R	R	R			R	
Reset	1	1	0	1			1	

Bits 13:11 – CLSIZE[2:0] Cache Line Size
This field configures the Cache Line Size.

Value	Name	Description
0x0-0x1	-	Reserved
0x2	CLSIZE_16B	Cache Line Size is 16 bytes
0x3-0x7	-	Reserved

Bits 10:8 – CSIZE[2:0] Cache Size
This bit field configures the cache size.

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

Bit 7 – LCKDOWN Lock Down Supported
Writing a '0' to this bit disables the Lock Down feature.
Writing a '1' to this bit enables the Lock Down feature.

Value	Description
0	Lock Down feature is not supported.
1	Lock Down feature is supported.

Bits 6:5 – WAYNUM[1:0] Number of Way
This bit field configures the mapping of the cache.

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

Bit 4 – RRP Round Robin Policy Supported
Writing a '0' to this bit disables Round Robin Policy.
Writing a '1' to this bit enables Round Robin Policy.

Value	Description
0	Round Robin Policy is disabled.
1	Round Robin Policy is enabled.

Bit 1 – GCLK Dynamic Clock Gating
Writing a '0' to this bit disables the Dynamic Clock Gating feature.
Writing a '1' to this bit enables the Dynamic Clock Gating feature.

Value	Description
0	Dynamic Clock Gating is disabled.
1	Dynamic Clock Gating is enabled.

14.8.2. Cache Configuration

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

Note: Do not reconfigure or service cache while it is enabled as this could cause corruption.

Table 14-4. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

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

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

Value	Name	Description
`0x0`	CONF_CSIZE_1KB	The Cache Size is configured to 1 KB
`0x1`	CONF_CSIZE_2KB	The Cache Size is configured to 2 KB
`0x2`	CONF_CSIZE_4KB	The Cache Size is configured to 4 KB
`0x3-0x7`	-	Reserved

Bit 2 – DCDIS Data Cache Disable
Writing a '0' to this bit enables data caching.
Writing a '1' to this bit disables data caching.

Value	Description
`0`	Data caching is enabled.
`1`	Data caching is disabled.

Bit 1 – ICDIS Instruction Cache Disable
Writing a '0' to this bit enables instruction caching.
Writing a '1' to this bit disables instruction caching.

Value	Description
0	Instruction caching is enabled.
1	Instruction caching is disabled.

14.8.3. Cache Control

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

Table 14-5. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

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

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

14.8.4. Cache Status

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

Table 14-6. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

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

Bit 0 – CSTS Cache Controller Status
Writing to this bit has no effect.

14.8.5. Cache Lock per Way

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

Table 14-7. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

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

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

Value	Name	Description
`0x0`	WAY0	Way 0 is Locked
`0x1`	WAY1	Way 1 is Locked
`0x2`	WAY2	Way 2 is Locked
`0x3`	WAY3	Way 3 is Locked

14.8.6. Cache Maintenance 0

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

Table 14-8. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

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

Bit 0 – INVALL Cache Controller Invalidate All
Writing a '0' to this bit has no effect.
Writing a '1' to this bit invalidates all cache entries.
Note: The Invalidate command must only be issued when the cache is OFF.

14.8.7. Cache Maintenance 1

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

Table 14-9. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	WAY[3:0]							
Access	W	W	W	W				
Reset	0	0	0	0				

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
					INDEX[7:4]			
Access					W	W	W	W
Reset					0	0	0	0

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

Bits 31:28 – WAY[3:0] Invalidate Way

Value	Name	Description
`0x0`		Reserved
`0x1`	WAY0	Way 0 is selection for index invalidation
`0x2`	WAY1	Way 1 is selection for index invalidation
`0x3`		Reserved
`0x4`	WAY2	Way 2 is selection for index invalidation
`0x5-0x7`		Reserved
`0x8`	WAY3	Way 3 is selection for index invalidation
`0x9-0xF`		Reserved

Bits 11:4 – INDEX[7:0] Invalidate Index
This field selects the index value for invalidation

14.8.8. Cache Monitor Configuration

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

Table 14-10. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

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

Bits 1:0 – MODE[1:0] Cache Controller Monitor Counter Mode
This field selects the type of data monitored.

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

14.8.9. Cache Monitor Enable

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

Table 14-11. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

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

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

Value	Description
0	The Monitor counter is disabled.
1	The Monitor counter is enabled.

14.8.10. Cache Monitor Control

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

Table 14-12. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

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

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

14.8.11. Cache Monitor Status

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

Table 14-13. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	EVENT_CNT[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	EVENT_CNT[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	EVENT_CNT[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

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

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

15. Implementation Defined Attribution Unit (IDAU)

15.1. Overview

The ARMv8-M Architecture defines a mechanism whereby an implementation defined mechanism (i.e., a component in the surrounding system) can define whether any particular address is exempt from checking, and is non-secure callable, or Secure. The Implementation Defined Attribution Unit partitions the memory map into IDAU regions which are subdivisions of the System Bus MAIN regions. IDAU region size and decoding granularity are either fixed or configurable from the APB configuration registers. The MAIN region subdivisions are watermark based.

15.2. IDAU Region Types

The IDAU region types are defined below:

Table 15-1. IDAU Region Types

IDAU REGION TYPE	DESCRIPTION
INVALID	Regions that are not decoded by the BMX.
SEC	Secure region. Region which cannot be accessed by a Non-secure master. Non-secure transactions are discarded inside the CPU and a security fault is generated internally to the core.
NONSEC, NONSEC_W	Non-secure region. Region which is accessible by a non-secure transaction. When a Secure transaction targets a Non-secure region, transaction type is changed to Non-secure. (_W means that the region implements a watermark register, refer to Section 2.9).
SECnNONSEC_B	Regions of this type can be either secure or non-secure. When a region is configured as secure, SEC region type properties apply. When configured as non-secure, NONSEC region type properties apply.
SECnNONSEC_L	This is a linked IDAU region (denoted by the suffix _L). This region type is very similar to the SECnNONSEC_B type. The only difference is that it does not implement any command to change its state (by trying to issue a command generate a CFGERR). Its secure or non-secure state depends on another IDAU region of either SECnNONSEC_B or EXEMPT type. To change its state, issue a command in the linked IDAU region. The new configuration is applied simultaneously to the linked regions. The status of the linked region is readable through RSTATUSB as for the SECnNONSEC_B and EXEMPT types. When the linked region is configured as secure, SEC region type properties apply. When the linked region is configured as non-secure, NONSEC region type properties apply. The “linked” type variant aims at ensuring coherency between the configuration of multiple interfaces of a same macro instance.
NSC_W	Non-secure callable region. Region which allows Non-secure software to call a Secure function, the first instruction in the called Secure function must be a SG (Secure Gateway). (_W means that the region implements a watermark register, refer to Section 2.9).
EXEMPT_B	Region exempt from attribution. The IDAU does not alter the transaction. (_B means that the region implements a block register, refer to Section 2.9).

CAUTION _W and _B suffixes do not have the same meaning for IDAU regions and MAIN regions.

15.3. Main Region Types

The Implementation Defined Attribution Unit bus system address map is strictly based on the system bus matrix address map. Each individual region decoded by the bus matrix is referred to as MAIN region further in this document. Each MAIN region is given a type.

The MAIN region type defines:

- The number of IDAU regions inside a MAIN region
- The type of each IDAU region inside a MAIN region

There is one valid region partition type:

- **WATERMARK:MAIN** region is divided by configuring the size of one or more watermarks (refer to Splittable Main Regions). Main region types using the watermark partition type are suffixed _W.

Splittable MAIN regions are split into several IDAU regions that each have a unique ID. In each watermark partition-able MAIN region, at least one IDAU region has a watermark configuration. Such configurable IDAU regions are suffixed _W. MAIN region types that are not splittable do not have a underscore in their name.

CAUTION _W and _B suffixes do not have the same meaning for IDAU regions and MAIN regions.

Table 15-2. Main Region Types

MAINREGION TYPE	DESCRIPTION	PARTITION TYPE
MR_INVALID	Single IDAU region of INVALID type	NONE
MR_SEC	Single IDAU region of SEC type	NONE
MR_NONSEC	Single IDAU region of NONSEC type	NONE
MR_SEC_NSC_W	Contains two IDAU regions of type (lowest to highest addresses): SEC NSC_W(size is configurable with a watermark register)	WATERMARK
MR_SEC_NSC_NONSEC_W	Contains three IDAU regions of type (lowest to highest addresses): SEC NSC_W(size is configurable) NONSEC_W(size is configurable with a watermark register)	WATERMARK
MR_SEC_NONSEC_W	Contains two IDAU regions of type (lowest to highest addresses): SEC NONSEC_W(size is configurable with a watermark register)	WATERMARK
MR_SECnNONSEC	Single IDAU region of SECnNONSEC_B type (either non-secure or secure).A SECnNONSEC_B IDAU region has a dedicated configuration register associated to it.	NONE
MR_SECnNONSEC_L	Single IDAU region of SECnNONSEC_L (either non-secure or secure).A SECnNONSEC_L IDAU region does not have a dedicated configuration register associated to it. It is using the configuration bit of another main region of MR_SECnNONSEC or MR_EXEMPT type.	NONE
MR_EXEMPT	Single IDAU region of EXEMPT_B type (exempt from security)	NONE

Note: IDAU region types are detailed in the RSTATUS[A,B] table.

15.4. Splittable Main Regions

Each IDAU region is associated with one MAIN region. When more than one IDAU regions are derived from a single MAIN region, this MAIN region is said to be splittable.

The following MAIN region types are not splittable:

- MR_INVALID
- MR_SEC
- MR_NONSEC
- MR_SECnNONSEC
- MR_EXEMPT

The following IDAU region types are splittable:

- MR_SEC_NSC_W

- MR_SEC_NSC_NONSEC_W
- MR_SEC_NONSEC_W

The following figure shows how the split is done for each MAIN region type. The vertical arrow shows which IDAU region size is configurable. The size of each MAIN region is fixed.

Figure 15-1. Splittable Main Regions

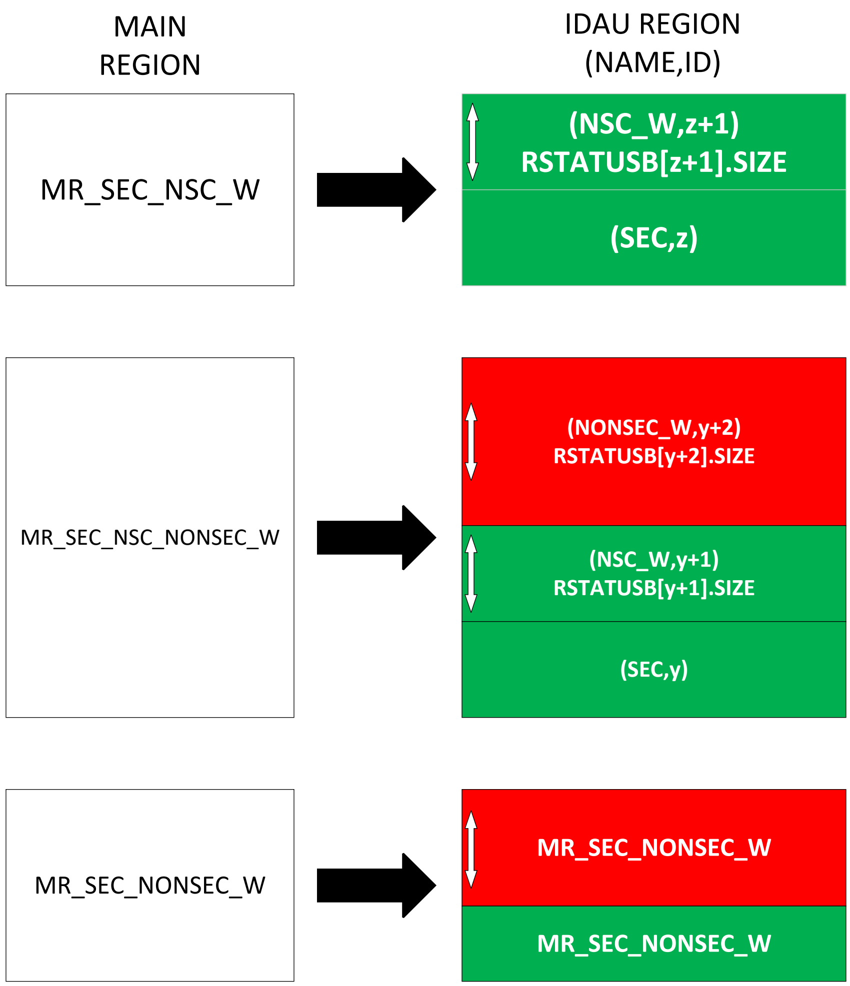


Table 15-3. Splittable Main Regions Configuration

SPLITTABLE MAIN REGION TYPE	DESCRIPTION
MR_SEC_NSC_W	NSC_W size is configurable using the RCTRL.CMD = WRSZ. Minimum size is 0 so that the SEC IDAU region size equals to the MAIN region size.
MR_SEC_NSC_NONSEC_W	NONSEC_W size is configurable using the RCTRL.CMD = WRSZ. The NSC_W and SEC IDAU regions occupy the rest of the main region. Minimum size is 0 allowing the SEC and NSC_W IDAU regions to occupy the whole MAIN region space. NSC_W size is configurable using the RCTRL.CMD = WRSZ and occupies the space left by the NONSEC_W region inside the main region. The SEC IDAU region occupies the space left by the NONSEC_W and NSC_W IDAU regions. Minimum size for the NSC_W is 0 allowing the SEC IDAU region size to occupy the MAIN region size when both NSC_W and NONSEC_W sizes are 0.

Table 15-3. Splittable Main Regions Configuration (continued)

SPLITTABLE MAIN REGION TYPE	DESCRIPTION
MR_SEC_NONSEC_W	NONSEC_W size is configurable using the RCTRL.CMD = WRSZ. Minimum size is 0 so that the SEC IDAU region size equals to the MAIN region size.

Figure 15-2. Splittable Main Regions Configurations

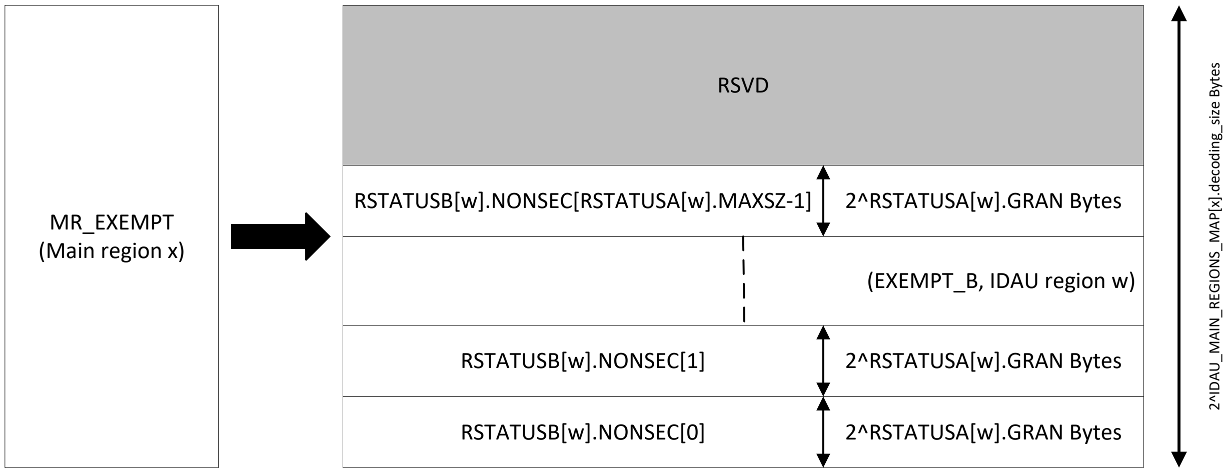
Notes:
1. Impossible configuration due to subregion1 maximum size constraint (legal values are integers in the range of [1,decoding_size-subregion1_granularity]), refer to the "IDAU Main Regions Map".
2. Impossible configuration due to subregion2 maximum size constraint (legal values are integers in the range of [1,decoding_size-subregion2_granularity]), refer to the "IDAU Main Regions Map".
3. NSC_W can overlap the SEC region when the NONSEC_W region size is not null and NSC_W size is greater than 0. Even if the NSC_W size exceeds the remaining space not occupied by the NONSEC_W region in the MAIN region, the NSC_W will only overlap the SEC region, it cannot overlap contiguous MAIN regions.

15.5. Exempt_B Layout

The MAIN region (x) induces a single IDAU region (w). It is split into 32 equally sized blocks of 2^RSTA- TUSA[w].GRAN Bytes. Each block has its own NON-SEC bit mapped in RSTATUSB[w]. Only the first RSTATUSA[w].MAXSZ is configurable. RSTA-TUSB[x] is a constant 1 when 32<x<RSTATUSA[w].MAXSZ-1.

The size of the MAIN region is 2^IDAU_MAIN_REGIONS_MAP[x].decoding_size Bytes which must be greater than or equal to 32*2^RSTATUSA[w].GRAN Bytes.

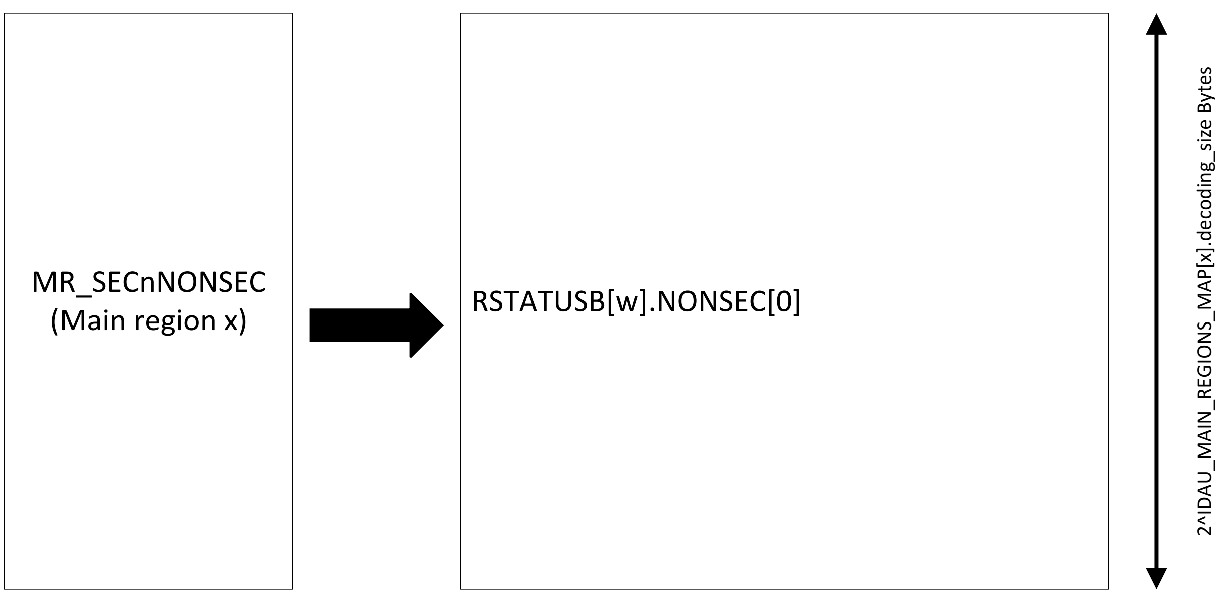
Figure 15-3. MR_EXEMPT



15.6. SECNNONSEC_B Layout

The maximum number of blocks is 1 and the size of the IDAU region is the same as the MAIN region.

Figure 15-4. SECNNONSEC_B Layout



15.7. IDAU Regions IDS

IDAU region identifiers are fixed. ID 0 is reserved for INVALID regions. The maximum legal ID is 0xFF. A region ID is defined by the IDAU_MAIN_REGIONS_MAP parameter and depends on the previously declared main regions and their types.

15.8. IDAU Regions Configuration

The module is disabled at reset (APB). This is reflected by STATUSA.ENABLE. All IDAU regions except those that are marked secure in the "disabled" column for the IDAU in the Product Mapping Section are considered exempt from security when the module is disabled. Only IDAU regions belonging to a main region of MR_SEC or MR_NON- SEC type can be valid when the module is disabled, which is typically the case for the main region corresponding to the Boot ROM. All IDAU region watermark settings are zeroed internally and exported as such to the system when STATUSA.ENABLE = 0 (associated regions cannot be reached). All IDAU region block settings are forced to 1's internally and exported as such to the system when STATUSA.ENABLE = 0 (forced to non-secure state). This allows to write the secure configuration and to apply it to the system at once by enabling the module and therefore to prevent intermediate states to appear at system level while writing the configuration registers. The device Boot ROM and/or secure boot are in charge of configuring the module. All IDAU regions settings MUST be initialized prior to enabling the module, refer to the Command and Arguments table to identify configurable regions. The state of the configuration registers is unknown at reset.

Each IDAU region has its own register group composed of RCTRL, RSTATUSA, RSTATUSB and RSTATUSC registers. The number of implemented register groups is readable from STATUSA.NBRG and cannot exceed 256.

Region register groups are mapped at offset 0x1000:

- **RCTRL[w]** is mapped at offset 0x1000+w*0x10
- **RSTATUSA[w]** is mapped at offset 0x1000+w*0x10+0x4
- **RSTATUSB[w]** is mapped at offset 0x1000+w*0x10+0x8
- **RSTATUSC[w]** is mapped at offset 0x1000+w*0x10+0xC

To configure an IDAU region, first read its RSTATUSA register to determine its type. The type will determine which commands are legal and legal argument values.

Table 15-4. RSTATUS[A,B]

IDAU Region Type: RSTATUSA.TYPE	RSTATUSA.GRAN	RSTATUSA.MAXSZ	RSTATUSB.[SIZE,NONSEC]
INVALID	**N/A(reads 0)**	**N/A(reads 0)**	**N/A(reads 0)**
SEC	**N/A(reads 0)**	**N/A(reads 0)**	**N/A(reads 0)**
NONSEC	**N/A(reads 0)**	**N/A(reads 0)**	**N/A(reads 0)**
NONSEC_W	**CFG** **Size in Bytes of one RCTRL[w].ARGLSb**	**CFG** **MAXSZ-1is the maximum value of RCTRL[w].ARG**	IDAU region size in Bytes
SECnNONSEC_L	**N/A(reads 0)**	**N/A(reads 0)**	Defines the state of the region/block: non-secure when NONSEC[0] = 1, secure otherwise. This is a mirror of the linked IDAU region RSTATUSB block NONSEC bit.
NSC_W	**CFG** **Size in Bytes of one RCTRL[w].ARGLSb**	**CFG** **MAXSZ-1is the maximum value of RCTRL[w].ARG**	IDAU region size in Bytes
EXEMPT_B	**CFG** **Size in Bytes of a block**	**CFG** **Maximum number of blocks inside the region**	Defines the state of the region blocks: For each valid NONSEC[y], y<MAXSZ block y is non-secure when NONSEC[y] = 1, secure otherwise

Notes:
1. **Bolded** cells denote static configuration (combo logic) whereas white cells denote a configuration that can be modified using RCTRL commands.
2. _W suffix indicates that the IDAU region has a watermark configuration.
3. _B suffix indicates that the IDAU region is divided into blocks that can be configured as secure or Non-secure.
4. _L suffix indicates that the configuration of the IDAU region is linked to another one.
5. **CFG:** Chip specific information extracted from the IDAU_MAIN_REGIONS_MAP parameter.

Table 15-5. Command and Arguments

RSTATUSA.TYPE	Valid RCTRL.CMD commands	RCTR.ARG Valid when
INVALID	None	N/A
SEC	None	N/A
NONSEC	None	N/A
NONSEC_W	WRSZ	< RSTATUSA.MAXSZ
SECnNONSEC_B	[CLR,SET]NONSEC	0
SECnNONSEC_L	None	N/A
NSC_W	WRSZ	< RSTATUSA.MAXSZ (1)
EXEMPT_B	[CLR,SET]NONSEC	< RSTATUSA.MAXSZ

For each IDAU region, the command and argument validity are checked by the macro. When a command or argument is not valid:

- The command is discarded
- An error is reported to STATUSB.CFGERR

Before writing CTRL.ENABLE = 1:

- All configurable settings must be written (see the non-bolded cells in the RSTATUS[A,B] table)
- STATUSB.CFGERR must be read 0

The CTRL.CMD ENABLE command is discarded when STATUSB.CFGERR is high.

Once STATUSA.ENABLE is set, configuration is applied at the system level.

Note:
1. The NSC_W size can exceed the remaining space size left by the NONSEC IDAU region in the MAIN region. In this case, no CFGERR error is reported and the NSC_W region overlaps exactly the whole SEC IDAU region, the NSC_W region cannot overlap MAIN regions it does not belong to.

15.9. Configuration Write Lock

The CTRL and all RCTR registers are write-protected after writing CTRL.CMD = WLCK. The write protection status is reflected by STATUSA.WLCK. Only a module reset can revert the write-protection.

These registers are also write-protected by the PAC.

STATUSB has no write-protection: it can be cleared on write when the configuration is locked and/or the module is PAC write-protected.

15.10. Linked IDAU Regions

Many peripherals implement several memory spaces (often command and data) mapped at different locations in the global memory map. These correspond to different IDAU regions that cannot be merged. Only one IDAU region is configurable, and others use a linked IDAU region type that instructs the user that the region state is driven by another IDAU region. This is identified in the RSTATUSC register. The secure state (secure or non-secure) is also readable from the RSTATUSB register.

15.11. IDAU Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRL	31:24	CMD[15:8]							
		23:16	CMD[7:0]							
		15:8								
		7:0								
0x04	STATUSA	31:24								
		23:16								
		15:8	NBRG[7:0]							
		7:0							WLCK	ENABLE
0x08	STATUSB	31:24								
		23:16								
		15:8								
		7:0								CFGERR
0x0C ... 0x0FFF	Reserved									
0x1000	RCTRL	31:24	CMD[7:0]							
		23:16	ARG[23:16]							
		15:8	ARG[15:8]							
		7:0	ARG[7:0]							
0x1004	RSTATUSA	31:24								
		23:16			MAXSZ[5:0]					
		15:8			GRAN[5:0]					
		7:0					TYPE[3:0]			
0x1008	RSTATUSB	31:24	SIZE[31:24]							
		23:16	SIZE[23:16]							
		15:8	SIZE[15:8]							
		7:0	SIZE[7:0]							
0x100C	RSTATUSC	31:24								
		23:16								
		15:8				BLK[4:0]				
		7:0	RGN[7:0]							

15.11.1. Control Register

Name: CTRL
Offset: 0x0000
Reset: 0x00000000
Property: PAC Write-Protection

This Register is write-protected by the PAC and by STATUSA.WLCK.

Invalid commands are reported in STATUSB.CFGERR and not reported to the PAC.

Table 15-6. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	CMD[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
	CMD[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
Access								
Reset								

Bits 31:16 – CMD[15:0] Command Register
The ENABLE command is discarded when STATUSB.CFGERR is high.

Value	Description
`0x0 – 0xA500`	Reserved
`0xA501`	Module Enable (ENABLE)
`0xA502`	Module Disable (DISABLE)
`0xA503`	Write Lock (WLCK)
`0xA504 – 0xFFFF`	Reserved

15.11.2. STATUSA Register

Name: STATUSA
Offset: 0x0004
Reset: 0x00000000
Property: PAC Write-Protection

Table 15-7. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
	NBRG[7:0]							
Access	cfg/R	cfg/R	cfg/R	cfg/R	cfg/R	cfg/R	cfg/R	cfg/R
Reset	0	0	0	0	0	0	0	0

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

Bits 15:8 – NBRG[7:0] Number of IDAU Regions
Indicates the number of IDAU Regions.
The Region register group composed of RCTRL, RSTATUS[A,B,C] is repeated NBRG+1 times.
Region0 is always present and always of MR_INVALID type.
Regions1 to NBRG configuration and status registers are accessible through the IDAU Region register groups.

Bit 1 – WLCK Write Lock

Value	Description
0	The module is not Write-protected.
1	The module is Write-protected until next reset, writes are discarded and reported as PAC errors.

Bit 0 – ENABLE Enable

Value	Description
0	The module is disabled, all regions are considered exempt from security except those configured as valid when disabled (refer to the IDAU_MAIN_REGIONS_MAP parameter).
1	The module is enabled, dynamic configuration is applied to the system.

15.11.3. STATUSB Register

Name: STATUSB
Offset: 0x0008
Reset: 0x00000000
Property: PAC Write-Protection

Note: This Register is not write-protected.

Table 15-8. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								CFGERR
Access								R/K
Reset								0

Bit 0 – CFGERR Configuration Error
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the CFGERR bit.
This bit is set when an invalid command is issued (from CTRL or RCTRL[x] registers). This bit is set when a valid command is issued with an invalid argument.

15.11.4. (Mode=Watermark) - Region Control Register

Name: RCTRL
Offset: 0x1000
Reset: 0x00000000
Property: PAC Write-Protection

This register is part of a register group (composed of RCTRL, RSTATUS[A,B,C]) that is repeated IDAU_REGION_NUM times. Register group x controls and returns status of the IDAU region x.

Table 15-9. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	CMD[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	ARG[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
	ARG[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
	ARG[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bits 31:24 – CMD[7:0] Command
The commands below apply to IDAU region types suffixed _W. Illegal commands are reported to STATUSB.CFGERR.

Value	Description
`0x0 – 0x5B`	Reserved
`0x5C`	Write IDAU region x Size (RSTATUSB[x].SIZE=ARG) (WRSZ)
`0x5D – 0xFF`	Reserved

Bits 23:0 – ARG[23:0] Command Argument (Watermark)
Argument defines the watermark in the associated main region that splits it in 2 distinct IDAU regions. This watermark is an offset in Bytes from the top address of the MAIN region equal to ARG.2^RSTATUSA[x].GRAN Bytes. ARG must be strictly lower than 2^(RSTATUSA[x].MAXSZ). The size of the IDAU region is ARG times the size of a single block of 2^RSTATUSA[x].GRAN Bytes .
Note: The block term used here as a generic term and does not refer to block type.

CAUTION ARG meaning in the WATERMARK mode differs from the BLOCK mode.

15.11.5. Region STATUSA Register

Name: RSTATUSA
Offset: 0x1004
Reset: 0x00000000
Property: PAC Write-Protection

This register is part of a register group (composed of RCTRL, RSTATUS[A,B,C]) that is repeated IDAU_REGION_NUM times. Register group x controls and returns status of the IDAU region x.

This register is meaningful only for IDAU region types suffixed _B and _W (static configuration), for other types it always reads 0.

Table 15-10. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
			MAXSZ[5:0]					
Access			cfg/R	cfg/R	cfg/R	cfg/R	cfg/R	cfg/R
Reset			x	x	x	x	x	x

Bit	15	14	13	12	11	10	9	8
			GRAN[5:0]					
Access			cfg/R	cfg/R	cfg/R	cfg/R	cfg/R	cfg/R
Reset			x	x	x	x	x	x

Bit	7	6	5	4	3	2	1	0
					TYPE[3:0]			
Access					cfg/R	cfg/R	cfg/R	cfg/R
Reset					x	x	x	x

Bits 21:16 – MAXSZ[5:0] Command Argument Maximum Size

Bits 13:8 – GRAN[5:0] Region Granularity
_W suffixed types (Watermark based regions): Size in Bytes of one RCTRL[x].ARG LSb.
_B suffixed types (Block based regions): Size in Bytes of a block.

Value	Description
`0x0`	Not Applicable
`0x1 - 0xB`	Reserved
`0xC`	(GRAN_4KB)
`0xD`	(GRAN_8KB)
`0xE`	(GRAN_16KB)
`0xF`	(GRAN_32KB)
`0x10`	(GRAN_64KB)
`0x11`	(GRAN_128KB)
`0x12`	(GRAN_256KB)

Value	Description
0x13	(GRAN_512KB)
0x14	(GRAN_1MB)
0x15	(GRAN_2MB)
0x16	(GRAN_4MB)
0x17	(GRAN_8MB)
0x18	(GRAN_16MB)
0x19	(GRAN_32MB)
0x1A	(GRAN_64MB)
0x1B	(GRAN_128MB)
0x1C	(GRAN_256MB)
0x1D	(GRAN_512MB)
0x1E	(GRAN_1GB)
0x1F	(GRAN_2GB)
0x20	(GRAN_4GB)
0x21 – 0x3F	Reserved

Bits 3:0 – TYPE[3:0] IDAU Reion Type

Value	Description
0x0	Region is invalid (INVALID)
0x1	Region is secure (SEC)
0x2	Region is non secure (NONSEC)
0x3	Region is non secure, the associated register group implements a watermark configuration (NONSEC_W)
0x4	Region is either secure or non-secure, the associated register group implements a block configuration (SECnNONSEC_B)
0x5	Region is either secure or non-secure, configuration is linked to another IDAU region configuration bit (the linked configuration bit identification info is readable from the RSTATUSC register inside the same register group) (SECnNONSEC_L)
0x6	Region is non-secure callable, the associated register group implements a watermark configuration (NSC_W)
0x7	Region is exempt from security, the associated register group implements a block configuration (EXEMPT_B)
0x8 – 0xF	Reserved

15.11.6. (Mode = Watermark) - Region STATUSB Register

Name: RSTATUSB
Offset: 0x1008
Reset: 0x00000000
Property: PAC Write-Protection

This register is part of a register group (composed of RCTRL, RSTATUS[A,B,C]) that is repeated IDAU_REGION_NUM times. Register group x controls and returns status of the IDAU region x.

This register contains the dynamic configuration of region x.

This register is meaningful only for IDAU region types suffixed _B and _W, for other types it always reads 0.

Table 15-11. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	SIZE[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
	SIZE[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
	SIZE[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
	SIZE[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	x	x	x	x	x	x	x	x

Bits 31:0 – SIZE[31:0] Watermark Based Region Size in Bytes

15.11.7. (MODE=LINK) - Region STATUSC Register

Name: RSTATUSC
Offset: 0x100C
Reset: 0x00000000
Property: PAC Write-Protection

This register is part of a register group (composed of RCTRL, RSTATUS[A,B,C]) that is repeated IDAU_REGION_NUM times. Register group x controls and returns status of the IDAU region x.

This register is meaningful only for block-based IDAU regions (_B suffixed types), otherwise it reads 0.

This register exists only in register groups where RSTATUSA.TYPE = EXEMPT_B or RSTA-TUSA.TYPE = SECnNONSEC_L. Otherwise, accesses at this register group offset are illegal.

Table 15-12. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
				BLK[4:0]				
Access				R	R	R	R	R
Reset				x	x	x	x	x

Bit	7	6	5	4	3	2	1	0
	RGN[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	x	x	x	x	x	x	x	x

Bits 12:8 – BLK[4:0] Linked IDAU Region Block ID

Bits 7:0 – RGN[7:0] Linked IDAU Region ID

16. Peripheral Access Controller (PAC)

16.1. Overview

The Peripheral Access Controller (PAC) provides an interface for the locking and unlocking of peripheral registers within the device. It reports all violations which occur due to an improper access of a peripheral: write protected access, illegal access, enable protected access, or access when clock synchronization or software reset is on-going. These errors are reported with a unique interrupt flag for each peripheral. The PAC module also reports errors occurring at the client bus level, when an access to a non-existing address is detected.

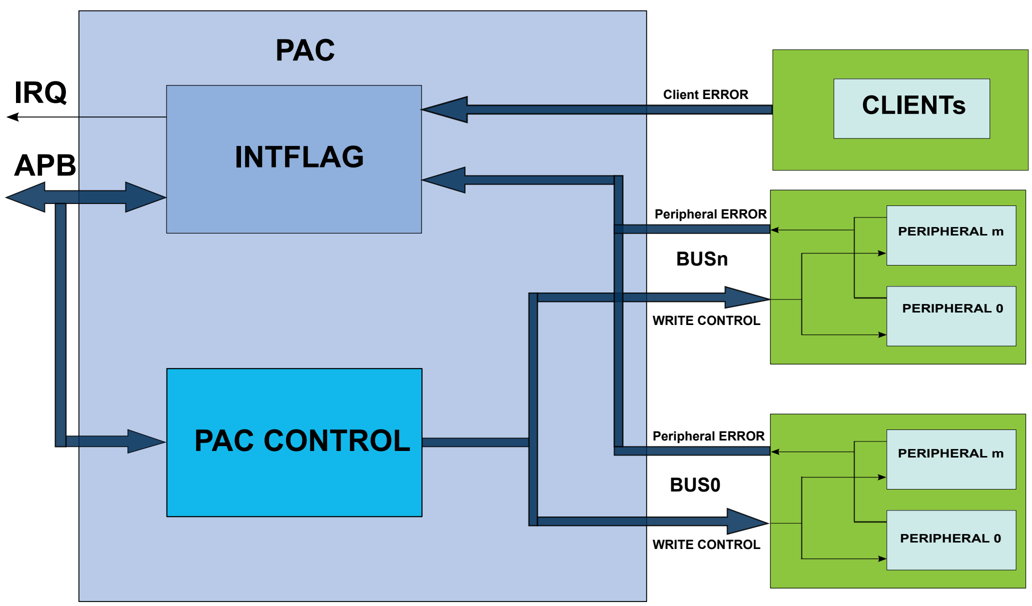
16.2. Features

The following are key features of the PAC module:

- Programmable write protect bit for each peripheral that supports it
- Interrupt generation on access errors to peripherals
  - Write protect error
  - Unused Address
  - Access when clock synchronization is active
  - Access when software reset is on-going
  - Access to reserved memory regions

16.3. Block Diagram

Figure 16-1. PAC Block Diagram



16.4. Peripheral Dependencies

Peripheral Name	Base Address	NVIC IRQ Index: Source	MCLK AHBx/APBx Clock Enable Mask Bit	PAC Peripheral Identifier (PAC.WRCTRL.PERIDx)	EVSYS Generator (EVSYS.CHANNELn.EVGENx)	Power Domain
PAC	0x4401_C000	30 : ERR	AHB: MCLK.CLKMSK0[6] APB: MCLK.CLKMSK1[14]	14	37 : ACCERR	VDDREG

16.4.1. Debug Operation

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

16.4.2. Register Access Protection

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

- The Write Control (WRCTRL) register
- The Peripheral Interrupt Flag Status and Clear n (INTFLAG A/B/C...) registers

Optional write protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write Protection" property in each individual register description.

PAC write protection does not apply to accesses through an external debugger.

16.5. Functional Description

16.5.1. Basic Operation

16.5.1.1. Operations

The PAC module allows the user to set, clear, or lock the write protected status of any peripherals. If a peripheral access violation occurs, the Peripheral Interrupt Flag registers (INTFLAGx, x=A, B, C, or D) are updated to inform the user which peripherals have had access violations. The corresponding Peripheral Write Control Status n register (STATUSx) gives the state of the write protection for all peripherals.

16.5.1.2. 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 one of the INTFLAGx 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 INTFLAGx registers bits are set, an interrupt will be requested if the PAC interrupt enable bit is set.

16.5.1.3. 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 a 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.

Any peripheral access control status can be read from a corresponding STATUSn register.

16.5.1.4. Write Access Protection Management Errors

Only word-wise (or register-wise) writes to the WRCTRL register will change the access protection. Other types of accesses will have no effect and will cause a PAC write access error. This error is reported in the INTFLAGA.PAC bit.

The PAC also generates an interrupt on double write clear or double write set operations.

- If a peripheral is write protected and a subsequent set protection or set and lock operation is detected then the PAC returns an error.
- If a peripheral is not protected and a subsequent clear protection operation is detected then the PAC returns an error.
- If a peripheral is write protected and locked and either a clear protection or set protection operation is detected, then the PAC returns an error.

This can be used to ensure that the application follows any write protect with an unprotect and any unprotect with a protect. In applications where the write protect state of a peripheral is manipulated in several contexts, care must be taken so that the contexts do not interfere with each other.

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

16.5.1.5. 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'.

16.5.2. 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.

The interrupt flag in the Interrupt Flag Status and Clear (INTFLAGAHB and INTFLAGx) 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 INTFLAGx registers to determine which interrupt condition is present.

16.5.3. 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.

16.5.4. 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.

16.6. Register Summary

For descriptions and definitions of both Register and bitfield properties, refer to Register Properties.

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 ... 0x13	Reserved									
0x14	INTFLAGA	31:24								
		23:16								
		15:8	TRAM	PAC	EIC	RTC	WDT	FREQM	MCLK	GCLK
		7:0	OSC32KCTRL	OSCCTRL	RSTC	SUPC	PM	FCW	FCR	DSU
0x18	INTFLAGB	31:24								
		23:16								
		15:8			TCC3	TCC2	TCC1	TCC0	SERCOM3	SERCOM2
		7:0	SERCOM1	SERCOM0	EVSYS	IDAU	PRM	DMA1	DMA0	PORT
0x1C	INTFLAGC	31:24								
		23:16							USB	TRNG
		15:8	ETH	PDEC	CCL		SPI_IXS	PTC	AC	ADC
		7:0	TCC7	TCC6	TCC5	TCC4	SERCOM7	SERCOM6	SERCOM5	SERCOM4
0x20	INTFLAGD	31:24								
		23:16								
		15:8								
		7:0								SQI
0x24 ... 0x33	Reserved									
0x34	STATUSA	31:24								
		23:16								
		15:8	TRAM	PAC	EIC	RTC	WDT	FREQM	MCLK	GCLK
		7:0	OSC32KCTRL	OSCCTRL	RSTC	SUPC	PM	FCW	FCR	DSU
0x38	STATUSB	31:24								
		23:16								
		15:8			TCC3	TCC2	TCC1	TCC0	SERCOM3	SERCOM2
		7:0	SERCOM1	SERCOM0	EVSYS	IDAU	PRM	DMA1	DMA0	PORT
0x3C	STATUSC	31:24								
		23:16							USB	TRNG
		15:8	ETH	PDEC	CCL		SPI_IXS	PTC	AC	ADC
		7:0	TCC7	TCC6	TCC5	TCC4	SERCOM7	SERCOM6	SERCOM5	SERCOM4
0x40	STATUSD	31:24								
		23:16								
		15:8								
		7:0								SQI

16.6.1. Write Control

Name: WRCTRL
Offset: 0x00
Reset: 0x00000000
Property: –

Table 16-1. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	KEY[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
	PERID[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
	PERID[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 – KEY[7:0] Peripheral Access Control Key
These bits define the peripheral access control key:

Value	Name	Description
0x0	OFF	No action
0x1	CLEAR	Clear the peripheral write control
0x2	SET	Set the peripheral write control
0x3	LOCK	Set and lock the peripheral write control until the next hardware reset

Bits 15:0 – PERID[15:0] Peripheral Identifier
The PERID represents the peripheral whose control is changed using the WRCTRL.KEY.

Table 16-2. PERID Values

Peripheral	PERID	SFR Suffix	Bit	SFR's	
				STATUS	INTFLAG
DSU	0	A	0	STATUSA[0]	INTFLAGA[0]
FCR	1	A	1	STATUSA[1]	INTFLAGA[1]
FCW	2	A	2	STATUSA[2]	INTFLAGA[2]
PM	3	A	3	STATUSA[3]	INTFLAGA[3]
SUPC	4	A	4	STATUSA[4]	INTFLAGA[4]
RSTC	5	A	5	STATUSA[5]	INTFLAGA[5]

Table 16-2. PERID Values (continued)

Peripheral	PERID	SFR Suffix	Bit	SFR's	
				STATUS	INTFLAG
OSCCTRL	6	A	6	STATUSA[6]	INTFLAGA[6]
OSC32KCTRL	7	A	7	STATUSA[7]	INTFLAGA[7]
GCLK	8	A	8	STATUSA[8]	INTFLAGA[8]
MCLK	9	A	9	STATUSA[9]	INTFLAGA[9]
FREQM	10	A	10	STATUSA[10]	INTFLAGA[10]
WDT	11	A	11	STATUSA[11]	INTFLAGA[11]
RTC	12	A	12	STATUSA[12]	INTFLAGA[12]
EIC	13	A	13	STATUSA[13]	INTFLAGA[13]
PAC	14	A	14	STATUSA[14]	INTFLAGA[14]
TRAM	15	A	15	STATUSA[15]	INTFLAGA[15]
MBISTINTF	18				
TDM	19				
FlexRAM	20				
PORT	32	B	0	STATUSB[0]	INTFLAGB[0]
DMA0	33	B	1	STATUSB[1]	INTFLAGB[1]
DMA1	34	B	2	STATUSB[2]	INTFLAGB[2]
HMATRIX	35				
CMCC	36				
PRM	37	B	3	STATUSB[3]	INTFLAGB[3]
IDAU	38	B	4	STATUSB[4]	INTFLAGB[4]
EVSYS	39	B	5	STATUSB[5]	INTFLAGB[5]
SERCOM0	40	B	6	STATUSB[6]	INTFLAGB[6]
SERCOM1	41	B	7	STATUSB[7]	INTFLAGB[7]
SERCOM2	42	B	8	STATUSB[8]	INTFLAGB[8]
SERCOM3	43	B	9	STATUSB[9]	INTFLAGB[9]
TCC0	44	B	10	STATUSB[10]	INTFLAGB[10]
TCC1	45	B	11	STATUSB[11]	INTFLAGB[11]
TCC2	46	B	12	STATUSB[12]	INTFLAGB[12]
TCC3	47	B	13	STATUSB[13]	INTFLAGB[13]
SERCOM4	64	C	0	STATUSC[0]	INTFLAGC[0]
SERCOM5	65	C	1	STATUSC[1]	INTFLAGC[1]
SERCOM6	66	C	2	STATUSC[2]	INTFLAGC[2]
SERCOM7	67	C	3	STATUSC[3]	INTFLAGC[3]
TCC4	68	C	4	STATUSC[4]	INTFLAGC[4]
TCC5	69	C	5	STATUSC[5]	INTFLAGC[5]
TCC6	70	C	6	STATUSC[6]	INTFLAGC[6]
TCC7	71	C	7	STATUSC[7]	INTFLAGC[7]
ADC	72	C	8	STATUSC[8]	INTFLAGC[8]
AC	73	C	9	STATUSC[9]	INTFLAGC[9]
PTC	74	C	10	STATUSC[10]	INTFLAGC[10]
SPI_IXS	75	C	11	STATUSC[11]	INTFLAGC[11]
PCC	76				
CCL	77	C	13	STATUSC[13]	INTFLAGC[13]
PDEC	78	C	14	STATUSC[14]	INTFLAGC[14]
CAN0	79				
CAN1	80				
ETH	81	C	15	STATUSC[15]	INTFLAGC[15]
TRNG	82	C	16	STATUSC[16]	INTFLAGC[16]
SDMCC0	83				
SDMCC1	84				
USB	85	C	17	STATUSC[17]	INTFLAGC[17]

Table 16-2. PERID Values (continued)

Peripheral	PERID	SFR Suffix	Bit	SFR's	
				STATUS	INTFLAG
EBI	86				
SQI	96	D	0	STATUSD[0]	INTFLAGD[0]
USBHS	97				
HSM	98				

16.6.2. Event Control

Name: EVCTRL
Offset: 0x04
Reset: 0x00
Property: -

Table 16-3. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
								ERREO
Access								R/W
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.

16.6.3. 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 register (INTENSET).

Table 16-4. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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

Bit 0 – ERR Peripheral Access Error Interrupt Disable
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.

16.6.4. 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 Set register (INTENCLR).

Table 16-5. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
								ERR
Access								R/W
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.

16.6.5. Peripheral Interrupt Flag Status and Clear A

Name: INTFLAGA
Offset: 0x14
Reset: 0x00000000
Property: -

This flag is set when a Peripheral Access Error occurs while accessing the peripheral associated with the respective INTFLAGA bit. An interrupt request is generated if INTENCLR/SET.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.

Table 16-6. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
	TRAM	PAC	EIC	RTC	WDT	FREQM	MCLK	GCLK
Access	R/W	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
	OSC32KCTRL	OSCCTRL	RSTC	SUPC	PM	FCW	FCR	DSU
Access	R/W	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 – TRAM Trust RAM

Bit 14 – PAC Peripheral Access Controller

Bit 13 – EIC External Interrupt Controller

Bit 12 – RTC Real Time Clock

Bit 11 – WDT Watchdog Timer

Bit 10 – FREQM Frequency Meter

Bit 9 – MCLK Main Clock

Bit 8 – GCLK Generic Clock Controller

Bit 7 – OSC32KCTRL 32 K Oscillator Controller

Bit 6 – OSCCTRL Oscillator Controller

Bit 5 – RSTC Reset Controller

Bit 4 – SUPC Startup Power Controller

Bit 3 – PM Interrupt Flag for the Power Manager

Bit 2 – FCW Flash Controller Write

Bit 1 – FCR Flash Controller Read

Bit 0 – DSU Device Service Unit

16.6.6. Peripheral Interrupt Flag Status and Clear B

Name: INTFLAGB
Offset: 0x18
Reset: 0x00000000
Property: -

This flag is set when a Peripheral Access Error occurs while accessing the peripheral associated with the respective INTFLAGB bit. An interrupt request is generated if INTENCLR/SET.ERR is '1'.

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

Writing a '1' to this bit will clear the corresponding INTFLAGB interrupt flag.

Table 16-7. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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**
			TCC3	TCC2	TCC1	TCC0	SERCOM3	SERCOM2
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**
	SERCOM1	SERCOM0	EVSYS	IDAU	PRM	DMA1	DMA0	PORT
Access	R/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 10, 11, 12, 13 – TCCn Interrupt Flag for Timer Counter Controller, n = 0..3

Bits 6, 7, 8, 9 – SERCOMn Serial Communication Interface n, n = 0..3

Bit 5 – EVSYS Event System

Bit 4 – IDAU Implementation Defined Attribution Unit

Bit 3 – PRM Host Boot ROM

Bits 1, 2 – DMAn Direct Memory Access n = 0..1

Bit 0 – PORT PORT General Purpose Pin I/O Controller

16.6.7. Peripheral Interrupt Flag Status and Clear C

Name: INTFLAGC
Offset: 0x1C
Reset: 0x00000000
Property: -

This flag is set when a Peripheral Access Error occurs while accessing the peripheral associated with the respective INTFLAGC bit. An interrupt request is generated if INTENCLR/SET.ERR is '1'.

Writing a '0' to these bits has no effect.

Writing a '1' to these bits clears the corresponding INTFLAGC interrupt flag.

Table 16-8. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
							USB	TRNG
Access							R/W	R/W
Reset							0	0

Bit	15	14	13	12	11	10	9	8
	ETH	PDEC	CCL		SPI_IXS	PTC	AC	ADC
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
	TCC7	TCC6	TCC5	TCC4	SERCOM7	SERCOM6	SERCOM5	SERCOM4
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 17 – USB Universal Serial Bus

Bit 16 – TRNG True Random Number Generator

Bit 15 – ETH Ethernet

Bit 14 – PDEC Position Decoder

Bit 13 – CCL Configurable Custom Logic

Bit 11 – SPI_IXS Inter-IC Sound Interface

Bit 10 – PTC Peripheral Touch Controller

Bit 9 – AC Analog Comparator

Bit 8 – ADC Analog-to-Digital Converter

Bits 4, 5, 6, 7 – TCCn Timer Counter Controller n, n = 4..7

Bits 0, 1, 2, 3 – SERCOMn Serial Communication Interface n, n = 4..7

16.6.8. Peripheral Interrupt Flag Status and Clear D

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

This flag is set when a Peripheral Access Error occurs while accessing the peripheral associated with the respective INTFLAGD bit. An interrupt request is generated if INTENCLR/SET.ERR is '1'.

Writing a '0' to these bits has no effect.

Writing a '1' to these bits clears the corresponding INTFLAGC interrupt flag.

Table 16-9. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

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

Bit 0 – SQI Serial Quad Interface

16.6.9. Peripheral Write Protection Status A

Name: STATUSA
Offset: 0x34
Reset: 0x00000000
Property: -

Reading the STATUSA register returns the peripheral write protection status of the indicated peripherals:

0: Peripheral is not write protected.

1: Peripheral is write protected.

Table 16-10. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
	TRAM	PAC	EIC	RTC	WDT	FREQM	MCLK	GCLK
Access	R/W	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
	OSC32KCTRL	OSCCTRL	RSTC	SUPC	PM	FCW	FCR	DSU
Access	R/W	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 – TRAM Trust RAM

Bit 14 – PAC Peripheral Access Controller

Bit 13 – EIC External Interrupt Controller

Bit 12 – RTC Real Time Clock

Bit 11 – WDT Watchdog Timer

Bit 10 – FREQM Frequency Meter

Bit 9 – MCLK Main Clock

Bit 8 – GCLK Generic Clock Controller

Bit 7 – OSC32KCTRL 32 K Oscillator Controller

Bit 6 – OSCCTRL Oscillator Controller

Bit 5 – RSTC Reset Controller

Bit 4 – SUPC Startup Power Controller

Bit 3 – PM Interrupt Flag for the Power Manager

Bit 2 – FCW Flash Controller Write

Bit 1 – FCR Flash Controller Read

Bit 0 – DSU Device Service Unit

16.6.10. Peripheral Write Protection Status B

Name: STATUSB
Offset: 0x38
Reset: 0x00000000
Property: -

Reading the STATUSB register returns the peripheral write protection status of the indicated peripherals:

0 Peripheral is not write protected.

1 Peripheral is write protected.

Table 16-11. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
			TCC3	TCC2	TCC1	TCC0	SERCOM3	SERCOM2
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
	SERCOM1	SERCOM0	EVSYS	IDAU	PRM	DMA1	DMA0	PORT
Access	R/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 10, 11, 12, 13 – TCCn Interrupt Flag for Timer Counter Controller, n = 0..3

Bits 6, 7, 8, 9 – SERCOMn Serial Communication Interface n, n = 0..3

Bit 5 – EVSYS Event System

Bit 4 – IDAU Implementation Defined Attribution Unit

Bit 3 – PRM Host Boot ROM

Bits 1, 2 – DMAn n = 0..1

Bit 0 – PORT PORT General Purpose Pin I/O Controller

16.6.11. Peripheral Write Protection Status C

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

Reading the STATUSC register returns the peripheral write protection status of the indicated peripherals:

0 Peripheral is not write protected.

1 Peripheral is write protected.

Table 16-12. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
							USB	TRNG
Access							R/W	R/W
Reset							0	0

Bit	15	14	13	12	11	10	9	8
	ETH	PDEC	CCL		SPI_IXS	PTC	AC	ADC
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
	TCC7	TCC6	TCC5	TCC4	SERCOM7	SERCOM6	SERCOM5	SERCOM4
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 17 – USB Universal Serial Bus

Bit 16 – TRNG True Random Number Generator

Bit 15 – ETH Ethernet

Bit 14 – PDEC Position Decoder

Bit 13 – CCL Configurable Custom Logic

Bit 11 – SPI_IXS Inter-IC Sound Interface

Bit 10 – PTC Peripheral Touch Controller

Bit 9 – AC Analog Comparator

Bit 8 – ADC Analog-to-Digital Converter

Bits 4, 5, 6, 7 – TCCn Timer Counter Controller n, n = 4..7

Bits 0, 1, 2, 3 – SERCOMn Serial Communication Interface n, n = 4..7

16.6.12. Peripheral Write Protection Status and Clear D

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

Reading the STATUSD register returns the peripheral write protection status of the indicated peripherals:

0: Peripheral is not write protected.

1: Peripheral is write protected.

Table 16-13. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

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

Bit 0 – SQI Serial Quad Interface

17. Device Service Unit (DSU)

17.1. 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 as well as identification of other debug components within the system. Therefore, it complies with the Arm Peripheral Identification specification.

The DSU also provides system services to applications that need memory testing, as required for IEC60730 Class B compliance, for example. The DSU can be accessed simultaneously by a debugger and the CPU, as it is connected on the High-Speed Bus Matrix.

The DSU implements communication channels between the device and external tools which can be used at boot time to make use of Boot ROM services.

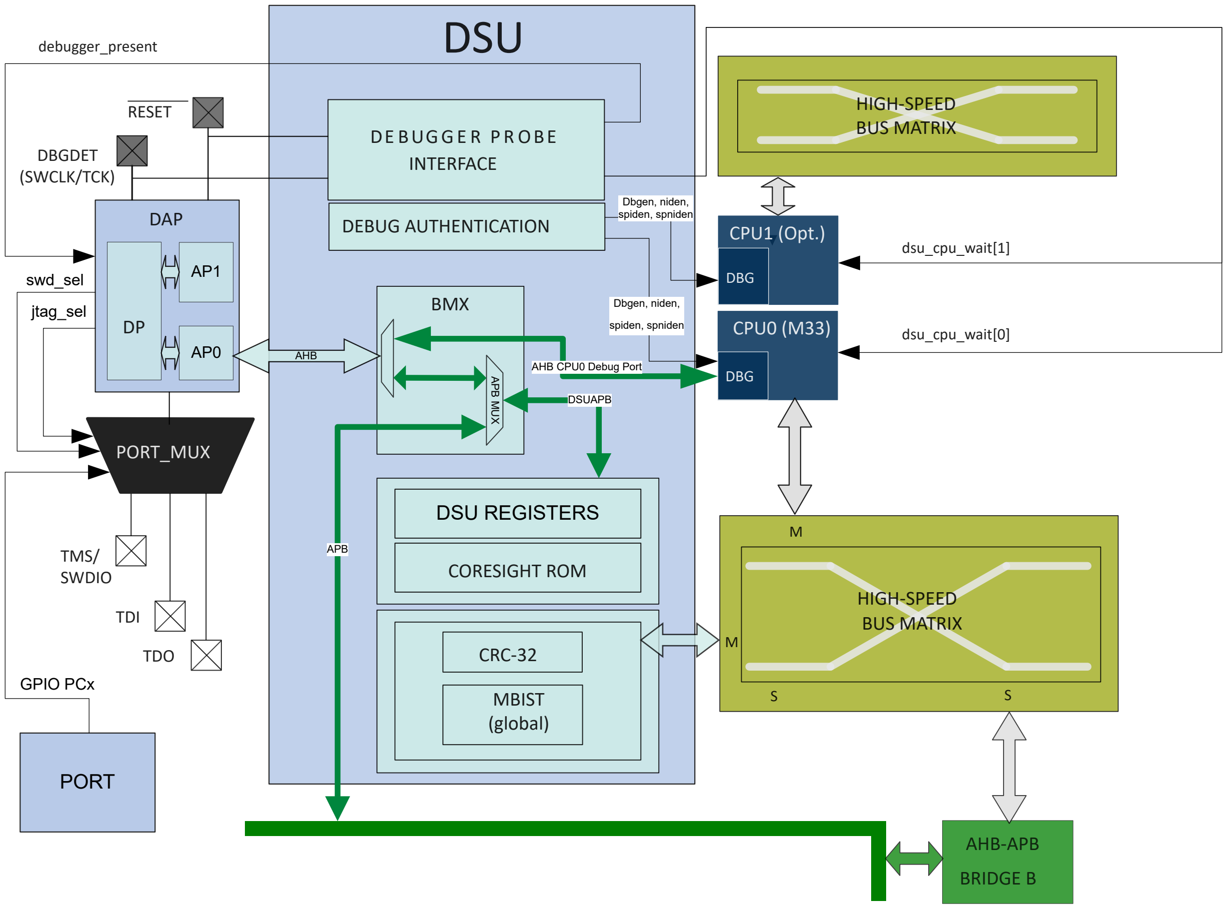
17.2. Features

The following are key features of the DSU module:

- Arm® CoreSight™ compliant device identification
- Microchip device identification
- CPU reset extension
- Debugger probe detection (Cold-Plugging and Hot-Plugging)
- Two Debug Communication Channels (DCC[0,1] with DMA signaling)
- Two Boot Communication Channels (BCC[0,1])
- Supports Debug Authentication and non-secure debug
- Support for M33 CPU
- Onboard (global) memory built-in self-test (MBIST)
- 32-bit cyclic redundancy check (CRC32)

17.3. PIC32CK SG/GC DSU Block Diagram

Figure 17-1. DSU Block Diagram



17.4. Signal Description

The DSU uses these signals to function.

Signal Name	Type	Description
$\overline{\text{RESET}}$	Digital Input	External Reset
SWCLK	Digital Input	Serial Wire Debug (SWD) clock
SWDIO	Digital I/O	Serial Wire Debug (SWD) bidirectional data pin
TCK	Digital Input	JTAG Test Clock
TMS	Digital Input	JTAG Test Mode Select
TDI	Digital Input	JTAG Test Data In
TDO	Digital Output	JTAG Test Data Out

Notes:

1. By default the Debug Port (DP) starts in JTAG mode after Power-on Reset, but it can switch to Single Wire Debug (SWD) mode using the Arm JTAG to SWD switching sequence.
2. Any I/O pins being used for SWD/JTAG functions should not be configured for Open Drain operation.

17.5. Peripheral Dependencies

Peripheral Name	Base Address	MCLK AHBx/APBx Clock Enable Bit	PAC Peripheral ID (PAC.WRCTRL.PERIDx)	DMA Trigger Index:Source (DMAC.CHCTRLBk.TRIGx)	Power Domain
DSU	0X4400 0000	AHB: MCLK.CLKMSK0[3] APB: MCLK.CLKMSK1[0]	0	2 : DCC0 3 : DCC1	VDDREG

17.6. Indexing

The DSU can service up to two CPUs: CPU0 is the device's M33 core. CPU1 is for an optional coprocessor.

17.7. Debug Operation

17.7.1. Principle of Operation

The DSU monitors the presence of a debugger to control appropriately on-chip resources when a debugger is attached to the device. The following are two attachment methods with different properties discussed later:

- Cold-plugging: Attachment when external reset is asserted.
- Hot-plugging: Attachment when external reset is not asserted.

A debugger presence has several effects on the device:

- Assigns JTAG or SWD pads to the Debug Port (DP) by controlling the port mux
- Enables Access Ports
- Extends the device CPUs reset phases

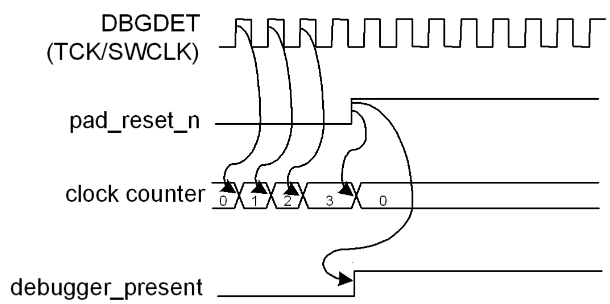
The DSU also implements Boot Communication Channels to enable communication with the device’s Boot ROM. For additional information on the Arm debug components, refer to the Arm “Debug Interface v5 Architecture Specification” document.

17.7.2. Debugger Probe Detection

17.7.2.1. Cold Plugging

Cold-Plugging is the detection of a debugger when the external reset is asserted ($\overline{\text{RESET}}$ pulled low). As shown in the following figure, a cold-plugging is detected when at least 3 TCK/SWCLK pin rising edges are detected while $\overline{\text{RESET}}$ is asserted. The detector state is updated upon a $\overline{\text{RESET}}$ pin rising edge. The Cold-Plugging detection is reset by a power-on reset or external reset. At startup, the TCK/SWCLK pin must be externally pulled up by 33 kΩ resistors to avoid false detection of a debugger when this pin is left unconnected on the application board. Cold-plugging detection is available once pads are correctly powered after a power-reset. If DAL.CPU0 equals 0 or 3, then Cold-Plugging is the only way to detect a debugger probe, and so the external reset timing must be longer than the power-reset timing. If external reset is de-asserted before power-reset release, the user must retry the cold-plugging procedure until it gets connected to the device.

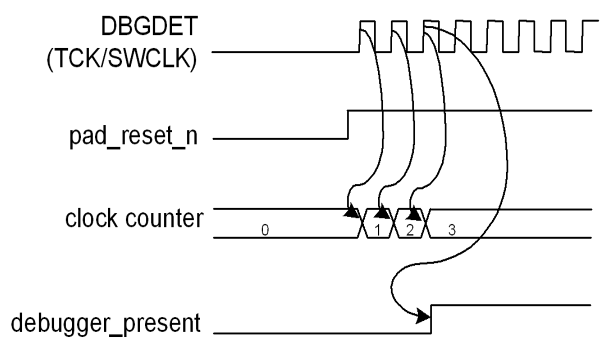
Figure 17-2. Cold-Plugging Timing Diagram



17.7.2.2. Hot Plugging

Hot-Plugging is the detection of a debugger probe when the external reset pin is not asserted. As shown in the following figure, Hot-Plugging is detected when at least 3 TCK/SWCLK pin rising edges are detected while $\overline{\text{RESET}}$ is not asserted. The hot-plugging detector is reset by a power-reset or a an external reset. The module assumes that the TCK/SWCLK function is controlled by the PORT and defaults to the Debug Port (DP) when any reset asserts. Any change to the PORT TCK/SWCLK pad configuration disables the hot-plugging detection immediately until a power-reset or external reset occurs. Therefore no debugger can connect after such configuration. Moreover, for security reasons, Hot-Plugging is not available when DAL.CPU0 equals to 0 or 3. Availability of the Hot-Plugging feature can be read from the Hot-Plugging Enable bit of the Status B register (STATUSB.HPE).

Figure 17-3. Hot-Plugging Detection Timing Diagram



17.7.2.3. Debugger Probe Detection Conditions and Effects

The presence of a debugger probe is detected when either Hot-Plugging or Cold-Plugging is detected .

Effects of a debugger detection are:

- PORT MUX JTAG or SWD functions are selected depending on the Debug Port (DP) selected protocol. These functions have a higher priority than the GPIO function controlled by the PORT module. User code cannot reclaim the pads that are claimed by the DP from that point on by configuring the PORT. The number of pads claimed by the DP depends on the DP selected protocol (JTAG or SWD). The DP starts in JTAG mode after a power-reset (TCK, TMS, TDI, TDO are claimed in this case) but can switch to SWD mode using the Arm JTAG to SWD switching sequence (the SWCLK and SWDIO pads only are claimed by the DP in this case).
- Access ports are enabled. Trying to access an access port register from the DP when it is disabled returns a DP fault (sets the DP sticky error bit).
- The Debugger Present bit of the Status B register (STATUSB.DBGPRES) reads one
- Extends the CPU reset (only with cold-plugging)

Note: Once in SWD mode it is possible to switch back to JTAG mode using the ARM SWD to JTAG switching sequence.

17.7.3. CPU Reset Extension

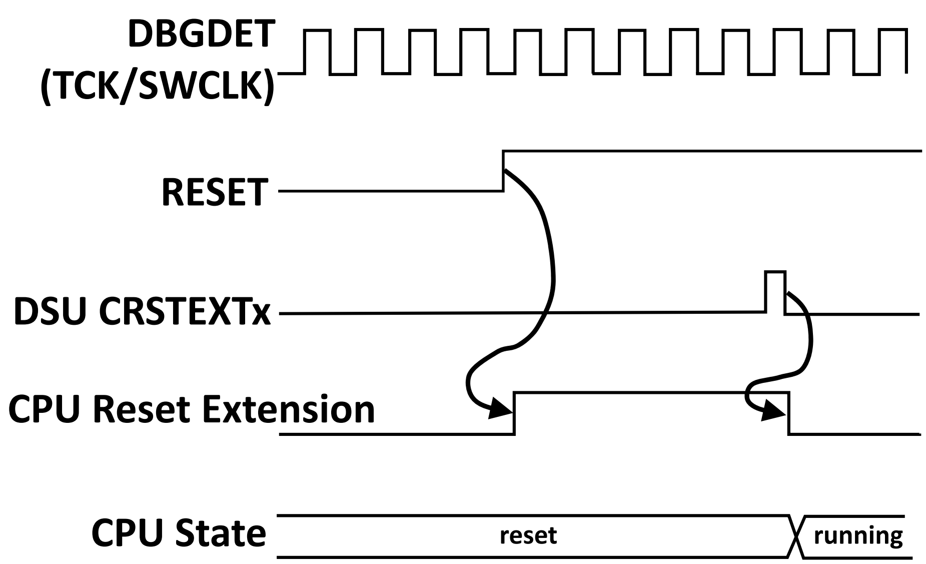
"CPU reset extension" refers to the extension of the reset phase of the device's CPU core(s) when a debugger connects with the cold-plugging procedure. This ensures that when a debugger connects:

- The CPU is not executing code at startup
- The device is in a safe and in a known state (reset)

For security reasons, the hot-plugging method is not allowed when CPU0.DAL=0. Cold-plugging must be used in this case.

For each CPU held in the reset extension phase, the CPU Reset Extension bits of the Status A register (STATUSA.CRSTEXTx) reads one. To release CPUx from reset, write a '1' in STATUSA.CRSTEXTx. STATUSA.CRSTEXTx will then be cleared. More than one CPU can be released at a time. (Writing a '0' to STATUSA.CRSTEXTx has no effect.) Releasing the "CPU reset extension" is possible for all DAL.CPUx levels. Each CPU then executes its Boot ROM. The host CPU (CPU 0) Boot ROM implements security checks at startup. It is not possible to access the bus system until the Boot ROM has performed these security checks.

Figure 17-4. Typical CPU Reset Extension Set and Clear Timing Diagram



17.7.4. Boot Communication Channels

Boot Communication Channels allow communication between a debug adapter and the Host CPU executing the Host Boot ROM at startup. The Host Boot ROM implements system level commands.

17.8. Multi-Processor Support

The DSU can support 2 CPU cores, referred to as CPU0, CPU1. Each CPU connects a different bus system. The DAP implements one MEMory Access Port (MEM-AP) per CPU and allows debug tools to access each CPU's bus system. Debug tools must select the correct MEM-AP at the debug port level depending on the targeted bus system.

- MEM-AP0 is disabled (by decreasing order of priority) when:
  - CPU0.DAL = 0 and no cold-plugging detected.

Once enabled, MEM-AP0 always has access to the DSU external space to enable communications between the device and external tools.

- MEM-AP1 is disabled when:
  - When STATUSB.DBGPRES = 0
  - when their respective DAL.CPU1 level is not equal to 2.

17.9. Programming

For each bus system, programming the Flash or RAM memories is only possible when the debugger access level is sufficient to access the desired resource. If MEM-APx is selected by the DP and if DAL.CPUx is equal to:

- 0x2: debugger can access all areas in the CPU bus system.
- 0x0: If the debugger targets the MEM-AP0 then it can only access the DSU external address space making it possible to communicate with the Boot ROM after reset. If debugger targets the MEM-AP1 then all transactions results in a DAP fault.

A typical programming procedure, when DAL.CPUx=0x2, is presented below:

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 all 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 generates at least 3 SWCLK clock cycles while $\overline{\text{RESET}}$ is asserted. $\overline{\text{RESET}}$ is then 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. If x = 0 go to 6 otherwise go to 10.
6. CPU 0 executes its Boot ROM.
7. It is recommended to issue a chip-erase (supported by the Boot ROM) to ensure that the Flash is fully erased prior to programming.
8. If the operation issued above was accepted and has completed successfully then DAL.CPU0 equals 0x2 therefore programming is available through the MEM-AP 0.
9. 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 clock is stopped while asserting $\overline{\text{RESET}}$ to prevent entering again the cold-plugging procedure.
   End of procedure for CPU 0.
10. CPU1 executes its Boot ROM, DAL.CPU1 is updated during this process.
11. If DAL.CPU1 is locked, unlock it using Boot ROM1 challenge/response features.
12. If DAL.CPU1 equals 0x2 then programming is available through the MEM-AP 1.
13. After the operation is completed, the chip can be restarted either by asserting $\overline{\text{RESET}}$, toggling power, or sending a command to the Host 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.
    End of procedure for CPU 1.

17.10. Security Enforcement

Security enforcement aims at protecting intellectual property, it consists of the following:

- Restricting access to access ports depending on the debugger access level
- Restricting access to internal memories from external tools depending on the debugger access level

The security at the Debug Access Port x level is enforced by setting the Debugger Access Level x bits in the DAL register (DAL.CPUx) to a value lower than 0x2. The DAL.CPUx setting can be elevated using Boot ROM commands depending on the Boot ROM user configuration, refer to chip-erase and Challenge/Response features in the “Boot ROM” chapter. When DAL.CPU0 is equal to 0 or 3, read/write accesses using the MEM-AP0 are limited to the DSU external address range and DSU commands are restricted. When issuing a (Host) Boot ROM Chip-Erase, sensitive Host information is erased from volatile memory and Flash. For more information about the (Host) Boot ROM features, such as the chip-erase, refer to the Boot ROM chapter. When x>0 and DAL.CPUx is equal to 0 all accesses to the MEM-APx are disabled.

The DSU implements an internal bus matrix which routes all AHB-AP0 accesses directed to the DSU address space directly to the DSU APB interface. Other access are routed to the CPU0 debug port, refer to the DSU Block Diagram).

The DSU also implements a Debug Authentication module that controls each AP and CPU debug feature depending on the CPU’s DAL level. When DAL.CPUx = 0, the CPUx AHB debug port doesn’t accept debug transactions, any MEM-APx transaction returns a bus error which translates to an ARM DP sticky error bit (refer to the Arm Debug Interface v5 Architecture Specification on www.arm.com).

The DSU APB interface address space is divided as follows:

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

Transactions initiated by a debug adapter are denoted as external transactions. All transactions directed to the CPU0 bus system go through the DSU Bus Matrix (BMX) which:

- Allows access to the full address space when DAL>0
- Restricts accesses to the DSU external space when DAL= 0

Figure 17-5. APB Memory Mapping

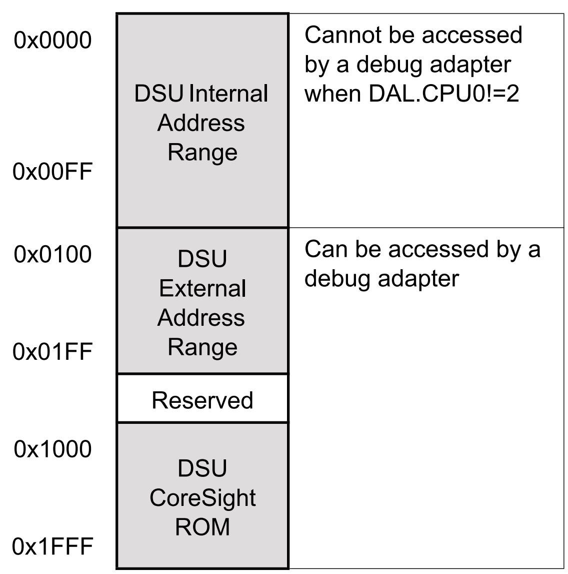


Table 17-1. MEM-AP0 Access Rights Depending on DAL.CPU0

Regions	DAL.CPU0			
	0 [3]	1	2	3
PPB or IOBUS	N	Y[1]	Y	N
DSU internal address space	N	N[2]	Y	N
DSU external address space	Y	Y	Y	Y
Other	N	Y	Y	N

Notes:

1. Refer to Arm v8m debug documentation for detailed information on PPB and IOBUS access restrictions.
2. When DAL.CPU0 = 1 DAP transfers are always non-secure. The internal address space is accessible only by secure hosts. DAP transactions will propagate to the DSU APB interface but will be ignored and STATUSA.PERR will be set.
3. MEM-AP0 is disabled until cold-plugging occurs.

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

Table 17-2. Feature Availability Under Protection

Features	DAL.CPU0			
	0	1	2	3
CPU Reset Extension	Y	Y	Y	Y
Clear CPU Reset Extension	Y	Y	Y	Y
Debugger Cold-Plugging	Y	Y	Y	Y
Debugger Hot-Plugging	N	Y	Y	N

17.11. Device Identification

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

17.11.1. 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 17-6. Conceptual 64-bit Peripheral ID

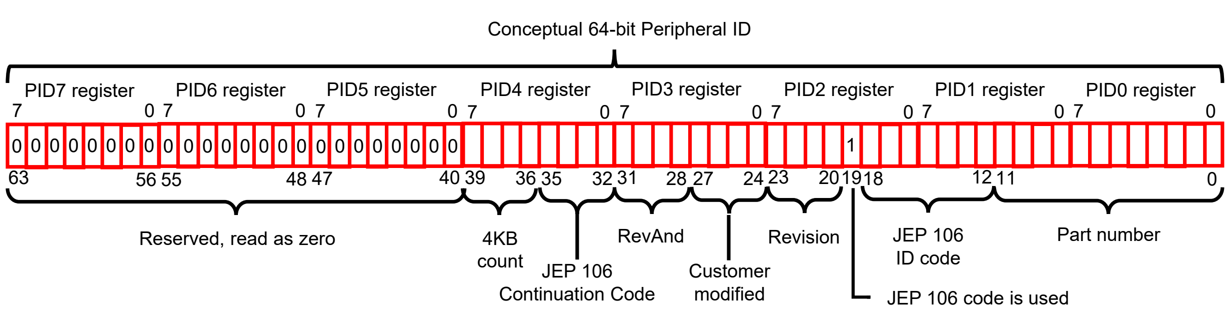


Table 17-3. Conceptual 64-Bit Peripheral ID Bit Descriptions

Field	Size	Description	Location
JEP-106 CC code	4	Microchip Continuation code: 0x0	PID4

Table 17-3. Conceptual 64-Bit Peripheral ID Bit Descriptions (continued)

Field	Size	Description	Location
JEP-106 ID code	7	Device ID: 0x29	PID1+PID2
4KB count	4	Indicates that the CoreSight component is a ROM: 0x0	PID4
Rev And	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. this indicates that device identification can be completed by reading the Device Identification register (DID).	PID0+PID1
REVISION	4	DSU revision (starts at 0x0 and increments by 1 at both major and minor revisions). Identifies DSU identification method variants.	PID2

For additional information, refer to the “Arm Debug Interface Version 5 Architecture Specification”.

17.11.2. Chip Identification Method

The DSU DID register identifies the device by implementing the following information:

- Processor identification
- Device select

17.12. Functional Description

17.12.1. Principle of Operation

The DSU provides memory services, such as CRC32 or MBIST that require almost the same interface. Therefore, 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 CTRL control register. When a command is ongoing, other commands are discarded until the current operation is completed. STATUSA.FAIL is set when commands are discarded. The recommended procedure to issue a command is given below:

1. Clear STATUSA.DONE (by writing a 1 into STATUSA.DONE).
2. Issue a command (by writing a valid command into CTRL.CMD).
3. Wait until STATUSA.DONE is set.

17.12.2. Basic Operation

17.12.2.1. Initialization

The module is enabled by enabling its clocks. For more details, refer to the Peripheral Dependencies section and MCLK chapter. The DSU registers can be PAC write-protected.

For additional information, refer to the PAC - Peripheral Access Controller.

17.12.2.2. Operation From a Debug Adapter

Debug adapters should access the DSU registers in the external address range [0x0100 – 0x1FFF].

If DAL.CPU0 is equal to 0 or 3, accessing the first 0x100 bytes causes the DSU bus matrix to return an error to the DAP. See the following table for information.

Table 17-4. DAP MEM AP0 Transaction Authorizations and Error Response Types

DBG Connection	DAL.CPU0	DAP transaction allowed?		
		DSU internal address space	DSU external address space	Other
Cold-p	0,3	N (BE)	Y	N (BE)
Hot-p	0,3	N (APD)		
any	2	Y		

BE: A Bus Error is sent back to the DAP setting its sticky bit error.

APD: Access port disabled. The transaction is discarded, the Debug Port (DP) sticky bit error is set.

Note: Refer to the *ARM Debug Interface Architecture Specification* for details.

17.12.3. 32-bit Cyclic Redundancy Check CRC32

The DSU unit provides support for calculating a cyclic redundancy check (CRC32) value for a memory area (including Flash and AHB RAM). The algorithm employed is the industry standard CRC32 algorithm using the generator polynomial 0xEDB88320 (reversed representation).

17.12.3.1. Starting CRC32 Calculation

CRC32 calculation for a memory range is started after writing the start address into the Address register (ADDR) and the size of the memory range into the Length register (LENGTH). Both must be word-aligned. The minimum LENGTH register value is 0x4, if LENGTH is 0 the command completes immediately.

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.

The actual test is started by writing the 32-bit Cyclic Redundancy Code (CRC32) command into the Control register command bit field (CTRL.CMD). A running CRC32 operation can be canceled by resetting the module (writing '1' to CTRL.SWRST).

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.

17.12.3.2. Interpreting the Results

The user should monitor the STATUSA 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.

17.12.4. Debug Communication Channels

The Debug Communication Channels (DCC0 and DCC1) consist of a pair of registers with associated handshake logic, accessible by both CPU and debugger with no security restriction. These registers are used to exchange data between the device and a debugger. 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. They are used to communicate with the Boot ROM after a cold-plugging procedure. For more information, refer to the Host “Boot ROM”.

Two Debug Communication Channel status bits in the STATUSB register (STATUSB.DCCDx) indicate that a new value has been written into DCC0 or DCC1. These bits, DCC0D and DCC1D, are located in the STATUSB register. They are automatically set by hardware and cleared on read.

Note: The DCC0 and DCC1 registers are shared with the Boot Communication Channels (BCC0 and BCC1) registers, therefore mixing DCC and BCC communication is not recommended.

17.12.5. 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. These registers are intended to communicate with the CPU while executing the Boot ROM, which implements security and failure analysis commands, and therefore these registers must not be used for another purpose.

Two Boot Communication Channel status bits in the Status B registers (STATUSB.BCCDx) indicate whether a new value has been written in BCC0 or BCC1. These bits, BCC0D and BCC1D, 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.

17.12.6. Testing of On-Board Memories MBIST (Global Memory BIST)

The DSU implements a feature for automatic testing of memory also known as Global Memory Built-In Self Test (Global MBIST). Only memory mapped memories (accessible from the DSU AHB Host) can be tested.

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.

17.12.6.1. Algorithm

The MBIST algorithm is provided in the following table:

Table 17-5. MBIST Algorithm

Mnemonic(STATUSC.STATE)	Phase	Iteration	Steps
MBIST_FILL	Write all bits to zero (32-bit AHB writes). No MBIST fault is detected in this phase.	For all words (ascending)	LENGTH
MBIST_SET1	32-bit read at address ADDR. Check that bit INDEX is ‘0’. 32-bit write at address ADDR writes previously read data with bit INDEX ‘1’.	For all words (descending)For all bits (descending)	LENGTH*32
MBIST_SET2	32-bit read at address ADDR. Check that bit INDEX is ‘1’. 32-bit write at address ADDR writes previously read data with bit INDEX ‘0’.	SET2, SET2B sequence: For all words (ascending) For all bits (ascending)	2*LENGTH*32
MBIST_SET2B	32-bit read at address ADDR. Check that bit INDEX is ‘0’. 32-bit write at address ADDR writes previously read data with bit INDEX ‘1’.		
MBIST_CLEAR1	32-bit read at address ADDR. Check that bit INDEX is ‘1’. 32-bit write at address ADDR writes previously read data with bit INDEX ‘0’.	For all words (ascending) For all bits (ascending)	LENGTH*32
MBIST_CLEAR2	32-bit read at address ADDR. Check that bit INDEX is ‘0’. 32-bit write at address ADDR writes previously read data with bit INDEX ‘1’.	CLEAR2, CLEAR2B sequence: For all words (ascending) For all bits (ascending)	2*LENGTH*32
MBIST_CLEAR2B	32-bit read at address ADDR. Check that bit INDEX is ‘1’. 32-bit write at address ADDR writes previously read data with bit INDEX ‘0’.		
MBIST_READ	32-bit read Check that all bits read as zero. (INDEX is not relevant in this phase.)	For all words (ascending) (pipelined reads)	LENGTH

The specific implementation used has a run time that depends on the CPU clock frequency and the number of bytes tested in the RAM. The detected faults are: as follows:

- Address decoder faults
- Stuck-at faults
- Transition faults
- Coupling faults
- Linked Coupling faults

17.12.6.2. Starting MBIST

To test a memory, write the start address of the memory to the ADDR.ADDR bit field, and the size of the memory into the LENGTH register. When LENGTH = 0 the algorithm ends immediately (STATUS.DONE = 1 without any actual check).

For best test coverage, an entire physical memory block must 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 the MBIST command into CTRL.CMD. A running MBIST operation can be canceled by writing a '1' to CTRL.SWRST.

17.12.6.3. Interpreting the Results

When the operation is completed, STATUSA.DONE is set. STATUSA.BERR indicates whether a bus error occurred, in this case the state machine returns to IDLE state immediately (and sets STATUSA.DONE). STATUSA.FAIL indicates that a fault has been detected. There are three different modes:

Table 17-6. ADDR.AMOD Bit Description

ADDR.AMOD	Description
0x0	Exit on Error
0x1	Pause on Error
0x2	Reserved
0x3	Reserved

ADDR.AMOD = 0: exit-on-error (default)

In this mode, the algorithm terminates either when a fault is detected (reported as STATUSA.FAIL) or when a bus error is detected (reported as STATUSA.BERR) or upon successful completion. STATUSA.DONE rises upon completion (with or without errors). STATUSC is copied to the DATA register upon error detection. Users must read the DATA register to locate the fault.

ADDR.AMOD = 1: pause-on-error

In this mode, the MBIST algorithm is paused when an error is detected and STATUSA.FAIL is asserted. Users must ensure STATUSA.FAIL is 0 prior to start the algorithm. Once started and once a fault has been detected the state machine waits for users to clear STATUSA.FAIL by writing a '1' into STATUSA.FAIL to resume the algorithm. Prior to resuming, users can read the STATUSC and ADDR registers to locate the fault. Bus errors stop the MBIST algorithm at any time (STATUSA.DONE and STATUSA.BERR both asserted).

17.12.6.4. Locating Faults

If the test stops with STATUSA.FAIL high, one or more bits have failed the test. The fault location depends on the selected mode and the state of the state machine when the fault was detected.

All accesses on the bus system are pipelined. In states where a read is followed by a write, the state machine stops after the write is posted on the bus at the same address, therefore the context does not change, error reporting is exact. On the other hand, the MBIST_READ state performs pipelined 32-bit reads. This means that the returned data from the bus is checked while a new read is requested on the bus. Therefore if a fault is detected, it is located on the previous word (at address ADDR-4).

Exit on error mode:

Table 17-7. Exit on Error Fault Location

DATA.STATE	Fault location
MBIST FILL	Not Applicable
MBIST SET1	address: ADDR bit: DATA.INDEX
MBIST SET2	
MBIST SET2B	
MBIST CLEAR1	
MBIST CLEAR2	
MBIST CLEAR2B	
MBIST READ	address: ADDR-4 bit: any of the 32 bits

Table 17-8. Pause on Error Fault Location

STATUSC.STATE	Fault location
MBIST_FILL	Not Applicable
MBIST_SET1	address: ADDR bit: STATUSC.INDEX
MBIST_SET2	
MBIST_SET2B	
MBIST_CLEAR1	
MBIST_CLEAR2	
MBIST_CLEAR2B	
MBIST_READ	address: ADDR-4 bit: any of the 32 bits

17.12.6.5. Fault Injection

This feature supports verifying that any ‘stuck-at’ fault can be caught by the MARCH-LR state machine and therefore is not supposed to be used very often. To minimize the gate-count of this feature, MBFI0 is an alias to the shared register used by DCC0 and BCC0, MBFI1 is an alias to the shared register used by DCC1 and BCC1. Therefore the fault injection mechanism should be used with extreme care if intended to operate along with data and boot communication channels. The MBFI0 and MBFI1 registers are write-protected when CFG.MBFI is 0. As these registers are not asynchronously reset it is recommended to clear them before attempting any MBIST operation with fault injection enabled.

Faults are injected in the data received by the MBIST state machine coming out from the DSU AHB Host when CFG.MBFI is high and when there is an address match between the address configuration held in MBFI0 and MBFI1 and the address presented by the DSU AHB Host.

There is an address match when:

- MBFI0.AMMOD = 1(ALWAYS) (all addresses match in this case)
- When ((DSU AHB Host byte address[31:2]) and ~{6'h00,MBFI0.AMMSK} = MBFI1.ADDR and ~{6'h00,MBFI0.AMMSK})

Each ONE written at position x of AMMSK indicates that the byte address bit x+2 generated by the DSU AHB host during MBIST operation matches (MBFI0.AMMSK and MBFI1.ADDR are word addresses, not Byte addresses).

Only a single fault can be injected at a time. The type of fault is configured by the MBFI0.FTYPE which is either STUCKAT0 (0) or STUCKAT1 (1). MBFI0.BIDX indicates the bit position of the fault.

17.12.7. Memory Set

The Memory Set feature initializes a memory range by writing LENGTH 32-bit words starting at address ADDR. The 32-bit pattern used must be stored in the DATA register prior to issuing the MSET command. The MSET command is started by writing the MSET command into CTRL.CMD. The values of ADDR and LENGTH must be word-aligned. The minimum LENGTH register value is 0x4, if LENGTH is 0 the command completes immediately. Writes are pipelined, therefore the transaction lasts LENGTH+1 cycles if the client responds with 0 wait state.

17.13. Register Summary

For descriptions and definitions of both Register and bitfield properties, refer to Register Properties.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRL	31:24	CMD[15:8]							
		23:16	CMD[7:0]							
		15:8								
		7:0								SWRST
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	CFG	31:24								
		23:16								
		15:8								
		7:0						MBFI	DCCDMALEVEL1	DCCDMALEVEL0
0x14	MBFI0	31:24	AMMSK[23:16]							
		23:16	AMMSK[15:8]							
		15:8	AMMSK[7:0]							
		7:0	FTYPE	AMMOD		BIDX[4:0]				
0x18	MBFI1	31:24	ADDR[29:22]							
		23:16	ADDR[21:14]							
		15:8	ADDR[13:6]							
		7:0	ADDR[5:0]							
0x1C ... 0xFF	Reserved									
0x0100	STATUSA	31:24								
		23:16							BREXT1	BREXT0
		15:8							CRSTEXT1	CRSTEXT0
		7:0					PERR	BERR	FAIL	DONE
0x0104	STATUSB	31:24								
		23:16								
		15:8						HPE		DBGPRES
		7:0					DCCD1	DCCD0	BCCD1	BCCD0
0x0108	STATUSC	31:24								
		23:16								
		15:8				INDEX[4:0]				
		7:0				STATE[4:0]				
0x010C ... 0x010F	Reserved									
0x0110	BCC0	31:24	DATA[31:24]							
		23:16	DATA[23:16]							
		15:8	DATA[15:8]							
		7:0	DATA[7:0]							
0x0114	BCC1	31:24	DATA[31:24]							
		23:16	DATA[23:16]							
		15:8	DATA[15:8]							
		7:0	DATA[7:0]							

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0118	DCC0	31:24	DATA[31:24]							
		23:16	DATA[23:16]							
		15:8	DATA[15:8]							
		7:0	DATA[7:0]							
0x011C	DCC1	31:24	DATA[31:24]							
		23:16	DATA[23:16]							
		15:8	DATA[15:8]							
		7:0	DATA[7:0]							
0x0120	DID	31:24	REVISION[3:0]				PRODUCT[7:4]			
		23:16	PRODUCT[3:0]				DEVSEL[7:4]			
		15:8	DEVSEL[3:0]				MANID[10:7]			
		7:0	MANID[6:0]							MARKER
0x0124	DAL	31:24								
		23:16								
		15:8								
		7:0					CPU1[1:0]		CPU0[1:0]	
0x0128 ... 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	ADDOFF[19:12]							
		23:16	ADDOFF[11:4]							
		15:8	ADDOFF[3:0]							
		7:0							FMT	EPRES
0x1008	ENTRY2	31:24	ADDOFF[19:12]							
		23:16	ADDOFF[11:4]							
		15:8	ADDOFF[3:0]							
		7:0							FMT	EPRES
0x100C	ENTRY3	31:24	ADDOFF[19:12]							
		23:16	ADDOFF[11:4]							
		15:8	ADDOFF[3:0]							
		7:0							FMT	EPRES
0x1010	ENTRY4	31:24	ADDOFF[19:12]							
		23:16	ADDOFF[11:4]							
		15:8	ADDOFF[3:0]							
		7:0							FMT	EPRES
0x1014	ENTRY5	31:24	ADDOFF[19:12]							
		23:16	ADDOFF[11:4]							
		15:8	ADDOFF[3:0]							
		7:0							FMT	EPRES
0x1018	ENTRY6	31:24	ADDOFF[19:12]							
		23:16	ADDOFF[11:4]							
		15:8	ADDOFF[3:0]							
		7:0							FMT	EPRES
0x101C	ENTRY7	31:24	ADDOFF[19:12]							
		23:16	ADDOFF[11:4]							
		15:8	ADDOFF[3:0]							
		7:0							FMT	EPRES
0x1020 ... 0x1FCB	Reserved									
0x1FCC	MEMTYPE	31:24								
		23:16								
		15:8								
		7:0								SMEMP

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x1FD0	PID4	31:24								
		23:16								
		15:8								
		7:0	FKBC[3:0]				JEPCC[3:0]			
0x1FD4	PID5	31:24								
		23:16								
		15:8								
		7:0								
0x1FD8	PID6	31:24								
		23:16								
		15:8								
		7:0								
0x1FDC	PID7	31:24								
		23:16								
		15:8								
		7:0								
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]							

17.13.1. Control

Name: CTRL
Offset: 0x0000
Reset: 0x00000000
Property: PAC Write-Protection

Table 17-9. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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

Bit	23	22	21	20	19	18	17	16
	CMD[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

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

Bits 31:16 – CMD[15:0] Command Register

`0x0` = No operation (NOOP)
`0x1-0xA4FF` = Reserved
`0xA500` = 32-bit Cyclic Redundancy Code (CRC32)
`0xA501` = Global Memory built-in self-test (MBIST)
`0xA502` = Memory Set (MSET)
`0xA503-0xFFFF` = Reserved

Note: Invalid commands are reported into STATUSA.PERR. Partial writes of the CMD bit field are ignored and reported in STATUSA.PERR.

Bit 0 – SWRST Software Reset

Software reset has the highest priority. If this bit is set in a write all other bits are ignored.

17.13.2. Address

Name: ADDR
Offset: 0x0004
Reset: 0x00000000
Property: PAC Write-Protection

Table 17-10. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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 or next word address to test during MBIST operation.

Bits 1:0 – AMOD[1:0] Access Mode
These bits are only available in MBIST mode operation mode, they are reserved otherwise.
0x0 = STATUSA.FAIL rises upon first error and algorithm stops (STATUSA.DONE rises). (EXIT_ON_ERROR)
0x1 = STATUSA.FAIL rises when an error is detected and algorithm stops until STATUSA.FAIL is cleared. Once cleared, the algorithm is resumed going to next test step. (PAUSE_ON_ERROR).
0x2-0x3 = Reserved

17.13.3. Length

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

Table 17-11. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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.

17.13.4. Data

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

For the MBIST mode register format see the DATA Register Summary in MBIST Mode.

Table 17-12. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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.

17.13.5. Configuration

Name: CFG
Offset: 0x0010
Reset: 0x0000000
Property: PAC Write-Protection

Table 17-13. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						MBFI	DCCDMALEVEL1	DCCDMALEVEL0
Access						R/W	R/W	R/W
Reset						0	0	0

Bit 2 – MBFI Enable Memory BIST Fault Injection
0x0 = MBFI0 and MBFI1 registers are write-protected.
0x1 = MBFI0 and MBFI1 registers can be written. Fault injection is enabled during MBIST operation.

Bits 0, 1 – DCCDMALEVELx DMA Trigger x Level [x=1..0]

Value	Name	Description
0	EMPTY	Trigger x rises when DCC is read and falls when it is written.
1	FULL	Trigger x rises when DCC is written and falls when it is read.

17.13.6. Memory BIST Fault Injection 0

Name: MBFI0
Offset: 0x0014
Reset: 0x00000000
Property: PAC Write Protection

Notes:
1. MBFI0, BCC[0] and DCC[0] are aliases that use the same internal 32-bit register.
2. This register is write-protected when CFG.MBFI = 0.
3. This register is not asynchronously reset, therefore it must be properly initialized before starting any MBIST operation.

Table 17-14. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	AMMSK[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	23	22	21	20	19	18	17	16
	AMMSK[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	15	14	13	12	11	10	9	8
	AMMSK[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

Bit	7	6	5	4	3	2	1	0
	FTYPE	AMMOD		BIDX[4:0]				
Access	R/W	R/W		R/W	R/W	R/W	R/W	R/W
Reset	x	x		x	x	x	x	x

Bits 31:8 – AMMSK[23:0] Address Matching Mask
Address matching address mask. Each ‘1’ at position x indicates that the byte address bit x+2 generated by the DSU AHB host during MBIST operation is "Don't Care".

Bit 7 – FTYPE Fault Type
0x0 = Stuck At 0 (STUCKAT0)
0x1 = Stuck At 1 (STUCKAT1)

Bit 6 – AMMOD Address Matching Mode
0x0 = Address match, fault injected when the masked host address matches with the masked address (ADDR).
0x1 = Always matches, fault injected every AHB access (ALWAYS).

Bits 4:0 – BIDX[4:0] Bit Index of Injected Fault
Indicates the bit position of the fault to be injected in the data read by DSU AHB Host (0 to 31).

17.13.7. Memory BIST Fault Injection 1

Name: MBFI1
Offset: 0x0018
Reset: 0x00000000
Property: PAC Write Protection

Notes:
1. MBFI1, BCC[1] and DCC[1] are aliases that use the same internal 32-bit register.
2. This register is write-protected when CFG.MBFI=0.
3. This register is not asynchronously reset, therefore it must be properly initialized before any MBIST operation.

Table 17-15. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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	x	x	x	x	x	x	x	x

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	x	x	x	x	x	x	x	x

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	x	x	x	x	x	x	x	x

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

Bits 31:2 – ADDR[29:0] Word Address

A fault is injected during the AHB data phase when CFG.MBFI = 1, MBIST is operating, and one of these conditions is true:

1. MBFI0.AMMOD = 1 (ALWAYS),
   or
2. ((DSU AHB Host byte address[31:2]) & ~{6'h00,MBFI0.AMMSK} == MBFI1.ADDR & ~{6'h00,MBFI0.AMMSK}).

17.13.8. Status A

Name: STATUSA
Offset: 0x0100
Reset: 0x00000000
Property: PAC Write-Protection

Table 17-16. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
							BREXT1	BREXT0
Access							R/K	R/K
Reset							x	x

Bit	15	14	13	12	11	10	9	8
							CRSTEXT1	CRSTEXT0
Access							R/K	R/K
Reset							x	x

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

Bits 16, 17 – BREXTx Boot ROMx Phase Extension for CPUx, x =0,1
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Boot ROM Phase Extension bit.
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. Refer to the Chapter “Boot ROM” for more details. When CPUx is not present it always reads as ‘1’ .

Bits 8, 9 – CRSTEXTx CPUx Reset Phase Extension, x = 0,1
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the CPUx Reset Phase Extension bit.
This bit is set when a debug adapter Cold-Plugging is detected, which extends the CPUx reset phase. When CPUx is not present it always reads as ‘1’ .

Bit 3 – PERR Protection Error
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Protection Error bit.
This bit is set when any illegal access is detected (from a debug adapter or any host) such as an access to an unimplemented register.
This bit is set when writing an invalid command into CTRL.CMD.

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 – 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 0 – DONE Done
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Done bit.

17.13.9. Status B

Name: STATUSB
Offset: 0x0104
Reset: 0x00000000
Property: PAC Write-Protection

Table 17-17. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
						HPE		DBGPRES
Access						R		R
Reset						0		x

Bit	7	6	5	4	3	2	1	0
					DCCD1	DCCD0	BCCD1	BCCD0
Access					R	R	R	R
Reset					0	0	x	x

Bit 10 – HPE Hot-Plugging Enable
This bit is set when Hot-Plugging is enabled.
This bit is cleared when Hot-Plugging is disabled. This is the case when the TCK/SWCLK function is changed. Only a power-reset or an external reset can set it again.

Bit 8 – DBGPRES Debugger Present
This bit is set when a debugger probe is detected.
Only a POR or external reset can reset this bit.

Bits 2, 3 – DCCDx Debug Communication Channel x Dirty [x=1..0]
This bit is set when DCCx register is written.
This bit is cleared when DCCx register is read.
Reset by APB reset.

Bits 0, 1 – BCCDx BOOT Communication Channel x Dirty [x=1..0]
This bit is set when BCCx register is written.
This bit is cleared when BCCx register is read.
Reset by APB reset and modified by the Boot ROM at boot time.

17.13.10. Status C

Name: STATUSC
Offset: 0x0108
Reset: 0x00000000
Property: -

Table 17-18. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
				INDEX[4:0]				
Access				R	R	R	R	R
Reset				0	0	0	0	0

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

Bits 12:8 – INDEX[4:0] Shows MBIST bit Index

Bits 4:0 – STATE[4:0] Core State
0x0 = State Machine Ready (IDLE)
0x1 = CRC32 operation ongoing (CRC32)
0x2-0x3 = Reserved
0x4 = Memory Set (MSET)
0x5-0x7 = Reserved
0x8 = MBIST fill memory with zeroes (MBIST FILL)
0x9 = SET1 Phase: read 0'write'1'(MBIST SET1)'
0xA = SET2 Phase: read 1'write'0'(MBIST SET2)'
0xB = SET2B Phase: read 0'write'1'(MBIST SET2B)'
0xC = CLEAR1 Phase: read 1'write'0'(MBIST CLEAR1)'
0xD = CLEAR2 Phase: read 0'write'1'(MBIST CLEAR2)'
0xE = CLEAR2B Phase: read 1'write'0'(MBIST CLEAR2B)'
0xF = READ Phase: check memory is cleared (MBIST_READ)
0x10-0x1F = Reserved

17.13.11. Boot ROM Communication Channel x

Name: BCCx
Offset: 0x0110 + x*0x04 [x=0..1]
Reset: N/A
Property: -

Table 17-19. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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	x	x	x	x	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 Register

Writing BCCx sets STATUSB.BCCx.
Reading BCCx clears STATUSB.BCCx.

Notes:

1. BCC[0] and DCC[0] are aliases that use the same internal 32-bit register.
2. BCC[1] and DCC[1] are aliases that use the same internal 32-bit register.

17.13.12. Debug Communication Channel x

Name: DCCx
Offset: 0x0118 + x*0x04 [x=0..1]
Reset: 0x00000000
Property: -

Table 17-20. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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	x	x	x	x	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 Register
Writing DCCx sets STATUSB.DCCx.
Reading DCCx clears STATUSB.DCCx.
Notes:
1. BCC[0] and DCC[0] are aliases that use the same internal 32-bit register.
2. BCC[1] and DCC[1] are aliases that use the same internal 32-bit register.

17.13.13. Device Identification

Name: DID
Offset: 0x0120
Reset: Device dependent
Property: PAC Write-Protection

Table 17-21. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	REVISION[3:0]				PRODUCT[7:4]			
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
	PRODUCT[3:0]				DEVSEL[7:4]			
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
	DEVSEL[3:0]				MANID[10:7]			
Access	R	R	R	R	R	R	R	R
Reset	x	x	x	x	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	MANID[6:0]							MARKER
Access	R	R	R	R	R	R	R	R
Reset	0	1	0	1	0	0	1	1

Bits 31:28 – REVISION[3:0] Revision Number
Identifies the revision of the product.
0000 = A0

Bits 27:20 – PRODUCT[7:0] Product
The value of this field corresponds to the Product part of the ordering code.

Bits 19:12 – DEVSEL[7:0] Device Select
This bit field identifies a device within a product. Refer to the Ordering Information for device configurations and corresponding values for Flash memory density, pin count, and device variant.

Bits 11:1 – MANID[10:0] Manufacturer ID
This is duplicate of the JEP-106 CC and ID code present in the Device Service Unit Coresight ROM table that always read 0x29.

Bit 0 – MARKER Marker bit
Marker Bit Always read as '1'.

17.13.14. Debugger Access Level

Name: DAL
Offset: 0x0124
Reset: 0x00000000
Property: -

Table 17-22. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
					CPU1[1:0]		CPU0[1:0]	
Access					R	R	R	R
Reset					x	x	x	x

Bits 0:1, 2:3 – CPUx CPUx Debugger Access Level [x=0..1]
Writing in this bitfield has no effect.
0x0 = Restricts debugger access to a part of the DSU registers.
0x1 = Reserved.
0x2 = Allows the debugger to access to the whole memory map.
0x3 = Reserved.

17.13.15. Coresight ROM Table Entry x

Name: ENTRYx
Offset: 0x1000 + x*0x04 [x=0..7]
Reset: 0xXXXXX00X
Property: -

Table 17-23. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	ADDOFF[19:12]							
Access	R	R	R	R	R	R	R	R
Reset	x	x	x	x	x	x	x	x

Bit	23	22	21	20	19	18	17	16
	ADDOFF[11:4]							
Access	R	R	R	R	R	R	R	R
Reset	x	x	x	x	x	x	x	x

Bit	15	14	13	12	11	10	9	8
	ADDOFF[3:0]							
Access	R	R	R	R				
Reset	x	x	x	x				

Bit	7	6	5	4	3	2	1	0
							FMT	EPRES
Access							R	P/R
Reset							x	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' if ADDOFF is not 0, 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 when DAL.CPU0 equals 0 indicating that the entry is not present.
This bit is cleared at power-up if DAL.CPU0 is greater than 0 indicating that the entry is present.

17.13.16. Coresight ROM Table Memory Type

Name: MEMTYPE
Offset: 0x1FCC
Reset: 0x00000000
Property: -

Table 17-24. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
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 when DAL.CPU0 is greater than 0 indicating that the system memory is accessible from a debug adapter.
This bit is cleared at power-up when DAL.CPU0 is equal to 0 indicating that the system memory is not accessible from a debug adapter.

17.13.17. Coresight ROM Table Peripheral Identification 4

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

Table 17-25. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
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, indicating a Microchip device.

17.13.18. Coresight ROM Table Peripheral Identification 5

Name: PID5
Offset: 0x1FD4
Reset: 0x00000000
Property: Read-Only

The bits in this register are unimplemented and always read as '0'.

Table 17-26. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

17.13.19. Coresight ROM Table Peripheral Identification 6

Name: PID6
Offset: 0x1FD8
Reset: 0x00000000
Property: Read-Only

The bits in this register are unimplemented and always read as '0'.

Table 17-27. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

17.13.20. Coresight ROM Table Peripheral Identification 7

Name: PID7
Offset: 0x1FDC
Reset: 0x00000000
Property: Read-Only

The bits in this register are unimplemented and always read as '0'.

Table 17-28. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

17.13.21. Coresight ROM Table Peripheral Identification 0

Name: PID0
Offset: 0x1FE0
Reset: 0x00000000
Property: -

Table 17-29. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
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.

17.13.22. Coresight ROM Table Peripheral Identification 1

Name: PID1
Offset: 0x1FE4
Reset: 0x0000009C
Property: -

Table 17-30. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	JEPIDCL[3:0]				PARTNBH[3:0]			
Access	R	R	R	R	R	R	R	R
Reset	1	0	0	1	1	1	0	0

Bits 7:4 – JEPIDCL[3:0] Low Part of the JEP-106 Identity Code
These bits will always return 0x9 when read, indicating a Microchip device (Microchip JEP-106 identity code is 0x29).

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

17.13.23. Coresight ROM Table Peripheral Identification 2

Name: PID2
Offset: 0x1FE8
Reset: 0x0000002A
Property: -

Table 17-31. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
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	1	0	1	0	1	0

Bits 7:4 – REVISION[3:0] Revision Number
Revision of the peripheral. Starts at 0x0 and increments by one at both major and minor revisions.

Bit 3 – JEPU JEP-106 Identity Code is used

Bits 2:0 – JEPIDCH[2:0] JEP-106 Identity Code High
These bits will always return 0x2 when read, indicating a Microchip device (Microchip JEP-106 identity code is 0x29).

17.13.24. Coresight ROM Table Peripheral Identification 3

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

Table 17-32. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
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.

17.13.25. Coresight ROM Table Component Identification 0

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

Table 17-33. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
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 0xD when read.

17.13.26. Coresight ROM Table Component Identification 1

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

Table 17-34. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
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 www.arm.com).

Bits 3:0 – PREAMBLE[3:0] Preamble
These bits will always return 0x00 when read.

17.13.27. Coresight ROM Table Component Identification 2

Name: CID2
Offset: 0x1FF8
Reset: 0x00000000
Property: -

Table 17-35. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
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 0x5 when read.

17.13.28. Coresight ROM Table Component Identification 3

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

Table 17-36. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	PREAMBLEB3[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	1	0	1	1	0	0	0	1

Bits 7:0 – PREAMBLEB3[7:0] Preamble Byte 3
These bits will always return 0xB1 when read.

17.14. DATA Register Summary in MBIST Mode

For descriptions and definitions of both Register and bitfield properties, refer to Register Properties.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00 ... 0x0B	Reserved									
0x0C	DATA	31:24								
		23:16								
		15:8				INDEX[4:0]				
		7:0				STATE[4:0]				

17.14.1. Data

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

Table 17-37. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
				INDEX[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
				STATE[4:0]				
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bits 12:8 – INDEX[4:0] MBIST bit Index

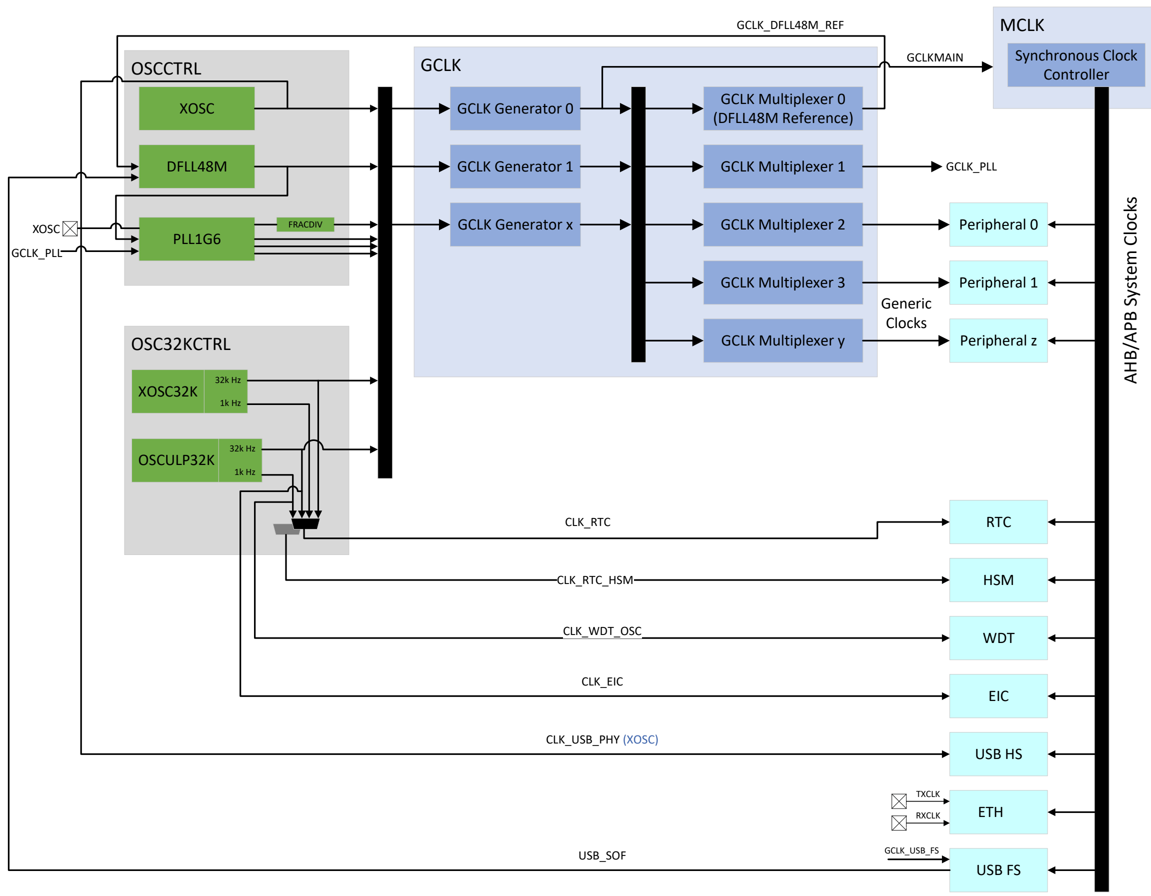
Bits 4:0 – STATE[4:0] MBIST State
0x0 = IDLE:State Machine Ready
0x1-0x7 = Reserved
0x8 = MBIST_FILL: MBIST fill memory with zeroes
0x9 = SET1 Phase: read 0'write'1'(SET1)'
0xA = SET2 Phase: read 1'write'0'(SET2)'
0xB = SET2B Phase: read 0'write'1'(SET2B)'
0xC = CLEAR1 Phase: read 1'write'0'(CLEAR1)'
0xD = CLEAR2 Phase: read 0'write'1'(CLEAR2)'
0xE = CLEAR2B Phase: read 1'write'0'(CLEAR2B)'
0xF = READ Phase: check memory is cleared (READ)
0x10-0x1F = Reserved

18. Clock Distribution System

This chapter summarizes the clock distribution and terminology in the PIC32CK SG/GC family of devices. For additional informatin refer to the peripheral chapters and the GCLK Generic Clock Controller documentation.

18.1. Clock Distribution

Figure 18-1. Clock Distribution Block Diagram



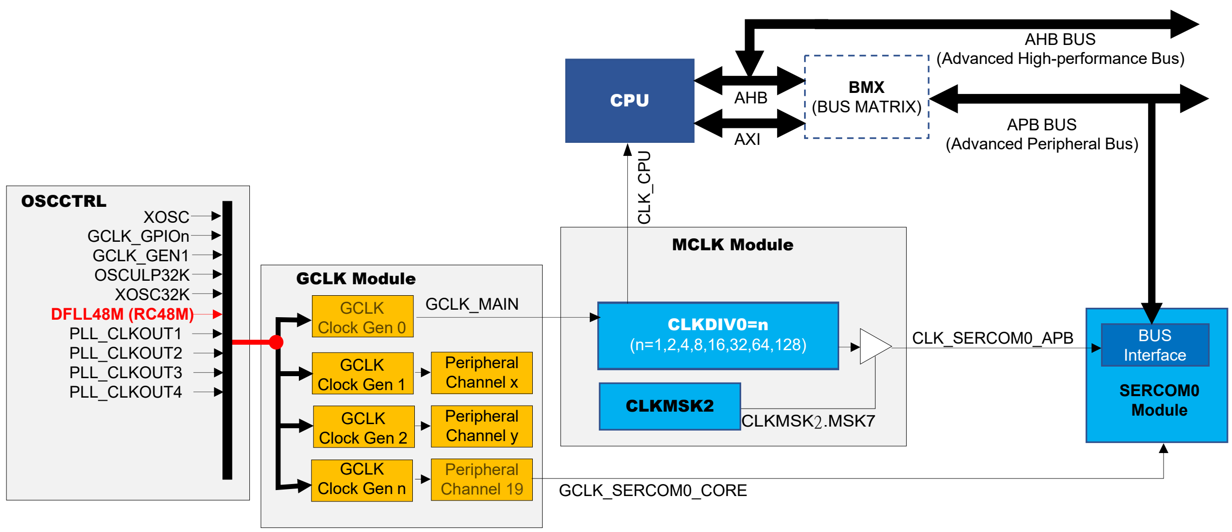
The PIC32CK SG/GC clock system features are as follows:

- Clock sources, that is oscillators controlled by OSCCTRL and OSC32KCTRL:
  - A clock source provides a time base that is used by other components, such as Generic Clock Generators. For example, clock sources are the internal 48 MHz DFLL48M, Ultra Low-power 32 kHz RC oscillator, OSCULP32K, PLL and external oscillators XOSC and XOSC32K.
- Generic Clock Controller (GCLK), which generates, controls, and distributes asynchronous clocks consisting of the following:
  - Generic Clock Generators: These have programmable prescalers that can use any of the system clock sources. The Generic Clock Generator 0 generates the clock signal GCLK_MAIN, which is used by the CPU and Data Ram Tightly Coupled Memory, which in turn generates synchronous clocks.
  - Generic Clocks: These are clock signals generated by Generic Clock Generators. They are the Peripheral Channels and serve as clocks for the peripherals of the system. Multiple instances of a peripheral will typically have a separate Generic Clock.

- Main Clock Controller (MCLK)
  - The MCLK generates and controls the synchronous clocks for the system. This includes the CPU, bus clocks (AHB (Advanced High-performance Bus), and APB (Advanced Peripheral Bus)), as well as the Special Function Register interfaces of the peripherals. It contains clock masks that can turn on/off the user interface of a peripheral as well as prescalers for the CPU and bus clocks.

The following figure illustrates an example, where SERCOM0 is clocked by the DFLL48M (RC 48 MHz) in Open-Loop mode. The DFLL48M is enabled, the Generic Clock Generator n uses the DFLL48M as its clock source and feeds into Peripheral Channel 19. The Generic Clock, also called GCLK_SERCOM0_CORE, is connected to SERCOM0 that supplies the functional logic clocks. The SERCOM0 interface, clocked by CLK_SERCOM0_APB, has been unmasked in the MCLK APB SERCOM0 mask register, MCLK.CLKMSK2.MSK7.

Figure 18-2. Example of SERCOM_0 Clock



To customize the clock distribution, refer to these registers and bit fields:

- The source oscillator for a generic clock generator 'n' is selected by writing to the Source bit field in the Generator Control n, (n = 0-15), 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, (i.e., GCLK Chapter).
- The AHB/APB/AIX Peripheral BUS clocks are enabled and disabled by writing to the respective bit in the AHB/APB/AIX Mask register. The AHB/APB/AIX clocks are enabled and disabled by writing to the respective bit in the AHB/APB/AIX Mask register (MCLK.CLKMSK[0,1,2,3].MSKn). They are enabled by default on reset.

18.2. Synchronous and Asynchronous Clocks

As the CPU and the peripherals can be in different clock domains, 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 complete.

In the data sheet, references to Synchronous Clocks are referring to the CPU and bus clocks (MCLK), while asynchronous clocks are generated by the Generic Clock Controller (GCLK).

18.3. Register Synchronization

18.3.1. Overview

All peripherals are composed of one digital bus interface connected to the APB, AHB, or AXI 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.

All registers in the bus interface are accessible without synchronization.

All registers in the peripheral core are synchronized when written. Some registers in the peripheral core are synchronized when read.

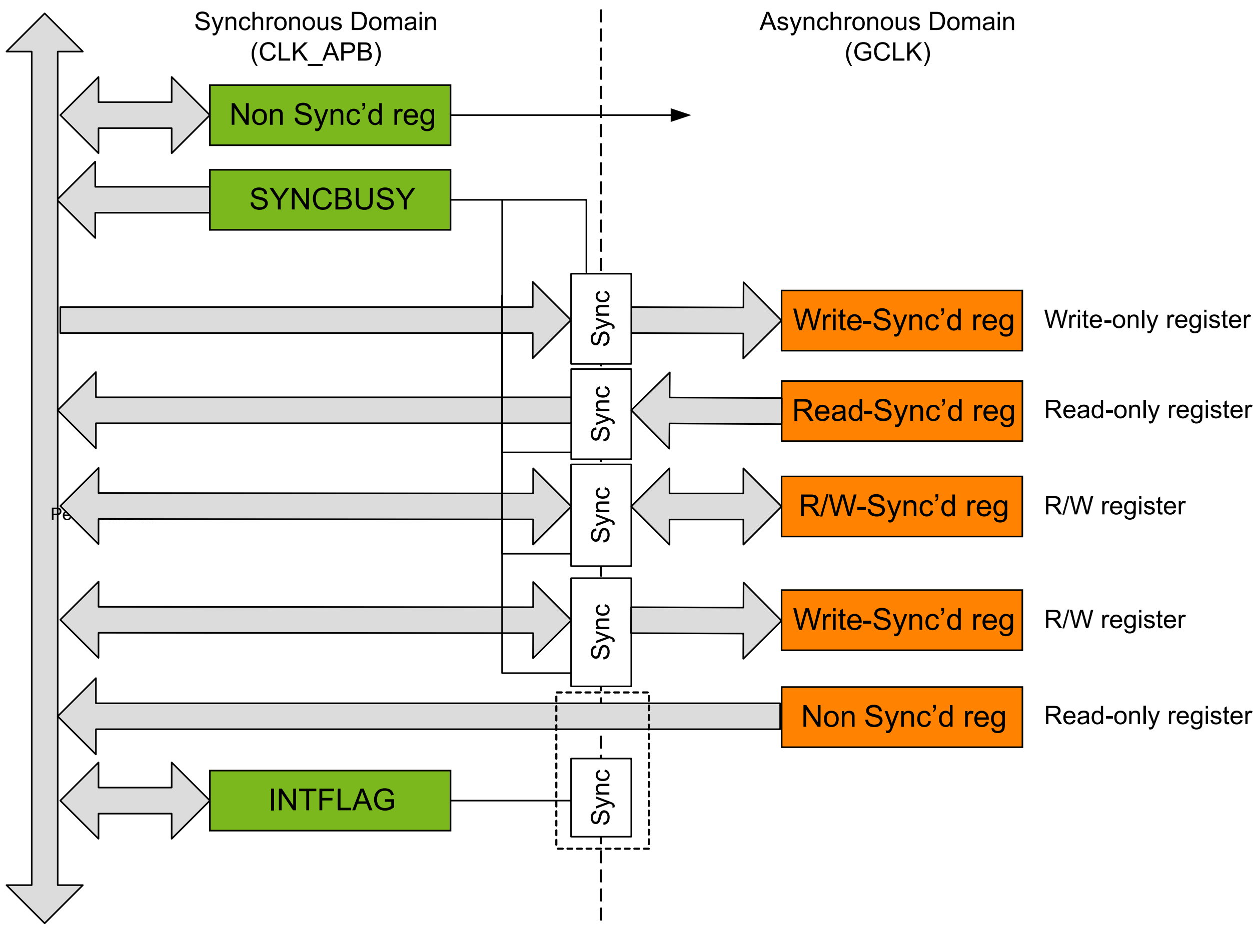
Each individual register description will have the properties Read-Synchronized and/or Write-Synchronized if a register is synchronized.

As shown in the figure below, each register that requires synchronization has its individual synchronizer and its individual synchronization status bit in the Synchronization Busy register (SYNCBUSY).

Notes:

1. For registers requiring both read-synchronization and write-synchronization, the corresponding bit in the SYNCBUSY register is shared.
2. Due to asynchronicity between the main clock domain and the peripheral clock domains, all Read-Synchronized registers must ensure synchronization is completed before accessing them after waking up from Standby Sleep mode. The respective bit in the Synchronization Busy register (SYNCBUSY) will be cleared when the write-synchronization is complete.

Figure 18-3. Register Synchronization Overview



18.3.2. 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. For additional information, refer to the 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:

The REGA and REGB are 8-bit core registers, and the 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.

18.3.3. General Read Synchronization

Read-synchronized registers are synchronized each time the register value is updated but the corresponding SYNCBUSY bits are not set. Reading a read-synchronized register does not start a new synchronization, it returns the last synchronized value.
Note: The corresponding bits in SYNCBUSY will automatically be set when the device wakes up from sleep because read-synchronized registers need to be synchronized. Therefore reading a read-synchronized register before its corresponding SYNCBUSY bit is cleared will return the last synchronized value before sleep mode.
Moreover, if a register is also write-synchronized, any write access while the SYNCBUSY bit is set will be discarded and generate an error.

18.3.4. Completion of Synchronization

In order to check if synchronization is complete, the user can either poll the relevant bits in SYNCBUSY or use the Synchronisation 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'.

18.3.5. Write Synchronization for CTRLA.ENABLE

Setting the Enable bit in a peripheral module's Control A register (CTRLA.ENABLE) will trigger write-synchronization and set SYNCBUSY.ENABLE.

CTRLA.ENABLE will read its new value immediately after being written.

SYNCBUSY.ENABLE will be cleared by hardware when the operation is complete.

The Synchronization Ready interrupt (if available) cannot be used to enable write-synchronization.

18.3.6. Write-Synchronization for Software Reset Bit

Setting the Software Reset bit in CTRLA (CTRLA.SWRST=1) will trigger write-synchronization and set SYNCBUSY.SWRST. When writing a ‘1’ to the CTRLA.SWRST bit it will immediately read as ‘1’.

CTRL.SWRST and SYNCBUSY.SWRST will be cleared by hardware when the peripheral has been reset.

Writing a '0' to the CTRL.SWRST bit has no effect.

The Ready interrupt (if available) cannot be used for Software Reset write-synchronization.

Note: Not all peripherals have the SWRST bit in the respective CTRLA register.

18.3.7. 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 $2 \times P_{APB}$.

18.4. Enabling a Peripheral

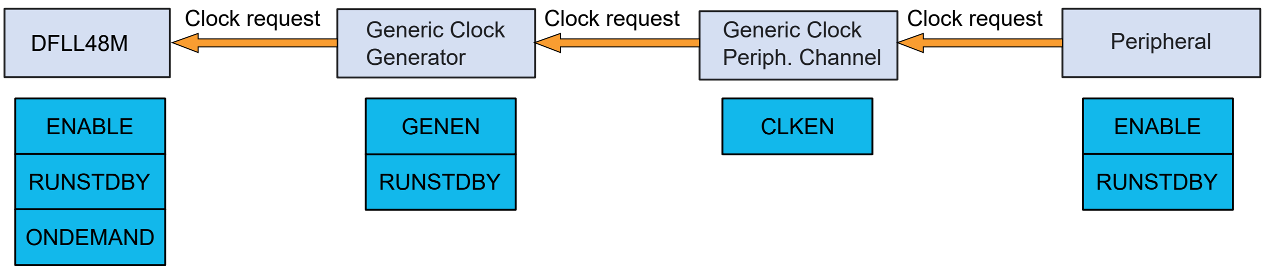
In order to enable a peripheral that is clocked by a Generic Clock, the following parts of the system needs to be configured:

- A running Clock Source as selected in the OSCCTRL Module
- A clock from the Generic Clock Generator (GCLK.GENCTRLn.SRC), 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 designated peripheral "m", (i.e., PCHCTRLm), must be configured to use a running Generic Clock Generator (i.e., PCHCTRLm.GEN), and the Generic Clock must be enabled.
- The user system bus interface of the peripherals are enabled by default on reset, (i.e., MCLK.CLKMSK[0,1,2].MSKn = 1), and doesn't need to be unmasked to enable it unless disabled at run time by the users software and the user needs to access the peripheral once again. In that case the user must re-enable, unmask, in the Main Clock Controller (MCLK.CLKMSK[0,1,2].MSKn = 1). If this is not done the peripheral in question registers will read all '0's and any writing attempts to the peripheral will be discarded.

18.5. On Demand Clock Requests

Figure 18-4. 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 or 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 (see Figure 18-4).

18.6. Power Consumption Versus Speed

When targeting for either a low-power or a fast acting system, some considerations have to be taken into account due to the nature of the asynchronous clocking of the peripherals:

Clocking a peripheral with a very low clock, the active power consumption of the peripheral will be lower. At the same time the synchronization to the synchronous (CPU) clock domain is dependent on the peripheral clock speed, and will take longer with a slower peripheral clock. This will cause worse response times and longer synchronization delays.

18.7. Clocks after Reset

On any Reset the synchronous clocks start to their initial state:

- DFLL48M is enabled and configured to run at 48 MHz Open Loop
- Generic Clock Generator 0 uses DFLL48M by default as a source and generates GCLK_MAIN and CLK_MAIN
- CPU and BUS clocks are undivided and enabled

On a Power-on Reset, the 32 KHz clock sources are reset and the GCLK module starts to its initial state:

- All Generic Clock Generators are disabled except Generator 0
- All Peripheral Channels in GCLK are disabled

On a User Reset the GCLK module starts to its initial state, except for:

- Generic Clocks that are write-locked, that is, the according WRTLOCK is set to 1 prior to Reset

19. Oscillator Controller (OSCCTRL)

19.1. Overview

The Oscillators Controller (OSCCTRL) provides a user interface to the XOSC48, RC DFLL48M, and one PLL 12.7 MHz to 200 MHz or up to 1.6 GHz for fractional divider module use. Through the interface registers, it is possible to enable, disable, calibrate, and monitor the OSCCTRL sub-peripherals. All sub-peripheral statuses are collected in the Status register (STATUS). They can additionally trigger interrupts upon status changes via the INTENSET, INTENCLR, and INTFLAG registers.

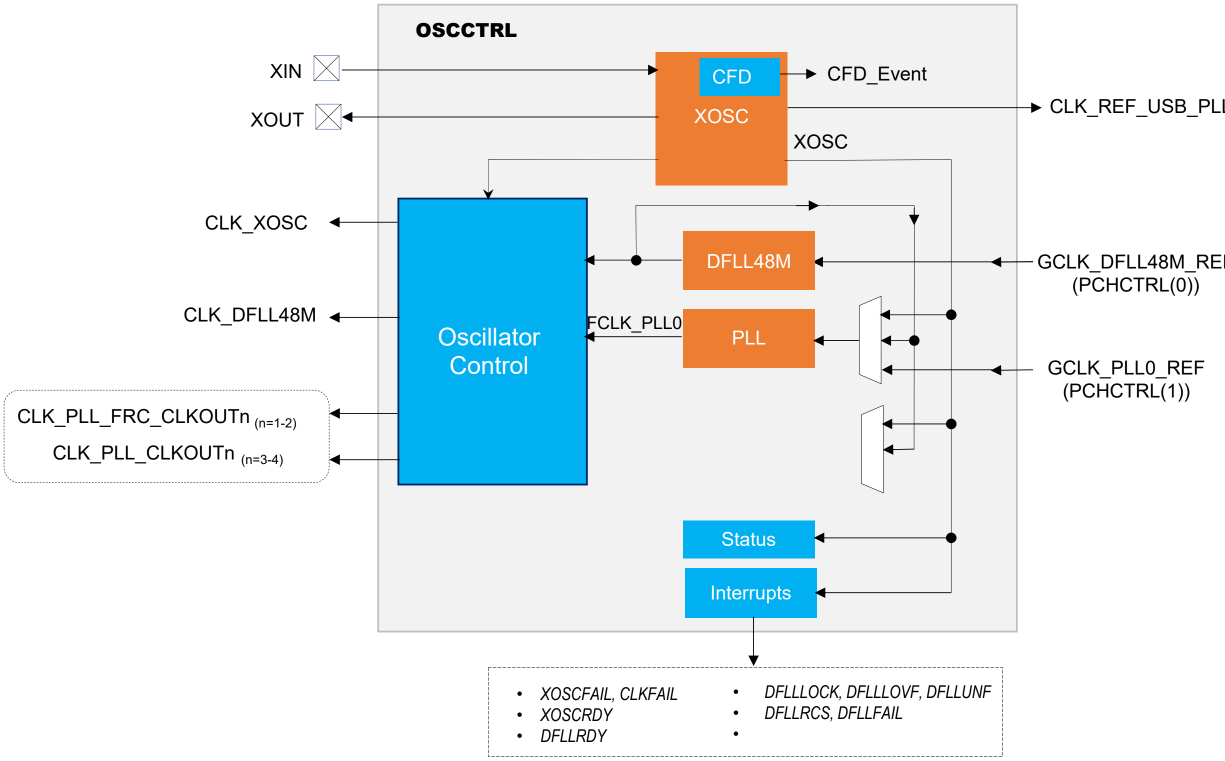
19.2. Features

The following are key features of the OSCCTRL module:

- 48 MHz Crystal Oscillator (XOSC48)
  - Supports 4 MHz to 48 MHz crystal oscillators
  - Supports ceramic resonators up to 10 MHz
  - AGC, Automatic gain control loop with manual override
  - User clock Ready Status
  - Clock-fail detection in AGC mode with safe clock switch
- RC48M (DFLL48M)
  - System clock output
  - 48.0 MHz (-5% to 2%) calibrated with DFLL48M in open-loop operation
  - 48.0 MHz (±0.25%) calibrated with DFLL48M in closed-loop operation
  - 8 MHz (±15%) in low-power mode with DFLL48M in open-loop operation
- One PLL, Phase-Locked Loop
  - 4 MHz to 48 MHz Reference Input Clock
  - Programmable Reference Input Clock divider
  - Single-ended output frequency: 12.7 MHz to 200 MHz or 1.6 GHz for fractional divider module input
  - Six single-ended outputs with programmable output clock dividers
  - Programmable closed loop bandwidth

19.3. OSCCTRL Block Diagram

Figure 19-1. OSCCTRL Block Diagram



19.4. Signal Descriptions

Signal Name	Type	Description
X_{IN}	Analog Input	Primary Multipurpose Crystal Oscillator or external clock generator input
X_{OUT}	Analog Output	Primary Multipurpose Crystal Oscillator output

19.5. Peripheral Dependencies

Peripheral Name	Base Address	NVIC IRQ Index: Source	MCLK AHBx/APBx Clock Enable Mask Bit	GCLK Peripheral Channel Clock Name:Register	PAC Peripheral Identifier (PAC.WRCTRL.PERIDx)	EVSYS Generator (EVSYS.CHANNELn.EVGENx)	Power Domain
OSCCTRL	0x4400_C000	5 : XOSCFAIL, XOSCRDY, CLKFAIL 6 : DFLLRDY, DFLLLOCK, DFLLOVF, DFLLUNF, DFLLRCS, DFLLFAIL 7 : PLLLOCKR	MCLK.CLKMSK1[6]	GCLK_OSCCTRL_DFLL48: GCLK.PCHCTRL[0] GCLK_OSCCTRL_PLL: GCLK.PCHCTRL[1]	6 : INTFLAGA[6]	2:XOSC_FAIL	VDDREG,AVDD

19.5.1. I/O Lines

The XOSC I/O lines are automatically configured when XOSC is enabled. There is no need for user configuration.

19.5.2. Power Management

The OSCCTRL can continue to operate in any sleep mode where the selected source clock is running. The OSCCTRL interrupts can be used to wake up the device from sleep modes. The events can trigger other operations in the system without exiting sleep modes.

For more information, refer to the Power Manager.

19.5.3. Clocks

The OSCCTRL gathers controls for all device oscillators and provides clock sources to the Generic Clock Controller (GCLK). The available clock sources are: XOSC, DFLL48M, PLL, and FRACDIV.

The DFLL48M requires a reference clock (GCLK_DFLL48M_REF) from the GCLK. The control logic uses the oscillator output, which is asynchronous to the user interface clock (CLK_OSCCTRL_APB). Due to this asynchronicity, writes to certain registers will require synchronization between the clock domains. Refer to Synchronization for further details.

The PLL requires a reference clock (GCLK_PLL_REF) from the GCLK when the PLL reference selector PLLCTRL.REFSEL is set to GCLK.

The FRACDIV control logic uses the fractional divider output, which is asynchronous to the user interface clock (CLK_OSCCTRL_APB). Due to this asynchronicity, writes to certain registers will require synchronization between the clock domains. Refer to Synchronization for further details.

19.5.4. Interrupts

The interrupt request line is connected to the Interrupt Controller. Using the OSCCTRL interrupts requires the interrupt controller to be configured first.

For more information, refer to the Nested Vector Interrupt Controller.

19.5.5. Events

The events are connected to the Event System. Using the events requires the Event System to be configured first. For more information, refer to the Event System (EVSYS).

19.5.6. 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.

19.5.7. Register Access Protection

All registers with write-access can be write-protected optionally by the Peripheral Access Controller (PAC), except for the following registers: The Interrupt Flag Status and Clear register (INTFLAG).

Optional write-protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write-Protection" property in each individual register description. PAC write-protection does not apply to accesses through an external debugger.

19.5.8. Analog Connections

The 4-48 MHz crystal must be connected between the XIN and XOUT pins, along with any required load capacitors.

Note: Refer to the *Electrical Characteristics* for more information about load capacitors.

19.6. Functional Description

19.6.1. Principles of Operation

XOSC, DFLL48M, and PLL 1.6G are configured through the OSCCTRL control registers. Through this interface, the oscillators are enabled, disabled, or have their calibration values updated. The Status register gathers different status signals coming from the oscillators controlled by the OSCCTRL. The status signals can be used to generate system interrupts, and in some cases wake up the system from Sleep mode, provided the corresponding interrupt is enabled.

19.6.2. External Multipurpose 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 4-48 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 pins 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 (XOSCCTRLA.ENABLE). To enable XOSC as an external crystal oscillator, the XTAL Enable bit (XOSCCTRLA.XTALEN) must be written to '1'. If XOSCCTRLA.XTALEN is zero, the external clock input on XIN will be enabled.

If the External Multipurpose Crystal Oscillator AGC, Auto Gain Control, Loop (XOSCCTRLA.AGC) is '1', the oscillator gain will be automatically adjusted starting from lowest setting, and will increase the gain in sequential steps every XOSCCTRLB.GBW[1:0] delay for a maximum of 16 steps accordingly, until a valid stable oscillation is detected which will result in lowest power consumption for a stable crystal oscillation. In this mode, XOSCCTRLA.AGC =1, the manual XOSC crystal gain settings XOSCCTRLB.GMAN[1:0] are ignored.

The AGC Gain control loop update rate can be controlled by the XOSCCTRLB.GBW[1:0] user configuration bits. This controls the AGC delays between each gain step increase. Some unique crystals may have slow start-up times, so to insure the lowest power gain setting it may be necessary to increase the AGC gain step delay update rate to allow enough time for a slow crystal start-up to be realized and detected before increasing the XOSC gain step again. This is rare and 95% of all 4 MHz to 48 MHz crystals start-up with the default XOSCCTRLB.GBW[1:0] value.

Alternatively, if the External Multipurpose Crystal Oscillator Auto Gain Control Loop (XOSCCTRLA.AGC) is '0', the user can manually select the crystal oscillator operating condition by setting the manual gain value in the XOSCCTRLB.GMAN[1:0] register. In this mode, XOSCCTRLA.AGC = 0, XOSCCTRLB.GBW[1:0] AGC update rate is ignored.

Start-Up time, XOSCCTRLA.STARTUP, select the maximum start-up time for the oscillator XOSC before a clock fail is acknowledged. The OSCULP32K oscillator is used to clock the start-up counter for the XOSC. Start-Up Time.

The XOSC will behave differently in different sleep modes, based on the settings of XOSCCTRLA.ONDEMAND, and XOSCCTRLA.ENABLE. XOSCCTRLA.ONDEMAND must be written when XOSCCTRLA.ENABLE = 0. Otherwise, the write of this bit is ignored. If XOSCCTRLA.ENABLE = 0, the XOSC will be always stopped. For XOSCCTRLA.ENABLE = 1, this table is valid:

Table 19-1. XOSC Sleep Behavior

CPU Mode	ON DEMAND	Sleep Behavior of XOSC and CFD
Active or Idle	0	Always run
Active or Idle	1	Run if requested by a peripheral
Standby	0	Always run
Standby	1	Run if requested by a peripheral
Backup	0	Always OFF
Backup	1	Always OFF

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 start-up time can be configured by changing the Oscillator Start-Up Time bit group (XOSCCTRLA.STARTUP) in the External Multipurpose Crystal Oscillator Control register. During the start-up 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 when the external clock or crystal oscillator is stable and ready to be used as a clock source. The INTFLAG.XOSCRDY bit is set on a zero-to-one transition of STATUS.XOSCRDY and an interrupt is generated if the External Multipurpose Crystal Oscillator Ready bit in the Interrupt Enable Set register (INTENSET.XOSCRDY) is set. If the External Multipurpose Crystal Oscillator Auto Gain Control Loop (XOSCCTRLA.AGC) is set, and the XOSC fail to oscillate after attempting all gain settings, the External Multipurpose Crystal Oscillator Startup Fail bit in the Status register (STATUS.XOSCFAIL) is set. The INTFLAG.XOSCFAIL bit is set on a zero-to-one transition of STATUS.XOSCFAIL and an interrupt is generated if the XOSC Startup Failure bit in the Interrupt Enable Set register (INTENSET.XOSCFAIL) is set. After the the startup time has elapsed and the External Multipurpose Crystal Oscillator did not fail oscillating, the output clock can also be monitored for failure by using the Clock Fail Detector (CFD). Refer to next section for CFD operations.

USBHS Reference Clock Division

The XOSC oscillator is the source of the USBHS PLL's reference clocks. Each USBHS PLL can request the XOSC as a reference clock. Upon request by a USBHS, the XOSC clock is prescaled by a clock divider and issued to the USBHS PLL. The clock division ratio and enable is configured by the XOSCCTRLA.USBHSDIV bitfield.

19.6.3. Clock Failure Detection Operation

The Clock Failure Detector (CFD) allows the user to monitor the external clock or crystal oscillator clock signal provided by the External Multipurpose Crystal Oscillator (XOSC). It detects failing operation of the XOSC clock and allows to switch to a safe clock in case of clock failure. The safe clock is derived from the DFLL48M with a configurable prescaler. The user can also switch from the safe clock to the XOSC clock in case of clock recovery. This allows to configure the safe clock in order to fulfill the operative conditions of the microcontroller. The CFD operation is automatically suspended when the XOSC clock is not requested in ONDEMAND mode or halted in STANDBY.

The user interface registers allow to enable, disable and configure the CFD. The Status register gives status on failure and clock switch conditions. The Clock Failure Detector can optionally trigger an interrupt or an event when a failure is detected.

Note: If the measured clock is intermittent or too slow, the STATUS:XOSCFAIL flag may appear to be toggling on and off until the clock is stopped or restored. However, the first rising edge will in any case be captured and logged in INTFLAG:XOSCFAIL.

Clock Failure Detection

At reset, the CFD is disabled. The CFD does not monitor the XOSC clock when the oscillator is disabled (XOSCCTRLA.ENABLE = 0).

Before starting the CFD operation, the user must start and enable the safe clock source (DFLL48M). To start the CFD operation, the user must write a one to the CFD Enable bit in the External Oscillator Control register (XOSCCTRLA.CFDEN). After the start or restart of the XOSC, the CFD does not detect failure until the start-up time, as configured by the Oscillator Start-Up Time (XOSCCTRLA.STARTUP) in the External Multipurpose Crystal Oscillator Control register, is elapsed. Once the XOSC Start-Up Time is elapsed, the XOSC clock is constantly monitored.

During a period of 4 safe clocks , the CFD watches for a clock activity from the XOSC. There must be one rising and one falling XOSC clock edges during a 4 safe clock periods to meet a non-failure status. If no activity is detected, the failure status is asserted. The Clock Failure status bit in the Status register (STATUS.CLKFAIL) is set. The Clock Failure interrupt flag bit in the Interrupt Flag register (INTFLAG.CLKFAIL) is set. If the CLKFAIL bit in the Interrupt Enable Set register (INTENSET.CLKFAIL) is set, an interrupt is generated . An output event is generated as well, if the Event Output enable bit in the Event Control register (EVCTRL.CFDEO) is set.

The XOSC clock continues to be monitored after a clock failure. The Clock Failure status bit in the Status register (STATUS.CLKFAIL) reflects the current XOSC clock activity.

Clock Switch Back

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 can be downscaled with 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 (STATUS.XOSCCKSW) in the Status register is set.

When the XOSC clock is switched to the safe clock, the External Multipurpose Crystal Oscillator Ready bit in the Status register (STATUS.XOSCRDY) is set when the safe clock is stable and ready to be used as a clock source.

When the CFD has switched to the safe clock, the XOSC is not disabled. 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 XOSC, the Clock Failure status bit in the Status register (STATUS.CLKFAIL) is cleared. The application can switch back to the XOSC clock by writing a one to Switch Back bit (XOSCCTRLA.SWBEN) in the External Oscillator Control register. Once the XOSC clock is switched back, the Switch Back bit (XOSCCTRLA.SWBEN) is cleared by the hardware.

Note: In the situation that the user has wrote a one to the Switch Back bit (XOSCCTRLA.SWBEN), the APB clock is permanently requested until the switch is completed.
If the switch back can not complete due to a persistent failure, it will prevent the system from entering into low power sleep mode.

To stop the clock request the user can disable the XOSC (XOSCCTRLA.ENABLE=0) and disable the CFD detection (XOSCCTRLA.CFDEN=0).

Prescaler:

The CFD has an internal configurable prescaler (XOSCCTRLA.CFDPRESC) to generate the safe clock from the DFLL48M clock. The prescaler size allows to scale down the DFLL48M clock such that the safe clock is not higher than the XOSC clock frequency monitored by the CFD. The frequency divider is 2^CFDPRESC where CFDPRESC range from 0 to 15.

Example: for an external crystal oscillator at 8 MHz and the DFLL48M internal oscillator configured to generate a 48 MHz clock, the prescaler must select a downscale value above 6 (48/8), for example, 8, therefore CFDPRESC = 3.

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. The CFD interrupt can be used to wake up the device from sleep modes.

Notes:

1. PLL0 must be stepped down in ≦ 75 MHz increments to ≦ 75 MHz output when entering sleep modes.
2. PLL0 must be stepped up to the operating frequency in ≦ 75 MHz increments after exiting sleep modes.
3. The step delay for both of these processes needs to be ≧ 1 us.

19.6.4. 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.

19.6.4.1. Basic DFLL48M Operation

DFLL48M Operating modes

The DFLL48M will behave differently in different sleep modes, based on the settings of DFLLCTRLA.ONDEMAND, and DFLLCTRLA.ENABLE. DFLLCTRLA.ONDEMAND must be written when DFLLCTRLA.ENABLE = 0 and DFLLSYNC.ENABLE = 0. Otherwise, the write of this bit is ignored. If DFLLCTRLA.ENABLE = 0, the DFLL48M will be always stopped. For DFLLCTRLA.ENABLE = 1, this table is valid:

Table 19-2. DFLL48M Sleep Behavior

CPU Mode	ON DEMAND	Sleep Behavior of DFLL48M
Active or Idle	0	Always run
Active or Idle	1	Run if requested by a peripheral
Standby	0	Always run

Table 19-2. DFLL48M Sleep Behavior (continued)

CPU Mode	ON DEMAND	Sleep Behavior of DFLL48M
Standby	1	Run if requested by a peripheral
Backup	0	Always OFF
Backup	1	Always OFF

The DFLL48M is used as a clock source for the generic clock generators, as described in the GCLK chapter. The DFLL48M is factory-calibrated for 48 MHz. The frequency calibration is applied at reset.

Register DFLLTEMPTRIM.TEMPTRIM stores the factory temperature calibration setting. This register cannot be written by the user.

DFLL48M Open-Loop Operation

After any reset, the open-loop mode is selected. When operating in open-loop mode, the output frequency of the DFLL48M will be determined by the values written to the DFLL TUNE register (DFLLTUNE.TUNE). It is possible to change the values of DFLLTUNE.TUNE and thereby the output frequency of the DFLL48M output clock, CLK_DFLL48M, while the DFLL48M is enabled and in use. CLK_DFLL48M is ready to be used when STATUS.DFLLRDY is set after enabling the DFLL48M.

DFLL48M Closed-Loop Operation

In closed-loop operation, the output frequency is continuously regulated against a reference clock. Once the multiplication factor is set, the oscillator tuning is automatically adjusted. The DFLL48M must be correctly configured before closed-loop operation can be enabled. After enabling the DFLL48M, it must be configured in the following way:

1. Enable and select a reference clock (CLK_DFLL48M_REF). CLK_DFLL48M_REF is Generic Clock Channel 0 (DFLL48M_Reference). Refer to GCLK for details.
2. Select the maximum step size allowed in finding the TUNE values by writing the appropriate values to the DFLL maximum step bit group (DFLLMUL.STEP) in the DFLL STEP register. A small step size will ensure low overshoot on the output frequency but will typically result in longer lock times. A high value might give a large overshoot but will typically provide faster locking. DFLLMUL.STEP should not be higher than 50% of the maximum value of DFLLTUNE.TUNE.
3. Select the multiplication factor in the DFLL Multiply Factor bit group (DFLLMUL.MUL) in the DFLL Multiplier register. Care must be taken when choosing DFLLMUL.MUL so that the output frequency does not exceed the maximum frequency of the device. If the target frequency is below the minimum or above the maximum frequency of the DFLL48M, the output frequency will be equal to the DFLL minimum or maximum frequency. Write the DFLLMUL.MUL to restore the TUNE register (DFLLTUNE.TUNE) to its reset value.
4. Start the closed loop mode by writing a one to the DFLL Loop Enable bit (DFLLCTRLB.LOOPEN) in the DFLL Control register.

The frequency of CLK_DFLL48M (Fclkdfll48m) is given by:

Equation 19-1. FCLKDFLL48M

$$F_{CLKDFLL48M} = (DFLLMUL.MUL * F_{CLKDFLL48MREF})$$

Where, $F_{CLKDFLL48MREF}$ is the frequency of the reference clock (CLK_DFLL48M_REF).

The DFLLTUNE register is read-only in closed loop mode and is controlled by the frequency tuner to meet user specified frequency.

DFLL48M Frequency Locking

In the lock of the frequency search in closed-loop mode, the control logic tunes the value in DFLLTUNE.TUNE, therefore the output frequency is very close to the desired frequency. On lock, the DFLL Locked bit (STATUS.DFLLLOCK) in the status register will be set.

Interrupts are generated by the rising of STATUS.DFLLLOCK if INTENSET.DFLLOCK is written to '1'.

CLK_DFLL48M is ready to be used when the DFLL Ready bit (STATUS.DFLLRDY) in the Status register is set, but the accuracy of the output frequency will not be met until the Lock state is reached. For lock times, refer to the Electrical Characteristics.

DFLL48M Frequency Error Measurement

The ratio between CLK_DFLL48M_REF and CLK48M_DFLL 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 (DFLLDIFF.DIFF) in the DFLL DIFF register. The relative error on CLK_DFLL48M compared to the target frequency is calculated as follows:

ERROR = (DIFF / MUL)

DFLL48M Drift Compensation

If the Stable DFLL Frequency bit (DFLLCTRLB.STABLE) in the DFLL Control register is zero, the frequency tuner will automatically compensate for drift in the CLK_DFLL48M without losing either of the locks. This means that DFLLTUNE.TUNE can change after every measurement of CLK_DFLL48M_REF. If the DFLLTUNE.TUNE value overflows or underflows due to large drift in temperature and/or voltage, the DFLL overflow or underflow bits (STATUS.DFLLOVF or STATUS.DFLLUNF) in the Status register will be set. After an overflow or underflow error condition, the user must rewrite DFLLMUL.MUL to ensure correct CLK_DFLL48M frequency. An interrupt is generated on a zero-to-one transition on STATUS.DFLLOVF or STATUS.DFLLUNF if the DFLL overflow or underflow bit (INTENSET.DFLLOVF or STATUS.DFLLUNF) in the Interrupt Enable Set register is set. If the Stable DFLL Frequency bit (DFLLCTRLB.STABLE) in the DFLL Control register is one, the DFLLTUNE.TUNE values will stay constant after the lock. The user can check for a possible drift by reading the frequency error in the DFLL Multiplication Ratio Difference bit group (DFLLDIFF.DIFF).

DFLL48M Reference Clock Stop Detection

If CLK_DFLL48M_REF stops or is running at a very low frequency (slower than CLK_DFLL48M/(2^17)), the DFLL Reference Clock Stopped bit (STATUS.DFLLRCS) in the Status register will be set. Detecting a stopped reference clock can take a long time, on 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 if the CLK_DFLL48M_REF is restarted. An interrupt is generated on a zero-to-one transition on STATUS.DFLLRCS if the DFLL Reference Clock Stopped bit (INTENSET.DFLLRCS) in the Interrupt Enable Set register is set.

DFLL48M Low Frequency, Low Power mode

The DFLL48M oscillator can operate in both open-loop or closed loop at a reduced frequency of 8 MHz. To select the low frequency/low power set the Low-Frequency bit (LOWFREQ) in the DFLL48M register. This bit is enable protected and can be changed only when the DFLL48M is disabled. In closed-loop the user must adjust the multiplier value DFLLMUL.MUL depending on the frequency of the reference clock (CLK_DFLL48M_REF) to have an output clock frequency of 8 MHz.

19.6.4.2. Additional DFLL48M Features

Dealing with Delay in the DFLL48M in Closed-Loop Mode

Note: During a maximum 30 cycles of the reference clock period, between lock flag asserted and frequency stabilization, DFLL accuracy will be limited. To minimize the cycle-to-cycle jitter during lock search, in addition to selecting a low step value (DFLLMUL.FSTEP, DFLLMUL.CSTEP), it is recommended to disable the Quick Lock feature.

The time from selecting a new CLK_DFLL48M frequency until this frequency is output by the DFLL48M can be up to several microseconds. If the value in DFLLMUL.MUL is small, this can lead to instability in the DFLL48M locking mechanism, which can prevent the DFLL48M from achieving lock. To avoid this, a chill cycle, during which the CLK_DFLL48M frequency is not measured, can be enabled. The chill cycle is enabled by default, but can be disabled by writing a one to the DFLL Chill Cycle Disable bit (DFLLCTRLB.CCDIS) in the DFLL Control register. Enabling chill cycles might double the lock time.

Another solution to this problem consists of using less strict lock requirements. This is called Quick Lock (QL), which is also enabled by default, but it can be disabled by writing a one to the Quick Lock Disable bit (DFLLCTRLB.QLDIS) in the DFLL Control register. The Quick Lock might lead to a larger spread in the output frequency than chill cycles, but the average output frequency is the same.

DFLL48M Lose Lock After Wake

DFLL48M can optionally reset its lock bit when it is disabled or stopped (not requested). This is configured by the Lose Lock After Wake bit (DFLLCTRLB.LLAW) in the DFLL Control register. If DFLLCTRLB.LLAW is zero, when the DFLL48M is re-enabled or requested again, it starts running with the same configuration as before being disabled, even if the reference clock is not available. The lock will not be lost. Therefore it is important that the user checks that the DFLL48M has reached the lock stage before entering a sleep mode. When the reference clock has restarted, the TUNE tracking will quickly compensate for any frequency drift during sleep if DFLLCTRLB.STABLE is zero. If DFLLCTRLB.LLAW is one when disabling or stopping the DFLL48M, the DFLL48M will lose its lock and needs to regain it through the full lock sequence.

DFLL48M Wait for Lock

DFLL48M can optionally control the issued clock. This is configured by the Wait For Lock bit (DFLLCTRLB.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 lock bit (STATUS.DFLLCK) 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.

DFLL48M Accuracy

The following two main factors that determine the accuracy of DFLL48M which can be tuned to obtain maximum accuracy when fine lock is achieved.

- Resolution: The frequency step between two tune values.
- The accuracy of the reference clock.

DFLL48M Backup oscillator

The DFLL48M has an alternate 6 MHz backup oscillator. At any DFLL48M oscillator power-up, the 6 MHz backup oscillator is started and watches for DFLL48M oscillations. If no oscillation is present after the longest DFLL48M startup time, the DFLL48M clock is switched to the 6 MHz backup oscillator.

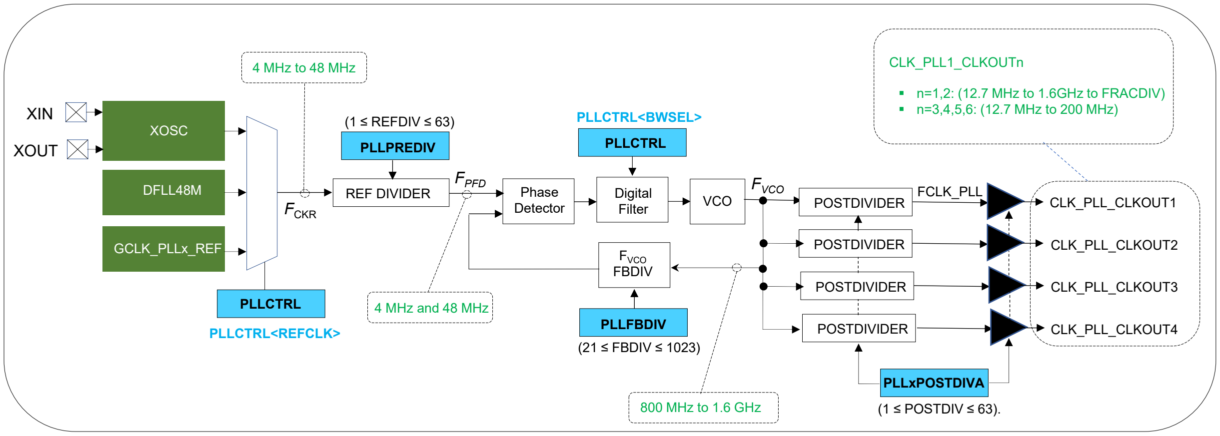
The DFLL Startup Failure bit in the STATUS register (STATUS.DFLLFAIL) will be set. The INTFLAG.DFLLFAIL bit is set on a zero-to-one transition of STATUS.DFLLFAIL and an interrupt is generated if the DFLL Startup Failure bit in the Interrupt Enable Set register (INTENSET.DFLLFAIL) is set.

19.6.5. Phase Locked Loop (PLL) Operation

The PLL provides a wide range of outputs from 12.7 MHz to 1600 MHz with support for input reference clock ranges from 4 MHz to 48 MHz. The PLL maintains a locked phase between the VCO input (reference) signal FPDF and the respective VCO output frequency FVCO through phase comparison and frequency multiplication.

The clocks from the PLL outputs (CLK_PLL) is a source for the Generic Clock module (GCLK).

Figure 19-2. PLL Block Diagram



Important: The frequency generated by the PLL oscillator can be up to 1600 MHz but is limited to 200 MHz (max.) for everything except inputs to FRACTIONAL DIVIDER. Depending on the operating conditions of the product using the PLL, the maximum allowed frequency can be as low as 12.7 MHz. Refer to the electrical characteristics of the product for a safe configuration of the PLL controller.

19.6.5.1. PLL Basic Operation

PLL Initialization, Enabling, and Disabling

The PLL is enabled by writing a one to the Enable bit in the Control register (PLLCTRL.ENABLE). The PLL is disabled by writing a zero to PLLCTRL.ENABLE.

Important: If the PLL is active, the user must ensure that at no time do they violate the minimum or maximum frequency ranges specified for F_{PFD}, F_{VCO} , and F_{CLK_PLL}. Failure to do so could destabilize the VCO and lead to unpredictable behavior. It may be necessary therefore to carefully select the order in which the user configures the various scaler values in the PLL based on the FCKR clock source frequencies used in the application.

PLL Reference selection

The PLL controller supports these independent sources of reference clock:

- DFLL48M: The reference clock is provided by the 48 MHz internal RC oscillator (DFLL48M).
- XOSC: The reference clock is provided by the External Multipurpose Crystal Oscillator (XOSC).
- GCLK_PLLn_REF: The reference clock “GCLK_PLLn_REF” is provided by the Generic Clock Controller.

The reference source can be selected by setting the REFSEL bits in the PLLCTRL register. The frequency of the reference must be comprised between 4 MHz and 48 MHz.

PLL settings

The frequency generated by the PLL is determined by the following control registers:

- PLLREFDIV: The PLL reference frequency divider. 1 ≤ REFDIV ≤ 63.
- PLLFBDIV: The PLL Feed Back frequency divider. 21 ≤ FBDIV ≤ 1023.

- POSTDIV: The PLL output frequency divider. 1 ≤ POSTDIV ≤ 63

When the controller is enabled, the relationship between the reference clock frequency and the output clock frequency is:

Equation 19-2. FCLK_PLL

$$F_{CLK\_PLL} = \left(F_{CKR} * \left(FBDIV / \left(REFDIV * POSTDIV\right)\right)\right)$$

Where,

- F_{CLK_PLL} is the frequency of the PLL output clock.
- F_{CKR} is the frequency of the selected reference clock.
- REFDIV is the reference prescaler value.
- FBDIV is the loop divider value.
- POSTDIV is the output prescaler value.

Note: F_{CLK_PLL} must always remain between 12.7 MHz to 200 MHz while configuring the various stages of the PLL, so that the VCO does not become unstable and lead to unpredictable behavior.

Example:

The goal is to have F_{CLK_PLL} = 120 MHz:

Given:

- XOSC = 12 MHz = F_{CKR}
- If REFDIV = 2 then F_{PFD} = (F_{CKR} / 2) = 6 MHz
- If FBDIV = 140 then F_{VCO} = (F_{PDF} * FBDIV) = (6 MHz * 140) = 840 MHz
- Setting POSTDIV = 7 then gives F_{CLK_PLL} = (F_{VCO} / 7) = (840 MHz /7) = 120 MHz

The frequency after the reference divider (FPDF) is given by the formula:

- $\mathbf{F_{PFD}}$ = F_{CKR} / REFDIV (Must be between 4 MHz to 48 MHz.)

The frequency of the Voltage Controlled Oscillator (VCO) giving the PLL oscillation is given by the formula:

- $\mathbf{F_{VCO}}$ = F_{CKR} * (FBDIV / REFDIV) (Must be between 800 MHz and 1600 MHz).

Note: F_{CKR} , REFDIV and FBDIV must be selected to satisfy these conditions.

Notes:

1. The PLLREFDIV and PLLFBDIV registers are not write-protected by the PAC. Alternatively, they can be write protected by setting the control bit PLLCTRL.WRTLOCK. When the OSCCTRL is PAC write-protected, the user can still tune the PLL frequency when the control bit PLLCTRL.WRTLOCK is cleared, or the user can also write-protect the PLLREFDIV and PLLFBDIV registers when the Control bit PLLCTRL.WRTLOCK is set.
2. Each PLL has up to four outputs. Each output has an individual PLL output frequency divider POSTDIVn and an individual output control enable OUTENn with n=0..3. Depending on the selected REFDIV and the reference frequency, the user shall set the Band Width selection bits (BWSEL) in the PLLCTRL register. Refer to the PLLCTRL.BWSEL definition in the Register Description section.

PLL Lock and Clock Generation

After the PLL oscillator is enabled, the PLL controller waits for the oscillator to issue a Lock status. The frequency of the PLL output clock CLK_PLL is stable when the PLL Lock Rising bit (PLLLOCKR) in the INTFLAG register is set.

Note: During each PLL start-up phase, the clock to the internal modules is not delivered as long as the first lock is not detected. When the lock is detected, the clock is released to the GCLK as long as requested. The PLL outputs whose OUTEN bit are set will start issuing a clock when the PLL Lock Rising bit (PLLOCKR) in the INFTLAG register is set. This bit will only be valid the first time it is set (i.e., the first lock detection). Any other PLLLOCKR values should be ignored after it is set, and the PLLOCKR interrupt should be disabled to prevent unexpected interrupts.

PLL Disabling

The PLL is disabled by writing a zero to PLLCTRL.ENABLE. Due to the synchronization of control and configuration registers, generation of internal timings to stop and power down properly the PLL, the PLL will be active for a few microseconds after CTRLA.ENABLE is cleared.

PLL Operation in Sleep Modes

The PLL will behave differently in different sleep modes, based on the settings of PLLCTRL.ONDEMAND and PLLCTRL.ENABLE. PLLCTRL.ONDEMAND must be written when PLLCTRL.ENABLE = 0. Otherwise, the write of this bit is ignored. If PLLCTRL.ENABLE = 0, the PLL will always be stopped. For PLLCTRL.ENABLE = 1, this table is valid:

Table 19-3. PLL Sleep Behavior

CPU Mode	ON DEMAND	Sleep Behavior of DFLL48M
Active or Idle	0	Always run
Active or Idle	1	Run if requested by a peripheral
Standby	0	Always run
Standby	1	Run if requested by a peripheral
Backup	0	Always OFF
Backup	1	Always OFF

PLL Reference Clock Switching

When a software operation requires reference clock switching, the normal operation is to disable the PLL, modify the PLLCTRL.REFCLK and PLLCTRL.BWSEL to select the desired reference source and activate the PLL again. The CLK_PLL output clock is ready when the INTFLAG.PLLOCKR bit is set.

PLL Updates

It is important to note that when doing run time PLL updates that the user ALWAYS insures that for any PLL register update, the PLL does not exceed any of the electrical specs listed below to ensure continued stable PLL operation. Depending on the PLL input clock frequency selected therefore the user may have to choose a specific sequence, (i.e. order), of PLL register updates to maintain specification compliance throughout the PLL module logic.

- F_{CKR} = 4 MHz to 48 MHz
- F_{PFD} = 4 MHz and 48 MHz
- F_{VCO} = 800 MHz to 1.6 GHz
- F_{CLK_PLL} = 12.7 MHz to 200 MHz

19.6.5.2. Fractional Divider

The Fractional Frequency Divider divides the PLL0 VCO, FVCO0, clock output frequency by a ratio composed of an integer part and a reminder part. Only the PLL0 supplies the clocks to the fractional dividers. FRACDIV0 is fed by PLL0 output 0 (under control of PLL0POSTDIVA[0]) and FRACDIV1 is fed by PLL0 output 1 (under control of PLL0POSTDIVA[1]). The maximum fractional divider input frequency is 1.6 GHz. The divided frequency is given by the integer and reminder part of the divider, FRACDIV.INTDIV and FRACDIV.REMDIV. The resulting frequency FFRACDIV is calculated using the following equation:

Equation 19-3. Fractional Divider Frequency

$$F_{CLK\_PLL0\_FRC\_CLKOUTn} = \left(F_{CLK\_PLL} / \left(2 * \left(INTDIV + \left(REMDIV / 512\right)\right)\right)\right)$$

Notes: This is not a true fractional divider in the sense that the resulting frequency is actually an average over time that represents the fractional frequency except in the cases where $F_{CLK\_PLL0\_FRC\_CLKOUTn}$ value corresponds to a whole integer value. For decimal fractional values, the fractional divider logic steals input F_{VCO} PLL clock cycles to produce and average output frequency equivalent to the desired frequency. As a result, the final $F_{CLK\_PLL0\_FRC\_CLKOUTn}$ output frequency will have jitter equivalent to:

- If Reminder ≤ 0.5:
  - $F_{CLK\_PLL0\_FRC\_CLKOUTn}$ Jitter = (Remainder / F_{VCO})
- If Reminder > 0.5:
  - $F_{CLK\_PLL0\_FRC\_CLKOUTn}$ Jitter = ((1-Remainder) / F_{VCO}))

Fractional Divider Operating mode

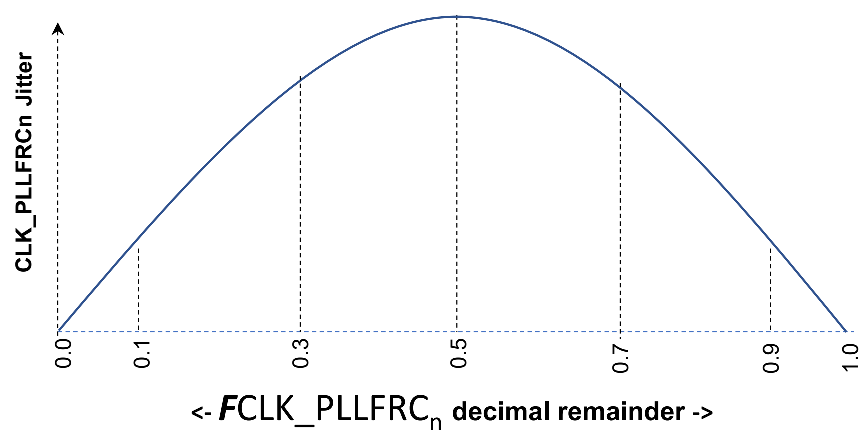
The Fractional Divider must be configured and enabled as shown in the following section.

19.6.5.2.1. Configure and Enable the PLL Output

1. Configure the integer and reminder divider factors FRACDIV.INTDIV and FRACDIV.REMDIV. If the PLL is not locked, the FRACDIV register write is pending until the PLL deliver a clock after the lock. The pending status can be checked in the SYNCBUSY.FRACDIVn register bit. The FRACDIV register can still be modified if the source PLL enable bit is not set.
2. Enable the PLL output.
3. Check the PLL lock with the register bit INTFLAG.PLLnLOCKR.
4. Check the SYNCBUSY.FRACDIVn register bit. When this bit is low, the FRACDIV is delivering the divided PLL clock. The Fractional Divider values FRACDIV.INTDIV and FRACDIV.REMDIV can be changed when the divider is operating. But before, the user must ensure that the previous Fractional Divider values change is completed by checking the bit SYNCBUSY.FRACDIVn is low.

Note: After a system reset the Fractional Divider starts in a frozen state. If the Fractional Divider is operating or synchronizing the divider factors, it is requesting a clock to the PLL source. The FRACDIV will stay frozen until the PLL source delivers a clock. After the PLL source delivers a clock, the FRACDIV is unfrozen. If the fractional Divider stops its operation (no more GCLK request and no more on-going synchronization), it will go back to a frozen state and request the PLL source clock until it is actually frozen. The freeze/unfreeze process ensures the FRACDIV divides on a locked PLL clock.

Figure 19-3. Fractional Divider



Notes:

1. FCLK_PLLFRC0 = (F_{CLK_PLL0} / (2 x (INTDIV + (REMDIV / 512)))).
2. The maximum permitted fractional output frequency, FCLK_PLLFRC0, must always be limited to 200MHz by the user.
3. Setting both INTDIV and REMDIV = 0 will yield FCLK_PLLFRC0 = F_{CLK_PLL0} which effectively bypasses the fractional divider module in which case the user must limit the output of PLL0 to the fractional divider module to FPLL0 = F_{CLK_PLL0} = 200MHz max.

19.6.6. OSCCTRL 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 Startup Failure:** A “0-to-1” transition on the STATUS.XOSCFAIL bit is detected
- **CLKFAIL - Xosc Clock Failure:** A “0-to-1” transition on the STATUS.CLKFAIL bit is detected
- **DFLLRDY - DFLL48m Ready:** A “0-to-1” transition on the STATUS.DFLLRDY bit is detected
- **DFLLLOCK - DFLL48m Lock:** A “0-to-1” transition on the STATUS.DFLLLOCK bit is detected
- **DFLLOVF - DFLL48m Overflow:** A “0-to-1” transition on the STATUS.DFLLOVF bit is detected
- **DFLLUNF - DFLL48m Underflow:** A “0-to-1” transition on the STATUS.DFLLUNF bit is detected
- **DFLLRCS - DFLL48m Reference Clock Stop:** A “0-to-1” transition on the STATUS.DFLLRCS bit is detected
- **DFLLFAIL - DFLL Startup Failure:** A “0-to-1” transition on the STATUS.DFLLFAIL bit is detected

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 one to the corresponding bit in the Interrupt Enable Set register (INTENSET) and disabled by writing a one 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 fewer request lines than 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.

19.6.7. OSCCTRL Events

The CFD, “Clock Fail Detect”, can generate the following output event:

- Clock Failure (CLKFAIL): Generated when the XOSC Clock Failure status bit is set in the Status register (STATUS.CLKFAIL). The CFD event is not generated when the XOSC Clock Switch 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.

19.6.8. OSCCTRL Synchronization

Due to the multiple clock domains, some registers in the DFLL48M must be synchronized when accessed. A register can require:

- Synchronization when written

- Synchronization when read
- No synchronization When executing an operation that requires synchronization, the relevant synchronization bit in the Synchronization Busy register (DFLLSYNC) will be set immediately and cleared when synchronization is complete

The following registers need synchronization:

- ENABLE bit in DFLLCTRLA register - write-synchronized
- DFLLCTRLB register - read-synchronized
- DFLLTUNE register - read- and write-synchronized
- DFLLMUL register - write-synchronized
- FRACDIV

19.7. Register Summary

For descriptions and definitions of both Register and bitfield properties, refer to Register Properties.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	EVCTRL	31:24								
		23:16								
		15:8								
		7:0								CFDEO
0x04	INTENCLR	31:24								PLL0LOCKR
		23:16								
		15:8			DFLLFAIL	DFLLRCS	DFLLUNF	DFLLOVF	DFLLLOCK	DFLLRDY
		7:0						CLKFAIL	XOSCFAIL	XOSCRDY
0x08	INTENSET	31:24								PLL0LOCKR
		23:16								
		15:8			DFLLFAIL	DFLLRCS	DFLLUNF	DFLLOVF	DFLLLOCK	DFLLRDY
		7:0						CLKFAIL	XOSCFAIL	XOSCRDY
0x0C	INTFLAG	31:24								PLL0LOCKR
		23:16								
		15:8			DFLLFAIL	DFLLRCS	DFLLUNF	DFLLOVF	DFLLLOCK	DFLLRDY
		7:0						CLKFAIL	XOSCFAIL	XOSCRDY
0x10	STATUS	31:24								
		23:16								
		15:8			DFLLFAIL	DFLLRCS	DFLLUNF	DFLLOVF	DFLLLOCK	DFLLRDY
		7:0					XOSCCKSW	CLKFAIL	XOSCFAIL	XOSCRDY
0x14	XOSCCTRLA	31:24	WRTLOCK						USBHSDIV[1:0]	
		23:16					CFDPRESC[3:0]			
		15:8					STARTUP[3:0]			
		7:0	ONDEMAND		SWBEN	CFDEN	XTALEN	AGC	ENABLE	
0x18	XOSCCTRLB	31:24	WRTLOCK							
		23:16								
		15:8								
		7:0				GBW[1:0]		GRES	GMAN[1:0]	
0x1C ... 0x2B	Reserved									
0x2C	DFLLCTRLA	31:24								
		23:16								
		15:8								
		7:0	ONDEMAND				LOWFREQ	WRTLOCK	ENABLE	
0x30	DFLLCTRLB	31:24								
		23:16								
		15:8								
		7:0	WAITLOCK		QLDIS	CCDIS		LLAW	STABLE	LOOPEN
0x34	DFLLTUNE	31:24								
		23:16								
		15:8								
		7:0		TUNE[6:0]						
0x38	DFLLDIFF	31:24								
		23:16								
		15:8	DIFF[15:8]							
		7:0	DIFF[7:0]							
0x3C	DFLLMUL	31:24								
		23:16		STEP[6:0]						
		15:8	MUL[15:8]							
		7:0	MUL[7:0]							
0x40	PLL0CTRL	31:24								
		23:16								
		15:8			BWSEL[2:0]			REFSEL[2:0]		
		7:0	ONDEMAND					WRTLOCK	ENABLE	

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x44	PLL0FBDIV	31:24								
		23:16								
		15:8							FBDIV[9:8]	
		7:0	FBDIV[7:0]							
0x48	PLL0REFDIV	31:24								
		23:16								
		15:8								
		7:0			REFDIV[5:0]					
0x4C	PLL0POSTDIVA	31:24	OUTEN3		POSTDIV3[5:0]					
		23:16	OUTEN2		POSTDIV2[5:0]					
		15:8	OUTEN1		POSTDIV1[5:0]					
		7:0	OUTEN0		POSTDIV0[5:0]					
0x50	PLL0POSTDIVB	31:24								
		23:16								
		15:8	OUTEN5		POSTDIV5[5:0]					
		7:0	OUTEN4		POSTDIV4[5:0]					
0x54 ... 0x6B	Reserved									
0x6C	FRACDIV0	31:24		INTDIV[14:8]						
		23:16	INTDIV[7:0]							
		15:8	REMDIV[8:1]							
		7:0	REMDIV[0]							
0x70 ... 0x77	Reserved									
0x78	SYNCBUSY	31:24								
		23:16								
		15:8								
		7:0		FRACDIV0	DFLLMUL	DFLLDIFF	DFLLTUNE	DFLLCTRLB	DFLLENABLE	

19.7.1. Event Control

Name: EVCTRL
Offset: 0x00
Reset: 0x00000000
Property: PAC Write-Protection

Table 19-4. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

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

Bit 0 – CFDEO Clock Failure Detector Event Output Enable

This bit indicates whether the XOSC Clock Failure detector event output is enabled and an output event will be generated when the XOSC Clock Failure detector detects a clock failure.

Note: To prevent false event generation, the bit CFDEO must be set or cleared only when the XOSC is disabled (XOSCCTRLn.ENABLE=0).

Value	Description
0	Clock Failure detector event output is disabled and an event will not be generated on a clock fail.
1	Clock Failure detector event output is enabled and an event will be generated on a clock fail.

19.7.2. 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).

Table 19-5. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
								PLL0LOCKR
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
			DFLLFAIL	DFLLRCS	DFLLUNF	DFLLOVF	DFLLLOCK	DFLLRDY
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
						CLKFAIL	XOSCFAIL	XOSCRDY
Access						R/W	R/W	R/W
Reset						0	0	0

Bit 24 – PLL0LOCKR PLL Lock Rise Interrupt Enable
Writing a zero to this bit has no effect.
Writing a '1' to this bit will clear the PLL Lock Rise Interrupt Enable bit, which disables the PLL Lock Rise interrupt.

Value	Description
0	The PLL Lock Rise interrupt is disabled.
1	The PLL Lock Rise interrupt is enabled, and an interrupt request will be generated when the PLL Lock Rise Interrupt flag is set.

Bit 13 – DFLLFAIL DFLL Startup Failure Interrupt Enable
Writing a zero to this bit has no effect.
Writing a '1' to this bit will clear the DFLL Startup Failure Interrupt Enable bit, which disables the DFLL Startup Failure interrupt.

Value	Description
0	The DFLL48M Startup Failure interrupt is disabled.
1	The DFLL48M Startup Failure interrupt is enabled, and an interrupt request will be generated when the DFLL Startup Failure Interrupt flag is set.

Bit 12 – DFLLRCS DFLL Reference Clock Stopped Interrupt Enable
Writing a zero to this bit has no effect.
Writing a '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 DFLL48M Reference Clock Stopped interrupt is disabled.
1	The DFLL48M Reference Clock Stopped interrupt is enabled, and an interrupt request will be generated when the DFLL Reference Clock Stopped Interrupt flag is set.

Bit 11 – DFLLUNF DFLL Tuner Underflow Interrupt Enable
Writing a zero to this bit has no effect.
Writing a '1' to this bit will clear the DFLL Tuner Underflow Interrupt Enable bit, which disables the DFLL Tuner Underflow interrupt.

Value	Description
0	The DFLL Tuner Underflow interrupt is disabled.
1	The DFLL Tuner Underflow interrupt is enabled, and an interrupt request will be generated when the DFLL Tuner Underflow Interrupt flag is set.

Bit 10 – DFLLOVF DFLL Tuner Overflow Interrupt Enable
Writing a zero to this bit has no effect.
Writing a '1' to this bit will clear the DFLL Tuner Overflow Interrupt Enable bit, which disables the DFLL Tuner Overflow interrupt.

Value	Description
0	The DFLL Tuner Overflow interrupt is disabled.
1	The DFLL Tuner Overflow interrupt is enabled, and an interrupt request will be generated when the DFLL Tuner Overflow Interrupt flag is set.

Bit 9 – DFLLLOCK DFLL Lock Interrupt Enable
Writing a zero to this bit has no effect.
Writing a '1' to this bit will clear the DFLL Lock Interrupt Enable bit, which disables the DFLL Lock interrupt.

Value	Description
0	The DFLL Lock interrupt is disabled.
1	The DFLL Lock interrupt is enabled, and an interrupt request will be generated when the DFLL Lock Interrupt flag is set.

Bit 8 – DFLLRDY DFLL Ready Interrupt Enable
Writing a zero to this bit has no effect.
Writing a '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 2 – CLKFAIL XOSC Clock Failure Interrupt Enable
Writing a ‘0’ to this bit has no effect.
Writing a '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.

Value	Description
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 1 – XOSCFAIL XOSC Startup Failure Interrupt Enable

Writing a ‘0’ to this bit has no effect.
Writing a '1' to this bit will clear the XOSC Startup Failure Interrupt Enable bit, which disables the XOSC Startup Failure interrupt.

Value	Description
0	The XOSC Startup Failure interrupt is disabled.
1	The XOSC Startup Failure interrupt is enabled, and an interrupt request will be generated when the XOSC Startup Failure Interrupt flag is set.

Bit 0 – XOSCRDY XOSC Ready Interrupt Enable

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

19.7.3. Interrupt Enable Set

Name: INTENSET
Offset: 0x08
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 register (INTENCLR).

Table 19-6. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
								PLL0LOCKR
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
			DFLLFAIL	DFLLRCS	DFLLUNF	DFLLOVF	DFLLLOCK	DFLLRDY
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
						CLKFAIL	XOSCFAIL	XOSCRDY
Access						R/W	R/W	R/W
Reset						0	0	0

Bit 24 – PLL0LOCKR PLL Lock Rise Interrupt Enable

Writing a zero to this bit has no effect.
Writing a '1' to this bit will clear the PLL Lock Rise Interrupt Enable bit, which disables the PLL Lock Rise interrupt.

Value	Description
0	The PLL Lock Rise interrupt is disabled.
1	The PLL Lock Rise interrupt is enabled, and an interrupt request will be generated when the PLL Lock Rise Interrupt flag is set.

Bit 13 – DFLLFAIL DFLL Startup Failure Interrupt Enable

Writing a zero to this bit has no effect.
Writing a '1' to this bit will clear the DFLL Startup Failure Interrupt Enable bit, which disables the DFLL Startup Failure interrupt.

Value	Description
0	The DFLL48M Startup Failure interrupt is disabled.
1	The DFLL48M Startup Failure interrupt is enabled, and an interrupt request will be generated when the DFLL Startup Failure Interrupt flag is set.

Bit 12 – DFLLRCS DFLL Reference Clock Stopped Interrupt Enable

Writing a zero to this bit has no effect.
Writing a '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 DFLL48M Reference Clock Stopped interrupt is disabled.
1	The DFLL48M Reference Clock Stopped interrupt is enabled, and an interrupt request will be generated when the DFLL Reference Clock Stopped Interrupt flag is set.

Bit 11 – DFLLUNF DFLL Tuner Underflow Interrupt Enable

Writing a zero to this bit has no effect.
Writing a '1' to this bit will clear the DFLL Tuner Underflow Interrupt Enable bit, which disables the DFLL Tuner Underflow interrupt.

Value	Description
0	The DFLL Tuner Underflow interrupt is disabled.
1	The DFLL Tuner Underflow interrupt is enabled, and an interrupt request will be generated when the DFLL Tuner Underflow Interrupt flag is set.

Bit 10 – DFLLOVF DFLL Tuner Overflow Interrupt Enable

Writing a zero to this bit has no effect.
Writing a '1' to this bit will clear the DFLL Tuner Overflow Interrupt Enable bit, which disables the DFLL Tuner Overflow interrupt.

Value	Description
0	The DFLL Tuner Overflow interrupt is disabled.
1	The DFLL Tuner Overflow interrupt is enabled, and an interrupt request will be generated when the DFLL Tuner Overflow Interrupt flag is set.

Bit 9 – DFLLLOCK DFLL Lock Interrupt Enable

Writing a zero to this bit has no effect.
Writing a '1' to this bit will clear the DFLL Lock Interrupt Enable bit, which disables the DFLL Lock interrupt.

Value	Description
0	The DFLL Lock interrupt is disabled.
1	The DFLL Lock interrupt is enabled, and an interrupt request will be generated when the DFLL Lock Interrupt flag is set.

Bit 8 – DFLLRDY DFLL Ready Interrupt Enable

Writing a zero to this bit has no effect.
Writing a '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 2 – CLKFAIL XOSC Clock Failure Interrupt Enable

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

Value	Description
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 1 – XOSCFAIL XOSC Startup Failure Interrupt Enable
Writing a ‘0’ to this bit has no effect.
Writing a '1' to this bit will clear the XOSC Startup Failure Interrupt Enable bit, which disables the XOSC Startup Failure interrupt.

Value	Description
0	The XOSC Startup Failure interrupt is disabled.
1	The XOSC Startup Failure interrupt is enabled, and an interrupt request will be generated when the XOSC Startup Failure Interrupt flag is set.

Bit 0 – XOSCRDY XOSC Ready Interrupt Enable
Writing a ‘0’ to this bit has no effect.
Writing a '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.

19.7.4. Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x0C
Reset: 0x00000000

Note: Subsequent to an interrupt flag being cleared, the flag must be read back to verify the clear before exiting the ISR. Failure to do this can result in duplicate interrupts.

Table 19-7. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
								PLL0LOCKR
Access								HS/R/W
Reset								0

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
			DFLLFAIL	DFLLRCS	DFLLUNF	DFLLOVF	DFLLLOCK	DFLLRDY
Access			HS/R/W	HS/R/W	HS/R/W	HS/R/W	HS/R/W	HS/R/W
Reset			0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
						CLKFAIL	XOSCFAIL	XOSCRDY
Access						HS/R/W	HS/R/W	HS/R/W
Reset						0	0	0

Bit 24 – PLL0LOCKR PLL Lock Rise

This flag is cleared by writing a '1' to it.
This flag is set on a zero-to-one transition of the PLL Lock and will generate an interrupt request if INTENSET.PLLLOCKR is '1'.
Writing a ‘0’ to this bit has no effect.
Writing a '1' to this bit clears the PLL Lock Rise interrupt flag.
Note: This bit is only valid for the first time it sets, when the PLL locks. This bit should be ignored afterwards, and the interrupt disabled so there are no unexpected interrupts.

Bit 13 – DFLLFAIL DFLL Startup Failure

This flag is cleared by writing a '1' to it.
This flag is set on a zero-to-one transition of the DFLL Startup Failure bit in the Status register (STATUS.DFLLFAIL) and will generate an interrupt request if INTENSET.DFLLFAIL is '1'.
Writing a ‘0’ to this bit has no effect.
Writing a '1' to this bit clears the DFLL Startup Failure interrupt flag.

Bit 12 – DFLLRCS DFLL Reference Clock Stopped

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

This flag is set on a zero-to-one 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 a ‘0’ to this bit has no effect.
Writing a '1' to this bit clears the DFLL Reference Clock Stopped interrupt flag.

Bit 11 – DFLLUNF DFLL Tuner Underflow
This flag is cleared by writing a '1' to it.
This flag is set on a zero-to-one transition of the DFLL Tuner Underflow bit in the Status register (STATUS.DFLLUNF) and will generate an interrupt request if INTENSET.DFLLUNF is '1'.
Writing a ‘0’ to this bit has no effect.
Writing a '1' to this bit clears the DFLL Tuner Underflow interrupt flag.

Bit 10 – DFLLOVF DFLL Tuner Overflow
This flag is cleared by writing a '1' to it.
This flag is set on a zero-to-one transition of the DFLL Tuner Overflow bit in the Status register (STATUS.DFLLOVF) and will generate an interrupt request if INTENSET.DFLLOVF is '1'.
Writing a ‘0’ to this bit has no effect.
Writing a '1' to this bit clears the DFLL Tuner Overflow interrupt flag.

Bit 9 – DFLLLOCK DFLL Lock
This flag is cleared by writing a '1' to it.
This flag is set on a zero-to-one transition of the DFLL Lock bit in the Status register (STATUS.DFLLLOCK) and will generate an interrupt request if INTENSET.DFLLLOCK is '1'.
Writing a ‘0’ to this bit has no effect.
Writing a '1' to this bit clears the DFLL Lock interrupt flag.

Bit 8 – DFLLRDY DFLL Ready
This flag is cleared by writing a '1' to it.
This flag is set on a zero-to-one transition of the DFLL Ready bit in the Status register (STATUS.DFLLRDY) and will generate an interrupt request if INTENSET.DFLLRDY is '1'.
Writing a ‘0’ to this bit has no effect.
Writing a '1' to this bit clears the DFLL Ready interrupt flag.

Bit 2 – CLKFAIL XOSC Clock Failure
This flag is cleared by writing a '1' to it.
This flag is set on a zero-to-one transition of the XOSC Clock Failure 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 clears the XOSC Clock Failure interrupt flag.

Bit 1 – XOSCFAIL XOSC Startup Failure
This flag is cleared by writing a '1' to it.
This flag is set on a zero-to-one transition of the XOSC Startup Failure bit in the Status register (STATUS.XOSCFAIL) and will generate an interrupt request if INTENSET.XOSCFAIL is '1'.
Writing a ‘0’ to this bit has no effect.
Writing a '1' to this bit clears the XOSC Startup Failure interrupt flag.

Bit 0 – XOSCRDY XOSC Ready
This flag is cleared by writing a '1' to it.
This flag is set on a zero-to-one transition of the XOSC Ready bit in the Status register (STATUS.XOSCRDY) and will generate an interrupt request if INTENSET.XOSCRDY is '1'.
Writing a ‘0’ to this bit has no effect.
Writing a '1' to this bit clears the XOSC Ready interrupt flag.

19.7.5. Status

Name: STATUS
Offset: 0x10
Reset: 0x00000000

Table 19-8. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
			DFLLFAIL	DFLLRCS	DFLLUNF	DFLLOVF	DFLLLOCK	DFLLRDY
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
					XOSCCKSW	CLKFAIL	XOSCFAIL	XOSCRDY
Access					R	R	R	R/W
Reset					0	0	0	0

Bit 13 – DFLLFAIL DFLL Startup Failure

Value	Description
0	DFLL Startup failure is not detected.
1	DFLL Startup failure is detected.

Bit 12 – DFLLRCS DFLL Reference Clock Stopped

Value	Description
0	DFLL reference clock is running.
1	DFLL reference clock has stopped.

Bit 11 – DFLLUNF DFLL Tuner Underflow

Value	Description
0	DFLL Tuner Underflow is not detected.
1	DFLL Tuner Underflow is detected.

Bit 10 – DFLLOVF DFLL Tuner Overflow

Value	Description
0	DFLL Tuner Overflow is not detected.
1	DFLL Tuner Overflow is detected.

Bit 9 – DFLLLOCK DFLL Lock

Value	Description
0	DFLL lock is not detected.
1	DFLL lock is detected.

Bit 8 – DFLLRDY DFLL Ready

Value	Description
0	DFLL is not ready.
1	DFLL is stable and ready to be used as a clock source.

Bit 3 – 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 2 – CLKFAIL XOSC Clock Failure

Value	Description
0	XOSC Clock failure is not detected.
1	XOSC Clock failure is detected.

Bit 1 – XOSCFAIL XOSC Startup Failure

Note: If the measured clock is intermittent or too slow, this bit may appear to be toggling on and off until the clock is stopped or restored. However, the first rising edge will in any case be captured and logged in INTFLAG:XOSCFAIL.

Value	Description
0	XOSC Startup failure is not detected.
1	XOSC Startup failure is detected.

Bit 0 – XOSCRDY XOSC Ready

Note: If the CFD is enabled and the XOSC clock is failing, the XOSCRDY status bit remains high, if already set.

Value	Description
0	XOSC is not ready.
1	XOSC is stable and ready to be used as a clock source.

19.7.6. External Multipurpose Crystal Oscillator Control A

Name: XOSCCTRLA
Offset: 0x14
Reset: 0x00000D00
Property: PAC Write-Protection

Table 19-9. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	WRTLOCK						USBHSDIV[1:0]	
Access	R/W						R/W	R/W
Reset	0						0	0

Bit	23	22	21	20	19	18	17	16
					CFDPRESC[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
					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
	ONDEMAND		SWBEN	CFDEN	XTALEN	AGC	ENABLE	
Access	R/W		R/W	R/W	R/W	R/W	R/W	
Reset	0		0	0	0	0	0	

Bit 31 – WRTLOCK Write Lock for CTRLA register
Note: Once the WRTLOCK bit is set, it can only be cleared by a reset.

Value	Description
0	The XOSCCTRLA register can be modified by a system write.
1	The XOSCCTRLA (except XOSCCTRLA.SWBEN) register is write protected.

Bits 25:24 – USBHSDIV[1:0] USBHS Reference Clock Division
These bits select the XOSC division factor for the USBHS PLL reference clock. These bits are XOSCCTRLA.ENABLE protected and cannot be updated if XOSCCTRLA.ENABLE=1.

Value	Name	Description
0x0	DIS	USBHSPLL reference XOSC clock is disabled
0x1	DIV1	USBHSPLL reference XOSC clock is divided by 1
0x2	DIV2	USBHSPLL reference XOSC clock is divided by 2
0x3	DIV4	USBHSPLL reference XOSC clock is divided by 4

Bits 19:16 – CFDPRESC[3:0] Clock Failure Detector Prescaler
These bits select the DFLL48oscillator post scaler for the clock fail detector. The CFD safe clock frequency is the DFLL48 frequency divided by 2^CFDPRESC. These bits are XOSCCTRLA.ENABLE protected and cannot be updated if XOSCCTRLA.ENABLE=1

CFDPRESC[3:0]	Name	Frequency
0x0	DIV1	48 MHz
0x1	DIV2	24 MHz
0x2	DIV4	12 MHz
0x3	DIV8	6 MHz
0x4	DIV16	3 MHz
0x5	DIV32	1.5 MHz
0x6	DIV64	0.75 MHz
0x7	DIV128	0.3125 MHz

Bits 11:8 – STARTUP[3:0] Start-Up Time for External Multipurpose Crystal Oscillator

These bits select start-up time for the oscillator XOSC according to the table below before a clock fail is acknowledged. The OSCULP32K oscillator is used to clock the start-up counter. These bits are XOSCCTRLA.ENABLE protected and cannot be updated if XOSCCTRLA.ENABLE = 1.

STARTUP[3:0]	Number of OSCULP32KClock Cycles	Approximate Equivalent Time
0x0	1	31μs
0x1	2	61μs
0x2	4	122μs
0x3	8	244μs
0x4	16	488μs
0x5	32	977μs
0x6	64	1953μs
0x7	128	3906μs
0x8	256	7813μs
0x9	512	15625μs
0xA	1024	31250μs
0xB	2048	62500μs
0xC	4096	125000μs
0xD (Default)	8192	250000μs
0xE	16384	500000μs
0xF	32768	1000000μs

Notes:

1. It is critical when using AGC to allow ample startup time to avoid spurious CFD events.
2. When using AGC BW = 0x0, the minimum startup time is 6.25 ms.

Bit 7 – ONDEMAND On Demand Control

The ONDEMAND operation mode allows the XOSC to be enabled or disabled depending on peripheral clock requests.

Note: The XOSC is not running if no peripheral is requesting the clock source.

If ONDEMAND is set, the XOSC will only be running when requested by a peripheral and enabled (XOSCTRLA.ENABLE = 1). If there is no peripheral requesting the XOSC's clock source, the XOSC will be in a disabled state. If ONDEMAND is disabled, the XOSC will always be running when enabled (XOSCTRLA.ENABLE = 1). In Standby Sleep mode, the ONDEMAND operation is still active. This bit is XOSCCTRLA.ENABLE protected and cannot be updated if XOSCCTRLA.ENABLE = 1.

Value	Description
0	The XOSC is always on.
1	The XOSC is running when a peripheral is requesting the XOSC to be used as a clock source.

Bit 5 – SWBEN XOSC Clock Switch Back Enable

This bit controls the XOSC output clock switch back to the external clock or crystal oscillator in case of clock recovery.

Note: The SWBEN bit is also cleared when the Clock Failure Detector is disabled (CFDEN = 0).

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 4 – CFDEN Clock Failure Detector Enable

This bit controls the XOSC clock failure detector and is enable protected

Note: After setting CFDEN to enable clock failure detection, STATUS.CLKFAIL will always be set. This first detection must be ignored. Subsequent setting of this bit will indicate actual clock failure events.

Value	Description
0	Clock Failure Detector is disabled.
1	Clock Failure Detector is enabled.

Bit 3 – XTALEN Crystal Oscillator Enable

This bit controls the connections between the I/O pads and the external clock or crystal oscillator XOSC.

Notes:

1. If XOSCCTRLA.XTALEN = 0 then XOSCCTRLA.AGC = 1 is not permitted.
2. This bit is XOSCCTRLA.ENABLE protected and cannot be updated if XOSCCTRLA.ENABLE = 1.
3. If XOSCCTRLA.XTALEN = 0 then XOSCCTRLB.GMAN (User Manual Gain control) bits are ignored.

Value	Description
0	External clock oscillator connected on XIN. XOUT can be used as general-purpose I/O.
1	Crystal connected to XIN and XOUT.

Bit 2 – AGC Auto Gain Control Loop Enable

Notes:

- If XOSCCTRLA.XTALEN = 0 then XOSCCTRLA.AGC = 1 is not permitted.
- This bit is XOSCCTRLA.ENABLE protected and cannot be updated if XOSCCTRLA.ENABLE = 1.
- If AGC is enabled, XOSCCTRLB.GMAN (User Manual Gain control) bits are ignored.
- When the XOSCCTRLA.AGC = 1, the Primary Oscillator will automatically do a linear search to find the lowest power/gain setting to guarantee stable oscillation with the user's crystal.

Value	Description
0	The oscillator auto gain control loop is disabled.
1	The oscillator auto gain control loop is enabled.

Bit 1 – ENABLE Oscillator Enable

Value	Description
0	The oscillator XOSC is disabled.
1	The oscillator XOSC is enabled.

19.7.7. External Multipurpose Crystal Oscillator Control B

Name: XOSCCTRLB
Offset: 0x18
Reset: 0x00000000
Property: PAC Write-Protection

Table 19-10. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	WRTLOCK							
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
				GBW[1:0]		GRES	GMAN[1:0]	
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bit 31 – WRTLOCK Write Lock for CTRLB register
Note: Once the WRTLOCK bit is set, it can only be cleared by a reset.

Value	Description
0	The XOSCCTRLB register can be modified by a system write.
1	The XOSCCTRLB register is write protected.

Bits 4:3 – GBW[1:0] AGC Gain Bandwidth (Gain Step Loop Delay)
Notes:
1. These bits are ignored if XOSCCTRLA.AGC=0, only used if XOSCCTRLA.AGC=1.
2. The default setting should meet the vast majority of user crystal requirements. Internally, there are a maximum of 16 and a minimum of one AGC linear gain search steps the logic may utilize before locking. A lock will occur when the crystal is oscillating and the amplitude of the crystal signal is between a max and min fixed internal threshold for a fixed number of valid oscillator cycles. The GBW is the time for each of the possible AGC search steps settling time to allow the crystal to startup and amplitude to stabilize before determining if a lock is true or to continue to search for the required gain. The GBW bits represent a balance between start-up time and crystal power optimization. The lower the GBW delay time the faster the crystal start-up time but potentially at a higher crystal power level. The higher the GBW delay time the slower the crystal start-up time but with a better crystal power optimization level (i.e., less power).

3. Use of resonators with this product have not been confirmed - use at your own discretion. When using a resonator, due to their long start-up times, it may be necessary to use a longer AGC GBW step settling time.

Value	Description
11	Reserved
10	Reserved
01	Update loop every ~25ms
00	Update loop every ~6.25ms (Default)

Bit 2 – GRES Internal XOSC Gain Resistor

Important: If XOSCCTRLA.XTALEN = 0, clock oscillator instead of a crystal, then this bit is ignored. In all other configurations XOSCCTRLA.AGC=”x” or GMAN =0bxx this bit SHOULD always be set, XOSCCTRLB. GRES=1, by the user except in the case where the user is utilizing an external gain resistor between the XOSC XIN and XOUT pins.

Value	Description
0	Disconnect internal XOSC shunt Gain resistor (Default)
1	Use internal XOSC shunt Gain resistor

Bits 1:0 – GMAN[1:0] Manual User Crystal Control Gain Setting (XOSCCTRLA.AGC=0)
Gm3 >Gm3 >Gm1>Gm0
Note: These bits are ignored if XOSCCTRLA.AGC=1.

Value	Description
11	Gain_3
10	Gain_2
01	Gain_1
00	Gain_0 (Default)

19.7.8. DFLL48M Control A

Name: DFLLCTRLA
Offset: 0x2C
Reset: 0x00000082
Property: PAC Write-Protection

Table 19-11. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

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

Bit 7 – ONDEMAND On Demand Control

The ONDEMAND operation mode allows the DFLL to be enabled or disabled depending on peripheral clock requests.

Note: If ONDEMAND is set, the DFLL will only be running when requested by a peripheral and enabled (DFLLTRLA. ENABLE=1). If there is no peripheral requesting the DFLL's clock source, the DFLL will be in a disabled state. If ONDEMAND is disabled the DFLL will always be running when enabled (DFLLTRLA.ENABLE=1). In standby sleep mode, the ONDEMAND operation is still active This bit is DFLLCTRLA.ENABLE protected and cannot be updated if DFLLCTRLA.ENABLE=1.

Value	Description
0	The DFLL is always on.
1	The DFLL is running when a peripheral is requesting the DFLL to be used as a clock source. The DFLL is not running if no peripheral is requesting the clock source.

Bit 3 – LOWFREQ Low Frequency Mode

Note: This bit is DFLLCTRLA.ENABLE protected and cannot be updated if DFLLCTRLA.ENABLE=1.

Value	Description
0	The DFLL48M oscillator operates at high frequency.
1	The DFLL48M oscillator operates at low frequency.

Bit 2 – WRTLOCK Write lock

Note: Once the WRTLOCK bit is set, it can only be cleared by a reset.

Value	Description
0	The DFLLCTRLA and DFLLCTRLB registers can be modified by a system write.
1	The DFLLCTRLA and DFLLCTRLB registers are write protected.

Bit 1 – ENABLE DFLL48M Enable

Note: This bit is write-synchronized: Due to synchronization, there is delay from updating the register until the peripheral is enabled/disabled. The value written to DFLLCTRLA.ENABLE will read back immediately after written.

Value	Description
0	The DFLL48M oscillator is disabled.
1	The DFLL48M oscillator is enabled.

19.7.9. DFLL48M Control B

Name: DFLLCTRLB
Offset: 0x30
Reset: 0x00000000
Property: PAC Write-Protected, Write-Synchronized

Note: In the period of 80-100 us between lock and stabilization, DFLL accuracy will be limited to +/-1.5%. After stabilization has been achieved, the accuracy will be +/-0.25%. Disabling Quick Lock will eliminate this period of inaccuracy.

Table 19-12. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	WAITLOCK		QLDIS	CCDIS		LLAW	STABLE	LOOPEN
Access	R/W		R/W	R/W		R/W	R/W	R/W
Reset	0		0	0		0	0	0

Bit 7 – WAITLOCK Wait Lock

This bit controls the DFLL48M output clock, depending on lock status:

Value	Description
0	Output clock before the DFLL is locked.
1	Output clock when DFLL is locked.

Bit 5 – QLDIS Quick Lock Disable

Value	Description
0	Quick Lock is enabled.
1	Quick Lock is disabled.

Bit 4 – CCDIS Chill Cycle Disable

Value	Description
0	Chill Cycle is enabled.
1	Chill Cycle is disabled.

Bit 2 – 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 1 – STABLE Stable DFLL48M Frequency

Value	Description
0	Tune register tracks changes in output frequency.
1	Tune calibration register value will be fixed after a lock.

Bit 0 – LOOPEN Operating Mode Selection

Value	Description
0	The DFLL operates in open-loop operation.
1	The DFLL operates in closed-loop operation.

19.7.10. DFLL Tune

Name: DFLLTUNE
Offset: 0x34
Reset: 0x00000000
Property: PACWrite-Protection, Write-Synchronized, Read-Synchronized

Table 19-13. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

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

Bits 6:0 – TUNE[6:0] DFLL48M Tune Value
Sets the value of the Tune Calibration register.
Note: In closed-loop mode, this field is read-only.

Step	TUNE[6:0]	% Delta/step
+63	0b011 1111	+9.45%
•••	•••	•••
+1	0b000 0001	+0.15%
0	0b000 0000 / 0b111 1111	0%
-1	0b111 1110	-0.15%
•••	•••	•••
-63	0b100 0000	-9.45%

Note:
1. % Delta value is rounded to two decimal places.

19.7.11. DFLL48M Diff

Name: DFLLDIFF
Offset: 0x38
Reset: 0x00000000
Property: Read-Synchronized

Table 19-14. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
	DIFF[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
	DIFF[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 – DIFF[15:0] Multiplication Ratio Difference

In closed-loop mode (DFLLCTRLB.LOOPEN is written to one), this bit group indicates the difference between the ideal number of DFLL48M cycles and the counted number of cycles. This value is not updated in open loop mode and should be considered invalid in that case.

19.7.12. DFLL48M Multiplier

Name: DFLLMUL
Offset: 0x3C
Reset: 0x00000000
Property: PAC Write-Protection, Write-Synchronized

Table 19-15. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
		STEP[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
	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 22:16 – STEP[6:0] Tune Maximum Step

This bit group indicates the maximum step size allowed during tune adjustment in closed-loop mode. When adjusting to a new frequency, the expected output frequency overshoot depends on this step size.

Note: STEP[6:0] is nothing more than the first tuner adjustment value from where the tune search will start. Depending if the current DFLL frequency is slower of faster than the targeted frequency, the next tune register value is *tune+step* or *tune-step*. Each time the current frequency changes from slower to faster or faster to slower, step is divided by 2 until the minimum value of 1.

Example 1:
If STEP[6:0] = 64, dichotomic search will be:

- step1 = +/-64
- step2 = +/-32
- step3 = +/-16
- step4 = +/-8
- step5 = +/-4
- step6 = +/-2
- step7 = +/-1

Example 2:
If STEP[6:0] = 15, dichotomic search will be:

- step1 = +/-15
- step2 = +/-7
- step3 = +/-3
- step4 = +/-1

Bits 15:0 – MUL[15:0] DFLL Multiply Factor
This field determines the ratio of the CLK_DFLL output frequency to the CLK_DFLL_REF input frequency. Writing to the MUL bits will cause the lock to be lost and the DFLLTUNE.TUNE register value to be reset to its midpoint, 0b000_0000.
Example: CLK_DFLL_REF = XTAL, 32.768 kHz, 100 ppm, DFLLMUL.MUL[15:0] = 1464 = 0x5B8, then DFLL48 = 47.97 MHz.

19.7.13. PLL0 Control

Name: PLL0CTRL
Offset: 0x40
Reset: 0x00000000
Property: PAC Write-Protection

Table 19-16. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
				BWSEL[2:0]			REFSEL[2:0]	
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

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

Bits 13:11 – BWSEL[2:0] Bandwidth Selection
These bits select the PLL closed loop filter bandwidth, depending on the frequency after the reference divider F_{PFD} as shown in the table below. Selecting the correct filter bandwidth is important to operate the PLL'VCO in its best range.

Table 19-17. PLL0 BWSEL Filter

F_{PFD}	BWSEL[2:0]
Reserved	0b000
$4MHz \le F_{PFD} < 10MHz$	0b001
$10MHz \le F_{PFD} < 20MHz$	0b010
$20MHz \le F_{PFD} < 30MHz$	0b011
$30MHz \le F_{PFD} < 60MHz$	0b100
Reserved	0b101 – 0b111

Notes:

1. F_{PFD} is the frequency of the reference clock divided by the PLL0 reference divider PLLREFDIV.REFDIV. These bits are PLLCTRL0.ENABLE protected and cannot be updated if PLLCTRL0.ENABLE =1.
2. At elevated temperatures, the effective range of the Bandwidth setting will skew higher. Depending on the input frequency and operating temperature, it may be optimal to change the BWSEL setting to the next higher value.

Bits 10:8 – REFSEL[2:0] Reference Selection

These bits select the PLL0 clock reference, as shown in the table below.

REFSEL[2:0]	Selectedsource	Description
0x0 [2,3]	GCLK	DedicatedGCLK_PLL0_REF clock reference
0x1	XOSC	XOSC clock reference
0x2	DFLL48M	DFLL48Mclock reference
0x3- 0x7	n/a	Reserved

Important: If GCLK source is PLL0, then users should not use GCLK_PLL0_REF as input clock source to PLL0. It will create a circular reference, an unstable clock, and unexpected behavior.

Notes:

1. These bits are PLLCTRL0.ENABLE protected and cannot be updated if PLLCTRL0.ENABLE = 1.
2. The recommended clock sources for PLL0 are XOSC and DFLL48M. Using the GLCK as a source is not recommended.

Bit 7 – ONDEMAND On Demand Control

The ONDEMAND operation mode allows the PLL to be enabled or disabled depending on peripheral clock requests.
Note: If ONDEMAND is set, the PLL will only be running when requested by a peripheral and enabled (PLLCTRL. ENABLE = 1). If there is no peripheral requesting the PLL's clock source, the PLL will be in a disabled state. If ONDEMAND is disabled the PLL will always be running when enabled (PLLCTRL.ENABLE = 1). In Standby Sleep mode, the ONDEMAND operation is still active.

This bit is PLLCTRL0.ENABLE protected and cannot be updated if PLLCTRL0.ENABLE =1.

Value	Description
0	The PLL0 is always on.
1	The PLL0 is running when a peripheral is requesting the PLL to be used as a clock source. The PLL is not running if no peripheral is requesting the PLL clock source.

Bit 2 – WRTLOCK Write Lock

Note: Once the WRTLOCK bit is set, it can only be cleared by a reset.

Value	Description
0	The PLLCTRL, PLLFBDIV, PLLREFDIV and PLLPOSTDIVA/B registers can be modified by a system write.
1	The PLLCTRL, PLLFBDIV, PLLREFDIV and PLLPOSTDIVA/B registers are write protected, except for bits PLLPOSTDIVA.OUTENn.

Bit 1 – ENABLE PLL0 Enable

Value	Description
0	The PLL0 oscillator is disabled.
1	The PLL0 oscillator is enabled.

19.7.14. PLL0 Feed-Back Divider

Name: PLL0FBDIV
Offset: 0x44
Reset: 0x00000000
Property: PAC Write-Protection

Table 19-18. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
							FBDIV[9:8]	
Access							R/W	R/W
Reset							0	0

Bit	7	6	5	4	3	2	1	0
	FBDIV[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 9:0 – FBDIV[9:0] PLL0 Feed-Back Divider Factor

This field determines the ratio of the PLL's VCO output frequency to the PLL Reference input frequency. Writing to the FBDIV bits will cause lock to be lost.

The value of FBDIV, (i.e., PLLFBDIV) must be within the range 21 ≤ FBDIV ≤ 1023.

Note: Note: The frequency of the Voltage Controlled Oscillator (VCO) giving the PLL0 oscillation is given by the formula:

FVCO = FCKR * (FBDIV / REFDIV), (i.e., must be between 800 MHz and 1600 MHz).

f_{CKR} , REFDIV and FBDIV must be selected to satisfy this condition.

19.7.15. PLL0 Reference Divider

Name: PLL0REFDIV
Offset: 0x48
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Table 19-19. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			REFDIV[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 – REFDIV[5:0] PLL0 Reference Division Factor

This field determines the division factor of the PLL0 input reference frequency. Writing to the REFDIV bits will cause lock to be lost. REFDIV value must be in the range of $1 \leq REFDIV \leq 63$.

The frequency after the reference divider (FPFD) is given by the formula:

$F_{PFD} = F_{CKR}$ / REFDIV (i.e., F_{PFD} must always be between 4 MHz to 48 MHz.

19.7.16. PLL0 Post Output Clock Divider A

Name: PLL0POSTDIVA
Offset: 0x4C
Reset: 0x20202020
Property: PAC Write-Protection

> **Important:** The PLL0 frequency cannot be changed on the fly while it's the active enabled clock to the system.

PLL0 Output Clocks:

1. CLK_PLL0_CLKOUTn where n = 0.
2. CLK_PLL0_CLKOUTn where n = 1.
3. CLK_PLL0_CLKOUTn where n = 2.
4. CLK_PLL0_CLKOUTn where n = 3.

Table 19-20. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	OUTEN3		POSTDIV3[5:0]					
Access	R/W		R/W	R/W	R/W	R/W	R/W	R/W
Reset	0		1	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	OUTEN2		POSTDIV2[5:0]					
Access	R/W		R/W	R/W	R/W	R/W	R/W	R/W
Reset	0		1	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	OUTEN1		POSTDIV1[5:0]					
Access	R/W		R/W	R/W	R/W	R/W	R/W	R/W
Reset	0		1	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	OUTEN0		POSTDIV0[5:0]					
Access	R/W		R/W	R/W	R/W	R/W	R/W	R/W
Reset	0		1	0	0	0	0	0

Bits 7, 15, 23, 31 – OUTENn CLK_PLL0_CLKOUTn Output Enable

Value	Description
0	CLK_PLL0_CLKOUTn Output Disabled
1	CLK_PLL0_CLKOUTn Output Enabled

Bits 0:5, 8:13, 16:21, 24:29 – POSTDIVn PLL0 FVCO Output Clock Division Factor
This field determines the division factor of the PLL0 F_{VCO} output that creates FCLK_PLL0 and CLK_PLL0_CLKOUTn. The POSTDIV value must be between 1 ≤ POSTDIV ≤ 63.

Notes:

1. (F_{VCO} / POSTDIV) > FCLK_PLL0 > CLK_PLL0_CLKOUTn.
2. PLL0 must be disabled before making changes to POSTDIVn values.
3. It is not recommended to set the POSTDIV registers while the PLL is active and stable.

19.7.17. PLL0 Post Output Clock Divider B

Name: PLL0POSTDIVB
Offset: 0x50
Reset: 0x00002020
Property: PAC Write-Protection

Table 19-21. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
	OUTEN5		POSTDIV5[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	6	5	4	3	2	1	0
	OUTEN4		POSTDIV4[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

Bits 7, 15 – OUTENn CLK_PLL0_CLKOUTn Output Enable

Value	Description
0	CLK_PLL0_CLKOUTn Output Disabled
1	CLK_PLL0_CLKOUTn Output Enabled

Bits 0:5, 8:13 – POSTDIVn PLL0 FVCO Output Clock Division Factor

This field determines the division factor of the PLL0 F_{VCO} output that creates FCLK_PLL0 and CLK_PLL0_CLKOUTn. POSTDIV value must be between 1 ≤ POSTDIV ≤ 63.

Notes:

1. (FVCO/POSTDIV) > FCLK_PLL0 > CLK_PLL0_CLKOUTn.
2. PLL0 must be disabled before making changes to POSTDIVn values.
3. It is not recommended to set the POSTDIV registers while the PLL is active and stable.

19.7.18. Fractional Divider 0

Name: FRACDIV0
Offset: 0x6C
Reset: 0x00200000
Property: PAC Write-Protection, Write-Synchronized

Notes:

1. FCLK_PLLFRC0 = (F_{CLK_PLL0} / (2 x (INTDIV + (REMDIV / 512)))).
2. The maximum permitted fractional output frequency, FCLK_PLLFRC0, must always be limited to 200 MHz by the user.
3. Setting both INTDIV and REMDIV = 0 will yield FCLK_PLLFRC0 = F_{CLK_PLL0} which effectively bypasses the fractional divider module in which case the user must limit the output of PLL0 to the fractional divider module to FPLL0 = F_{CLK_PLL0} = 200MHz max.

Table 19-22. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
		INTDIV[14:8]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	1	0

Bit	23	22	21	20	19	18	17	16
	INTDIV[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
	REMDIV[8:1]							
Access	R/W	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
	REMDIV[0]							
Access	R/W							
Reset	0							

Bits 30:16 – INTDIV[14:0] Frequency Division Factor Integer Part
This field determines the integer part of the frequency divider and must be between 0 ≤ INTDIV ≤ 32767.
INTDIV (Default) = 0x20 = 32 decimal.

Bits 15:7 – REMDIV[8:0] Frequency Division Factor Remainder Part
This field determines the reminder part of the frequency divider and must be between 0 ≤ REMDIV ≤ 511.

19.7.19. PLL Synchronization Busy

Name: SYNCBUSY
Offset: 0x78
Reset: 0x00000000
Property: -

Table 19-23. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
		FRACDIV0	DFLLMUL	DFLLDIFF	DFLLTUNE	DFLLCTRLB	DFLLENABLE	
Access		R	R	R	R	R	R	
Reset		0	0	0	0	0	0	

Bit 6 – FRACDIV0 FRACDIV0 Synchronization Busy
This bit is cleared when the synchronization of FRACDIV0 register between the clock domains is complete. This bit is set when the synchronization of FRACDIV0 register between clock domains is started.
Note: The FRACDIV0 synchronization only applies for write operations.

Bit 5 – DFLLMUL DFLLMUL Synchronization Busy
This bit is cleared when the synchronization of DFLLMUL register between the clock domains is complete. This bit is set when the synchronization of DFLLMUL register between clock domains is started.
Note: The DFLLMUL synchronization only applies for write operations.

Bit 4 – DFLLDIFF DFLLDIFF Synchronization Busy
This bit is cleared when the synchronization of DFLLDIFF register between the clock domains is complete. This bit is set when the synchronization of DFLLDIFF register between clock domains is started.
Note: The DFLLDIFF synchronization only applies for read operations.

Bit 3 – DFLLTUNE DFLLTUNE Synchronization Busy

This bit is cleared when the synchronization of DFLLTUNE register between the clock domains is complete. This bit is set when the synchronization of DFLLTUNE register between clock domains is started.

Note: The DFLLTUNE synchronization applies for read and write operations.

Bit 2 – DFLLCTRLB DFLLCTRLB Synchronization Busy

This bit is cleared when the synchronization of DFLLCTRLB register between the clock domains is complete. This bit is set when the synchronization of DFLLCTRLB register between clock domains is started.

Note: The DFLLCTRLB synchronization only applies for write operations.

Bit 1 – DFLLENABLE DFLL48M Enable Synchronization Busy

This bit is cleared when the synchronization of the DFLLCTRLA.ENABLE register bit between the clock domains is complete.

Note: This bit is set when the synchronization of the DFLLCTRLA.ENABLE register bit between clock domains is started.

20. Generic Clock Controller (GCLK)

20.1. Overview

Depending on the application, peripherals may require specific clock frequencies to operate correctly. The Generic Clock controller (GCLK) features 12 Generic Clock Generators [15:0] that can provide a wide range of clock frequencies.

Generators can be set to use different external or internal oscillators as source. The clock of each Generator can be divided according to the user's requirements. The outputs from the Generators are used as sources for the 48 Peripheral Channels, which provide the Generic Clock (GCLK_PERIPH) to the peripheral modules, as shown in Figure 20-2. The number of Peripheral Clocks, GCLK_PERIPH, depends on how many peripherals the device has.
Note: The Generator 0 is always the direct source of the GCLK_MAIN signal.

20.2. Features

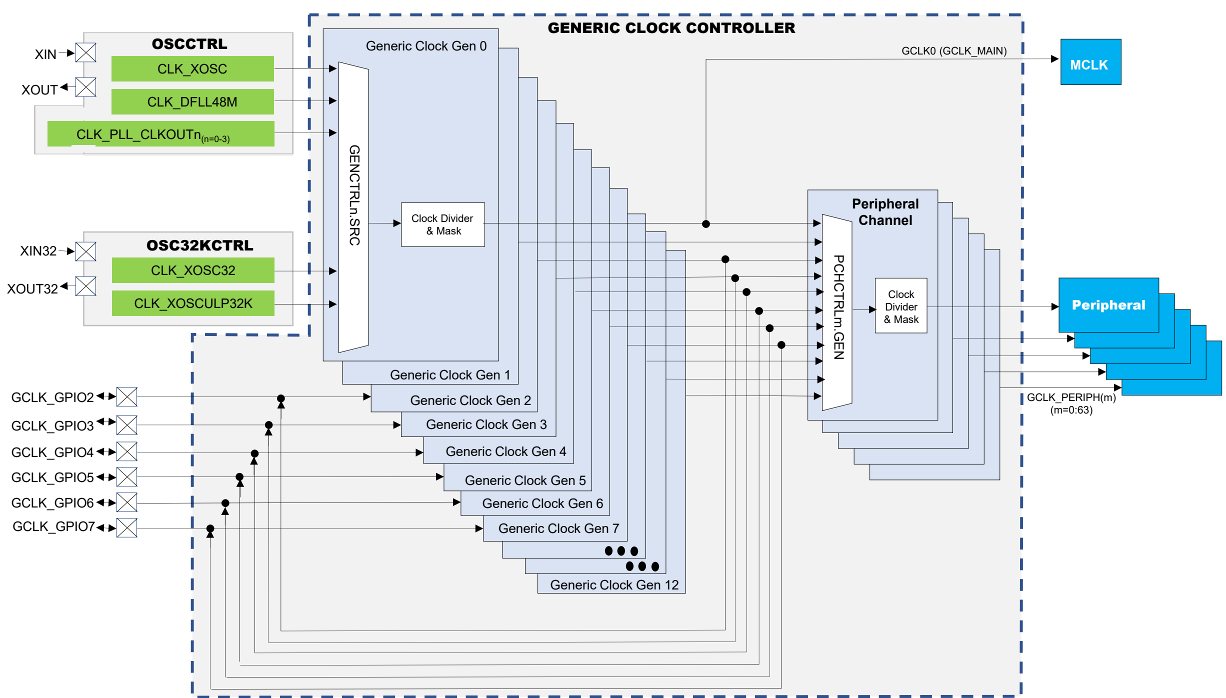
The following are key features of the GCLK module:

- Provides a device-defined, configurable number of Peripheral Channel clocks
- Wide frequency range:
  - Various clock sources
  - Embedded dividers

20.3. Block Diagram

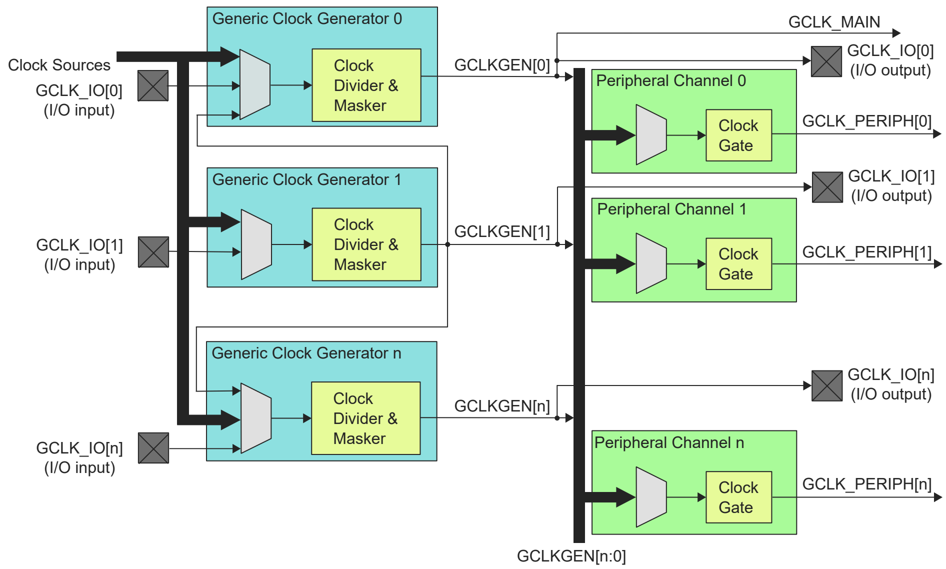
The generation of Peripheral Clock signals (GCLK_PERIPH) and the Main Clock (GCLK_MAIN) can be seen in Device Clocking Diagram.

Figure 20-1. Device Clocking Diagram



The GCLK block diagram is shown below:

Figure 20-2. Generic Clock Controller Detailed Block Diagram



20.4. Signal Description

Table 20-1. GCLK External Signal Descriptions

Signal Name	Type	Description
GCLK_IO[7:2]	Digital I/O	• External Clock source for Generators when input • Generic Clock signal when output

Notes:
1. One signal can be mapped on several pins.
2. GCLK_IOn is available only to the corresponding GCLK GENn, (i.e., GCLK_IO[3] is only available to GCLK GEN3 and so on).

20.5. Peripheral Dependencies

Peripheral Name	Base Address	PAC Peripheral ID (PAC.WRCTRL.PERIDx)	Power Domain
GCLK0	0x4401_0000	8	VDDREG

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

20.5.1. I/O Lines

Using the GCLK I/O lines requires the I/O pins to be configured.

20.5.2. Power Management

The GCLK can operate in sleep modes, if required. Refer to the Sleep mode description in the Power Manager (PM) section.

20.5.3. Clocks

The GCLK bus clock (CLK_GCLK_APB) can be enabled and disabled in the Main Clock Controller (MCLK).

20.5.4. 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 a peripheral to be periodically serviced by the CPU through interrupts or similar, improper operation or data loss may result during debugging.

20.5.5. Register Access Protection

All registers with write access can be optionally write-protected by the Peripheral Access Controller (PAC).

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

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

20.5.6. PIC32CK SG/GC TrustZone Specific Register Access Protection

On PIC32CK SG/GC devices, this peripheral has different access permissions depending on PAC Security Attribution (Secure or Non-Secure):

- 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

Refer to Implementation Defined Attribution Unit (IDAU) for additional information.

20.6. Functional Description

20.6.1. Principle of Operation

The GCLK module is comprised of 12 Generic Clock Generators (Generators) sourcing up to 48 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.

20.6.2. GCLK Basic Operation

20.6.2.1. 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 (GENCTRLn.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", (n=0-11), is configured by one dedicated register GENCTRLn.

Note: Each Peripheral Channel "m", (m=0-47), is configured by one dedicated register PCHCTRLm.

20.6.2.2. 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 Configuration Lock.

20.6.2.3. Generic Clock Generator

Each Generator (GCLK_GEN) can be set to run from one of 14 different clock sources, GENCTRLn.SRC, except GCLK_GEN[1], which can be set to run from one of 13 sources, GCLK_GEN1 is invalid for GCLK_GEN1. GCLK_GEN[1] is the only Generator that can be selected as source to others Generators.

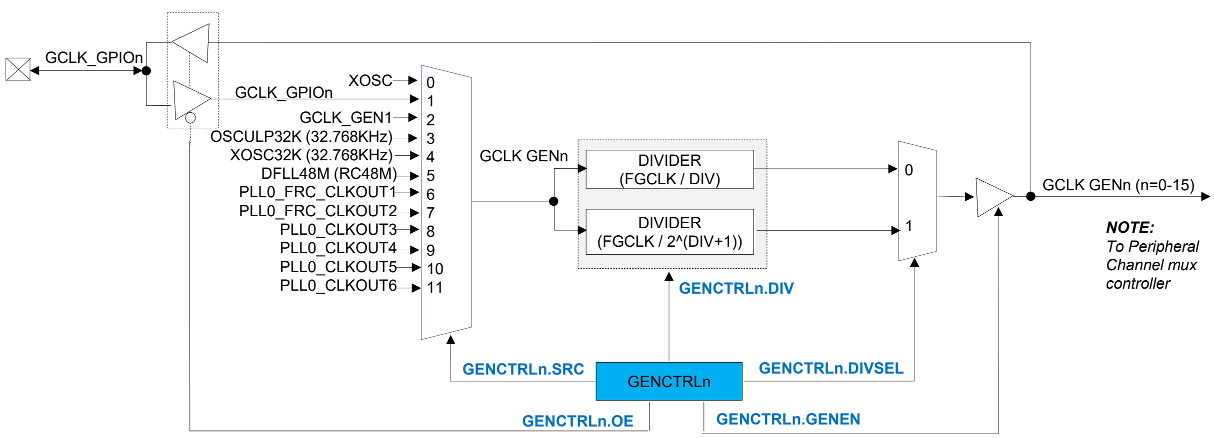
With respect to the GPIO[x] clock selection, GENCTRLx.SRC=0x01, each generator GCLK_GEN[x] can be connected to only one specific pin GCLK_IO[x] in a one-to-one GCLK_GENx to GCLK_GPIOx relationship. A pin GCLK_IO[x] can be set either to act as source to GCLK_GEN[x] or to output the clock signal generated by GCLK_GEN[x].

The selected source can be divided. Each Generator can be enabled or disabled independently.

Each GCLK_GEN clock signal can then be used as clock source for Peripheral Channels. Each Generator output is allocated to one or several Peripherals.

GCLK_GEN[0] is used as GCLK_MAIN for the synchronous clock controller inside the Main Clock Controller. Refer to the Main Clock Controller description for details on the synchronous clock generation.

Figure 20-3. Generic Clock Generator



20.6.2.4. Enabling a Generator

A Generator is enabled by writing a '1' to the Generator Enable bit in the Generator Control register (GENCTRLn.GENEN=1).

20.6.2.5. Disabling a Generator

A Generator is disabled by writing a '0' to GENCTRLn.GENEN. When GENCTRLn.GENEN=0, the GCLK_GEN[n] clock is disabled and gated off.

20.6.2.6. 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. Accordingly the SYNCBUSY.GENCTRLn bit in SYNCBUSY register will remain '1' until the switch operation is completed.

The available clock sources are device dependent (usually the oscillators, RC oscillators, PLL). Only clock Generator 1 can be used as a common source for all other generators.

20.6.2.7. 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 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 GCLK Generator source input clock, defined by GENCTRLx.SRC, can be divided as defined in GENCTRLx.DIV.

20.6.2.8. 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.

20.6.2.9. External Clock

The output clock (GCLK_GEN) of each Generator can be sent to I/O pins (GCLK_IO).

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_GEN 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.

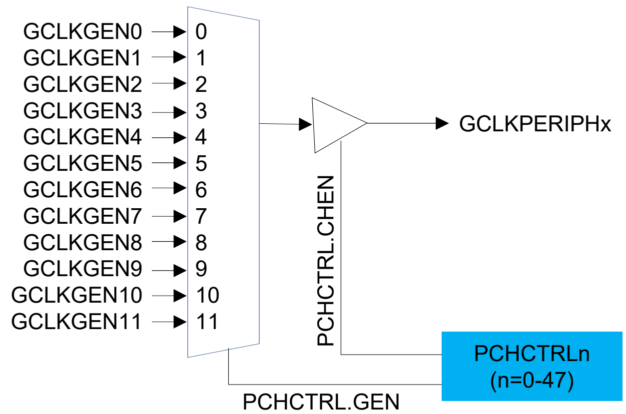
In Standby mode, if the clock is output (GENCTRLn.OE=1), the clock on the I/O pin is frozen to the OOV value if the Run In Standby bit of the Generic Control register (GENCTRLn.RUNSTDBY) is zero.
Note: With GENCTRLn.OE=1 and RUNSTDBY=0, entering the Standby mode can take longer due to a clock source dependent delay between turning off Power Domain 1 and 2. The maximum delay can be equal to the clock source period multiplied by the division factor.

Note: With GENCTRLn.OE=1 and RUNSTDBY=0, entering the Standby mode can take longer due to a clock source dependent delay. The maximum delay can be equal to the clock source period multiplied by the division factor.

If GENCTRLn.RUNSTDBY is '1', the GCLKGEN clock is kept running and output to the I/O pin.

20.6.3. Peripheral Clock

Figure 20-4. Peripheral Clock



20.6.3.1. 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.

Table 20-2. PCHCTRL (Index) GCLK Mapping

Target Destination	GCLK Name	PCHCTRL (Index)
OSCCTRL	GCLK_OSCCTRL_DFLL48	0
	GCLK_OSCCTRL_PLL	1
FREQM	GCLK_FREQM_MSR0	2
	GCLK_FREQM_MSR1	3
	GCLK_FREQM_REF	4
EIC	GCLK_EIC	5
EVSTS	GCLK_EVSYS_CH0	6
	GCLK_EVSYS_CH1	7
	GCLK_EVSYS_CH2	8
	GCLK_EVSYS_CH3	9
	GCLK_EVSYS_CH4	10
	GCLK_EVSYS_CH5	11
	GCLK_EVSYS_CH6	12
	GCLK_EVSYS_CH7	13
	GCLK_EVSYS_CH8	14
	GCLK_EVSYS_CH9	15
	GCLK_EVSYS_CH10	16
	GCLK_EVSYS_CH11	17
SERCOMm, m = 0...7 SDMMCn, n = 0,1	GCLK_SERCOMm_SLOW, m = 0...7 SDMMCn_SLOW, n=0,1	18
SERCOM0	GCLK_SERCOM0_CORE	19
SERCOM1	GCLK_SERCOM1_CORE	20

Table 20-2. PCHCTRL (Index) GCLK Mapping (continued)

Target Destination	GCLK Name	PCHCTRL (Index)
SERCOM2	GCLK_SERCOM2_CORE	21
SERCOM3	GCLK_SERCOM3_CORE	22
TCC0,TCC1	GLCK_TCC0, GCLK_TCC1	23
TCC2,TCC3	GLCK_TCC2, GCLK_TCC3	24
SERCOM4	GCLK_SERCOM4_CORE	25
SERCOM5	GCLK_SERCOM5_CORE	26
SERCOM6	GCLK_SERCOM6_CORE	27
SERCOM7	GCLK_SERCOM7_CORE	28
TCC4	GCLK_TCC4	29
TCC5	GCLK_TCC5	30
TCC6	GCLK_TCC6	31
TCC7	GCLK_TCC7	32
ADC	GCLK_ADC	33
AC	GCLK_AC	34
PTC	GCLK_PTC	35
SPI_IXS	GCLK_SPI_IXS	36
CCL	GCLK_CCL	37
PDEC	GCLK_PDEC	38
CAN0	GCLK_CAN0	39
CAN1	GCLK_CAN1	40
ETH	GCLK_ETH_TX	41
	GCLK_ETH_TSU	42
SQI	GCLK_SQI	43
SDMMC0	GLCK_SDMMC0	44
SDMMC1	GLCK_SDMMC1	45
USB	GCLK_USB	46
CPU0	GCLK_CPU0_TRACE	47

20.6.3.2. 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.

20.6.3.3. 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 as ‘1’.

20.6.3.4. 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 any 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 any 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.

20.6.4. GCLK Additional Features

20.6.4.1. 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 required.

Refer to GENCTRLn.SRC for details on GENCTRLn reset.

Refer to PCHCTRLm.SRC for details on PCHCTRLm reset.

20.6.5. Sleep Mode Operation

20.6.5.1. 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. Refer to External Clock for details.

20.6.5.2. Minimize Power Consumption in Standby Mode

The following table identifies when a Clock Generator is off in Standby mode, minimizing the power consumption:

Table 20-3. 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

20.6.5.3. Entering Standby Mode

There may occur a delay when the device is put into Standby, until all clocks are gated 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 off properly. The duration of this verification is frequency-dependent.

20.6.6. Synchronization

Due to asynchronicity between the main clock domain and the peripheral clock domains, some registers need to be synchronized when written or read.

An exception is the Channel Enable bit in the Peripheral Channel Control registers (PCHCTRLm.CHEN). When changing this bit, the bit value must be read-back to ensure the synchronization is complete and to assert glitch free internal operation. Note that changing the bit value under ongoing synchronization will *not* generate an error.

The following registers are synchronized when written:

- Generic Clock Generator Control register (GENCTRLn)
- Control A register (CTRLA)

Required write synchronization is denoted by the “Write Synchronized” property in the register description.

20.7. Register Summary

For descriptions and definitions of both Register and bitfield properties, refer to Register Properties.

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			GENCTRL11	GENCTRL10	GENCTRL9	GENCTRL8	GENCTRL7	GENCTRL6
		7:0	GENCTRL5	GENCTRL4	GENCTRL3	GENCTRL2	GENCTRL1	GENCTRL0		SWRST
0x08 ... 0x1F	Reserved									
0x20	GENCTRL0	31:24	DIV[15:8]							
		23:16	DIV[7:0]							
		15:8			RUNSTDBY	DIVSEL	OE	OOV	IDC	GENEN
		7:0				SRC[4:0]				
...										
0x4C	GENCTRL11	31:24	DIV[15:8]							
		23:16	DIV[7:0]							
		15:8			RUNSTDBY	DIVSEL	OE	OOV	IDC	GENEN
		7:0				SRC[4:0]				
0x50 ... 0x7F	Reserved									
0x80	PCHCTRL0	31:24								
		23:16								
		15:8								
		7:0	WRTLOCK	CHEN			GEN[3:0]			
...										
0xFC	PCHCTRL31	31:24								
		23:16								
		15:8								
		7:0	WRTLOCK	CHEN			GEN[3:0]			

20.7.1. Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized

Table 20-4. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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 any reset, except for generic clocks and associated Generators that have their WRTLOCK bit in PCHCTRLm set to 1.
Refer to GENCTRL Reset Value for details on GENCTRL register reset.
Refer to PCHCTRL Reset Value for details on PCHCTRL register reset.
Due to synchronization, there is a waiting period between setting CTRLA.SWRST and a completed Reset. CTRLA.SWRST and SYNCBUSY.SWRST will both be cleared when the reset is complete.

Value	Description
0	There is no Reset operation ongoing.
1	A Reset operation is ongoing.

20.7.2. Synchronization Busy

Name: SYNCBUSY
Offset: 0x04
Reset: 0x00000000
Property: –

Table 20-5. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
			GENCTRL11	GENCTRL10	GENCTRL9	GENCTRL8	GENCTRL7	GENCTRL6
Access			R	R	R	R	R	R
Reset			0	0	0	0	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, 10, 11, 12, 13 – GENCTRLn Generator Control n Synchronization Busy

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.

20.7.3. Generator Control

Name: GENCTRLn
Offset: 0x20 + n*0x04 [n=0..11]
Reset: 0x00000106
Property: PAC Write-Protection, Write-Synchronized

GENCTRLn controls the settings of Generic Generator n (n=0..11). The reset value is 0x00000105 for Generator n=0, else 0x00000000.

Table 20-6. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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								
Reset			0	0	0	0	0	1

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

Bits 31:16 – DIV[15:0] Division Factor
These bits represent a division value for the corresponding GCLK Generator input source clock defined by GENCTRLn.SRC. The actual division factor is dependent on the state of DIVSEL.

Table 20-7. Division Factor Bits

Generic Clock Generator	Division Factor Bits
Clock Generator [11:0]	8 division factor bits - DIV[7:0]

Notes:
If GENCTRLn.DIVSEL = 0 then FGCLK = GENCTRLn.SRC/DIV.

- If GENCTRLn.DIV is an odd number: then GENCTRLn.IDC must be set to IDC = 1
- If GENCTRLn.DIV is an even number: then GENCTRLn.IDC must be set to IDC = 0

If GENCTRLn.DIVSEL = 1 then FGCLK= GENCTRLn.SRC/2^(DIV+1).

- GENCTRLn.IDC must always be set to IDC = 0

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 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 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 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^(N+1), where N is the Division Factor Bits for the selected generator (refer to GENCTRLn.DIV).

Bit 11 – OE Output Enable

This bit is used to output the Generator clock output to the corresponding pin (GCLK_IO), as long as GCLK_IO is not defined as the Generator source in the GENCTRLn.SRC bit field.

Value	Description
0	No Generator clock signal on pin GCLK_IO.
1	The Generator clock signal is output on the corresponding GCLK_IO, unless GCLK_IO 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) when the Generator is turned off or the OE bit is zero, as long as GCLK_IO is not defined as the Generator source in the GENCTRLn.SRC bit field.

Value	Description
0	The GCLK_IO will be low when generator is turned off or when the OE bit is zero.
1	The GCLK_IO 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.
Note: If DIVSEL = 1 this bit must always be set to IDC = 0.
If DIVSEL = 0 and DIV = odd number then IDC = 1, else if DIV = even number IDC = 0.

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 4:0 – SRC[4:0] Generator Clock Source Selection

These bits select the Generator clock source, as shown in this table.

Table 20-8. Generator Clock Source Selection

Value (GENCTRLn.SRC)	Name	Description
0x00	XOSC	XOSC Crystal/Clock Oscillator
0x01	GCLK_IOn	Generator GPIO input pin
0x02	GCLK_GEN1	Generic clock generator 1 (GCLK1)
0x03	OSCULP32K (32.768KHz)	Internal Ultra-Low Power 32K RC Oscillator
0x04	XOSC32K (32.768KHz)	32 kHz Crystal Oscillator
0x05	DFLL48M	Internal DFLL48M
0x06	PLL0_FRC_CLKOUT0	Digital Phase Lock Loop, PLL0 Fractional Divider Output 0
0x07	PLL0_CLKOUT1	Digital Phase Lock Loop, PLL0 Output 1
0x08	PLL0_CLKOUT2	Digital Phase Lock Loop, PLL0 Output 2
0x09	PLL0_CLKOUT3	Digital Phase Lock Loop, PLL0 Output 3
0x0A	PLL0_CLKOUT4	Digital Phase Lock Loop, PLL0 Output 4
0x0B	PLL0_CLKOUT5	Digital Phase Lock Loop, PLL0 Output 5
0xC-0x1F	Reserved	Reserved

Note: GENCTRL1.SRC = 0x2 is invalid for GCLK_GEN1 only.

Any reset will reset all the GENCTRLn registers. The Reset values of the GENCTRLn registers are shown in table below.

Table 20-9. GENCTRLn Reset Value after a Power Reset

GCLK Generator	Reset Value after a Power Reset
GCLK0 (GENCTRL0)	GCLK.GENCTRL0 = 0x00000105 (DFLL48M, Internal 48MHz RC Oscillator, GCLK0 Enabled)
GCLK1 (GENCTRL1) - GCLK11 (GENCTRL11)	GCLK.GENCTRL1- GCLK.GENCTRL11 = 0x0000000 (XOSC, GCLK1-GCLK11 disabled)

A User Reset will reset the associated GENCTRL register unless the Generator is the source of a locked Peripheral Channel (PCHCTRLm.WRTLOCK = 1). The reset values of the GENCTRL register are as shown in the table below.

Table 20-10. GENCTRLn Reset Value after a User Reset

GCLK Generator	Reset Value after a User Reset
GCLK0 (GENCTRL0)	GCLK.GENCTRL0 = 0x00000105 (DFLL48M, Internal 48 MHz RC Oscillator, GCLK0 Enabled)
GCLK1 (GENCTRL1) - GCLK11 (GENCTRL11)	GCLK.GENCTRL1- GCLK.GENCTRL11 = No change if the generator is used by a Peripheral Channel m with PCHCTRLm.WRTLOCK = 1 else 0x00000000

20.7.4. Peripheral Channel Control

Name: PCHCTRLm
Offset: 0x80 + m*0x04 [m=0..31]
Reset: 0x00000000
Property: PAC Write-Protection

PCHCTRLm controls the settings of Peripheral Channel number m (m=0..47).

Table 20-11. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	WRTLOCK	CHEN			GEN[3:0]			
Access	R/W	R/W			R/W	R/W	R/W	R/W
Reset	0	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.

Value	Description
0	The Peripheral Channel is disabled
1	The Peripheral Channel is enabled

Bits 3:0 – GEN[3:0] 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 20-12. Generator Selection

Value	Description
0x0	Generic Clock Generator 0
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
0x8	Generic Clock Generator 8
0x9	Generic Clock Generator 9
0xA	Generic Clock Generator 10
0xB	Generic Clock Generator 11

Table 20-13. Reset Value after a User Reset or a Power Reset

Reset	PCHCTRLm.GEN	PCHCTRLm.CHEN	PCHCTRLm.WRTLOCK
Power Reset	0x0	0x0	0x0
User Reset	0x0	0x0	0x0

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.
The PCHCTRL register Reset values are shown in the table below, PCHCTRLm Mapping.

Table 20-14. PCHCTRL (Index) GCLK Mapping

Target Destination	GCLK Name	PCHCTRL (Index)
OSCCTRL	GCLK_OSCCTRL_DFLL48	0
	GCLK_OSCCTRL_PLL	1
FREQM	GCLK_FREQM_MSR0	2
	GCLK_FREQM_MSR1	3
	GCLK_FREQM_REF	4
EIC	GCLK_EIC	5
EVSYS	GCLK_EVSYS_CH0	6
	GCLK_EVSYS_CH1	7
	GCLK_EVSYS_CH2	8
	GCLK_EVSYS_CH3	9
	GCLK_EVSYS_CH4	10
	GCLK_EVSYS_CH5	11
	GCLK_EVSYS_CH6	12
	GCLK_EVSYS_CH7	13
	GCLK_EVSYS_CH8	14
	GCLK_EVSYS_CH9	15
	GCLK_EVSYS_CH10	16
	GCLK_EVSYS_CH11	17
SERCOMm, m = 0...7 SDMMCn, n = 0,1	GCLK_SERCOMm_SLOW, m = 0...7 SDMMCn_SLOW, n=0,1	18
SERCOM0	GCLK_SERCOM0_CORE	19
SERCOM1	GCLK_SERCOM1_CORE	20
SERCOM2	GCLK_SERCOM2_CORE	21
SERCOM3	GCLK_SERCOM3_CORE	22
TCC0,TCC1	GLCK_TCC0, GCLK_TCC1	23
TCC2,TCC3	GLCK_TCC2, GCLK_TCC3	24
SERCOM4	GCLK_SERCOM4_CORE	25
SERCOM5	GCLK_SERCOM5_CORE	26
SERCOM6	GCLK_SERCOM6_CORE	27

Table 20-14. PCHCTRL (Index) GCLK Mapping (continued)

Target Destination	GCLK Name	PCHCTRL (Index)
SERCOM7	GCLK_SERCOM7_CORE	28
TCC4	GCLK_TCC4	29
TCC5	GCLK_TCC5	30
TCC6	GCLK_TCC6	31
TCC7	GCLK_TCC7	32
ADC	GCLK_ADC	33
AC	GCLK_AC	34
PTC	GCLK_PTC	35
SPI_IXS	GCLK_SPI_IXS	36
CCL	GCLK_CCL	37
PDEC	GCLK_PDEC	38
CAN0	GCLK_CAN0	39
CAN1	GCLK_CAN1	40
ETH	GCLK_ETH_TX	41
	GCLK_ETH_TSU	42
SQI	GCLK_SQI	43
SDMMC0	GLCK_SDMMC0	44
SDMMC1	GLCK_SDMMC1	45
USB	GCLK_USB	46
CPU0	GCLK_CPU0_TRACE	47

21. Main Clock (MCLK)

21.1. Overview

The Main Clock (MCLK) controls the synchronous clock generation of the device.

Using a clock provided by the Generic Clock Generator 0 Module (GCLK_MAIN), the Main Clock Controller provides synchronous system clocks to the CPU and the modules connected to the bus APB (Advanced Peripheral Bus), and AHB (Advanced High-performance Bus) fabric.

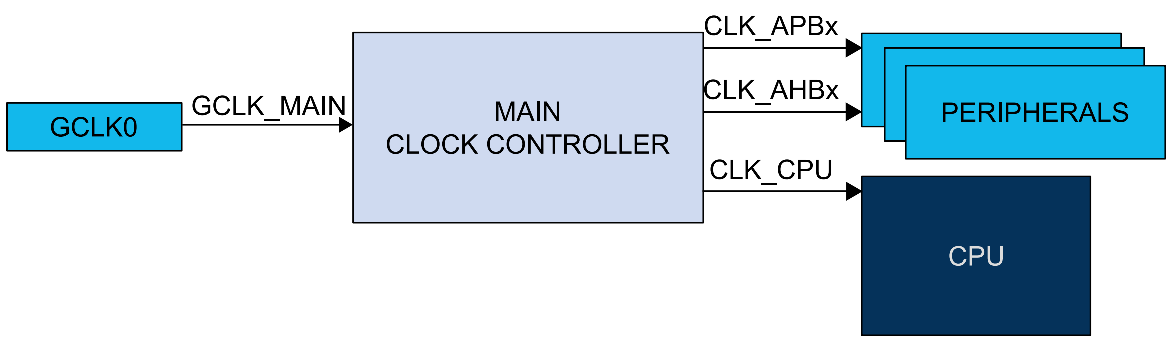
The synchronous system clocks are divided into a number of clock domains. Each clock domain can run at different frequencies, enabling the user to save power by running peripherals at a relatively low clock frequency, while maintaining high CPU performance or vice versa. In addition, the clock can be masked for individual modules, enabling the user to minimize power consumption.

21.2. Features

- Generates CPU, Advanced High-performance Bus (AHB), and Advanced Peripheral Bus (APB) system clocks
  - Clock source and division factor from GCLK0, (GCLK GENERATOR 0)
  - Clock prescaler with 1x to 128x division
- Safe run-time clock switching from GCLK0
- Module-level clock gating through maskable peripheral clocks

21.3. Block Diagram

Figure 21-1. MCLK Block Diagram



21.4. Peripheral Dependencies

Peripheral Name	Base Address	NVIC IRQ Index:Source	PAC Peripheral ID (PAC.WRCTRL.PERIDx)	Power Domain
MCLK	0x4401_2000	9 : CKRDY	9	VDDREG

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

21.4.1. Power Management

The MCLK will operate in all sleep modes if a synchronous clock is required in these modes.

21.4.2. Clocks

The MCLK bus clock (CLK_MCLK_APB) can be enabled and disabled in the Main Clock module, and the default state of CLK_MCLK_APB can be found in the Peripheral Clock Masking section. If this clock is disabled, it can only be re-enabled by a reset.

The Generic Clock GCLK_MAIN is required to generate the Main Clocks. GCLK_MAIN is configured in the Generic Clock Controller, and can be re-configured by the user if needed.

21.4.2.1. Main Clock

The main clock CLK_MAIN is the common source for the synchronous clocks. This is fed into the common 8-bit prescaler that is used to generate synchronous clocks to the CPU, AHBx, and APBx modules.

21.4.2.2. CPU Clock

The CPU clock (CLK_CPU) is routed to the CPU. Halting the CPU clock inhibits the CPU from executing instructions.

21.4.2.3. APBx and AHBx Clock

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, but can be divided by a prescaler, 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.

21.4.2.4. Clock Domains

The device has these synchronous clock domains:

- High-Speed synchronous clock domain (HS Clock Domain). Frequency is f_{HS}.
- CPU synchronous clock domain (CPU Clock Domain). Frequency is f_{CPU}.
- Low Power synchronous clock domain (LP Clock Domain). Frequency is f_{LP}.
- Backup synchronous clock domain. (BUP Clock Domain). Frequency is f_{BUP}.

21.4.3. DMA

Not applicable.

21.4.4. Interrupts

The interrupt request line is connected to the Interrupt Controller. Using the MCLK interrupt requires the Interrupt Controller to be configured first.

21.4.5. 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.

21.4.6. Register Access Protection

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

- Interrupt Flag register (INTFLAG)

Optional write protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write Protection" property in each individual register description.

PAC write protection does not apply to accesses through an external debugger.

21.4.7. PIC32CK SG/GC TrustZone Specific Register Access Protection

On PIC32CK SG/GC devices, this peripheral has different access permissions depending on PAC Security Attribution (Secure or Non-Secure):

- 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

Refer to Implementation Defined Attribution Unit (IDAU) for additional information.

21.5. Functional Description

21.5.1. Principle of Operation

The CLK_MAIN clock signal from the GCLK0 module is the source for the main clock, which in turn is the common root for the synchronous clocks for the CPU, APBx (Advanced Peripheral Bus), and AHBx (Advanced High-performance Bus). The CLK_MAIN is divided by an 8-bit prescaler. Each of the derived clocks can run from any divided or undivided main clock, ensuring synchronous clock sources for each clock domain. The clock domain (CPU, LP, BUP) can be changed on the fly to respond to variable load in the application as long as $f_{CPU} \geq f_{LP} \geq f_{BUP}f_{CPU} \geq f_{BUP}$. The clocks for each module in a clock domain can be masked individually to avoid power consumption in inactive modules. Depending on the sleep mode, some clock domains can be turned off.

21.5.2. Basic Operation

21.5.2.1. Initialization

After a Reset, the default clock source of the CPU (i.e., DFLL48Mhz through GCLK0 (GCLK_MAIN) through CLK_MAIN) is started and calibrated before the CPU starts running. The GCLK_MAIN clock is selected as the main clock without any prescaler division.

21.5.2.2. Enabling, Disabling, and Resetting

The MCLK module is always enabled and cannot be reset.

21.5.2.3. 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.

21.5.2.4. 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 Division (DIV) bits in the CPU Clock Division register CPUDIV, resulting in a CPU clock domain frequency determined by this equation:

$$f_{CPU} = \frac{f_{main}}{CPUDIV}$$

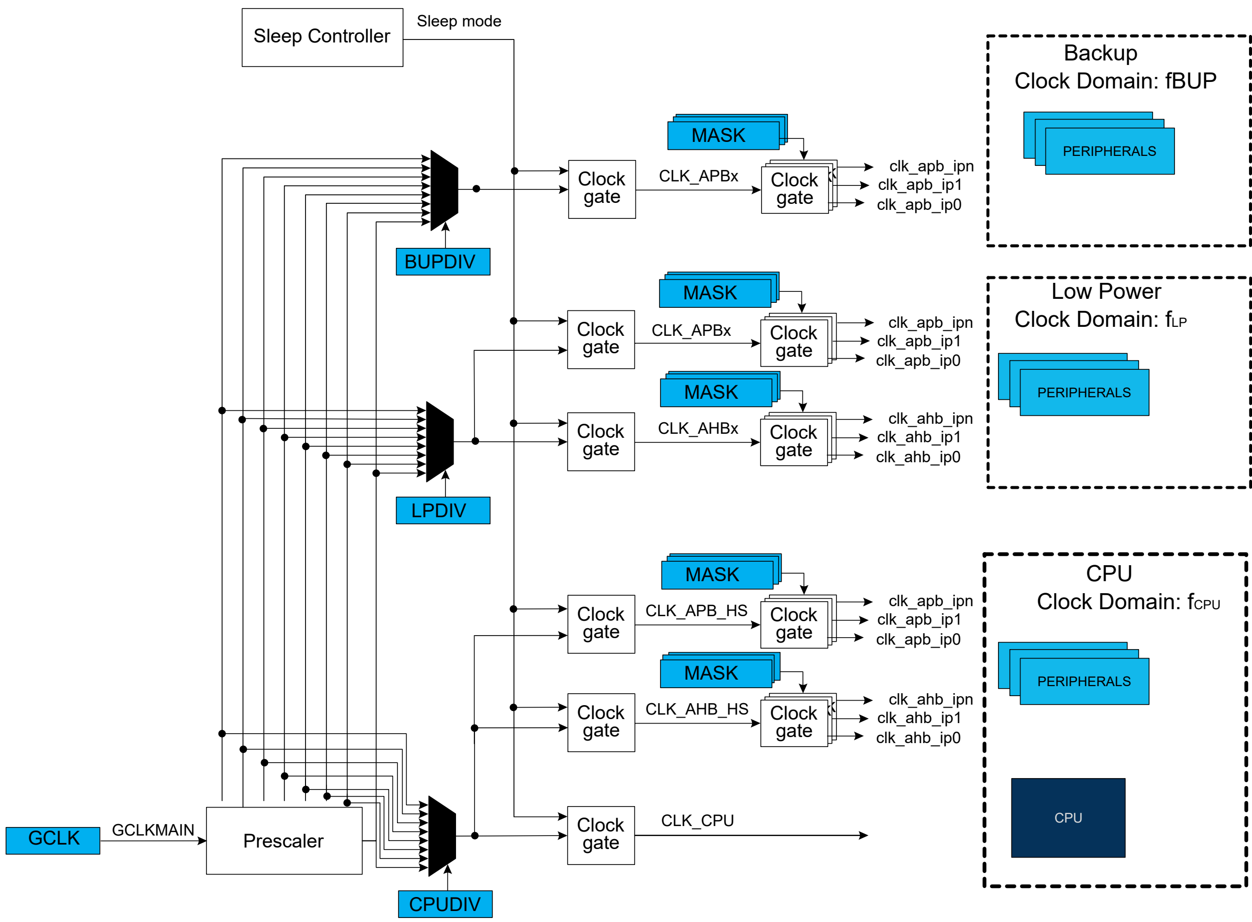
Similarly, the clock for the Low Power and Backup Clock Domain can be divided by writing their respective LPDIV and BUPDIV register. To ensure correct operation, frequencies must be selected so that $f_{CPU} \geq f_{LP} \geq f_{BUP}$. Also, frequencies must never exceed the specified maximum frequency for each clock domain given in the electrical characteristics specifications.

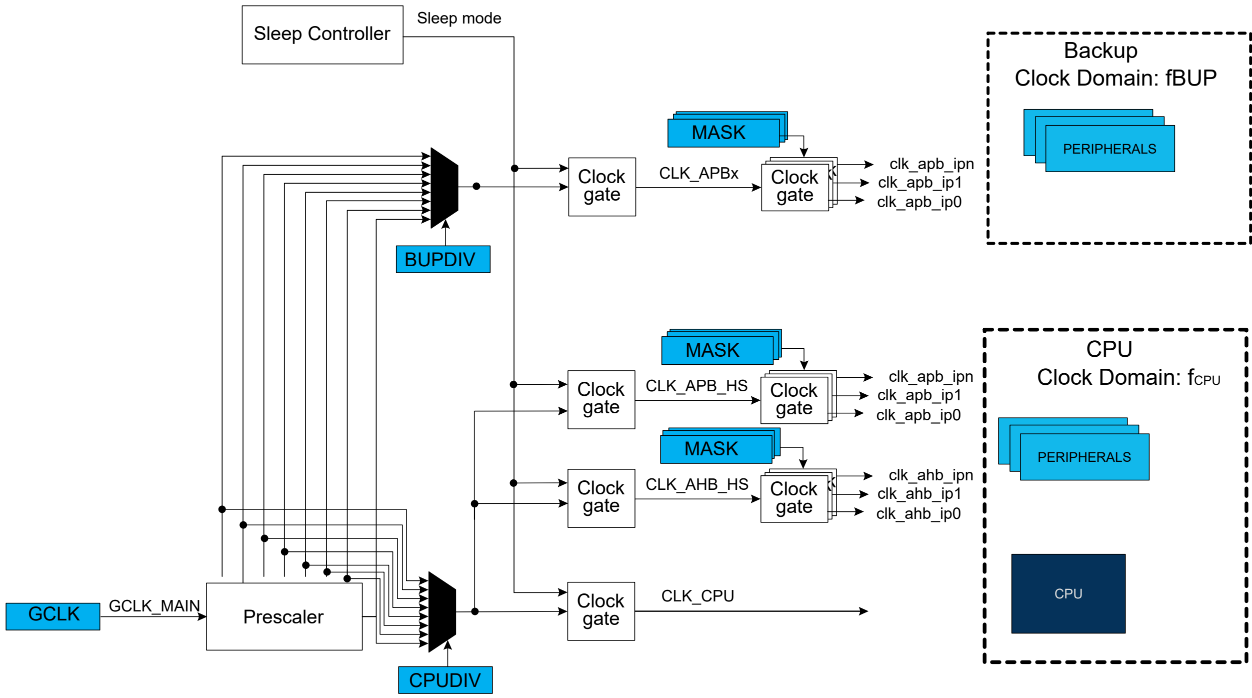
Similarly, the clock for the Backup Clock Domain can be divided by writing the BUPDIV register. To ensure correct operation, frequencies must be selected so that $f_{CPU} \geq f_{BUP}$. Also, frequencies must never exceed the specified maximum frequency for each clock domain given in the electrical characteristics specifications.

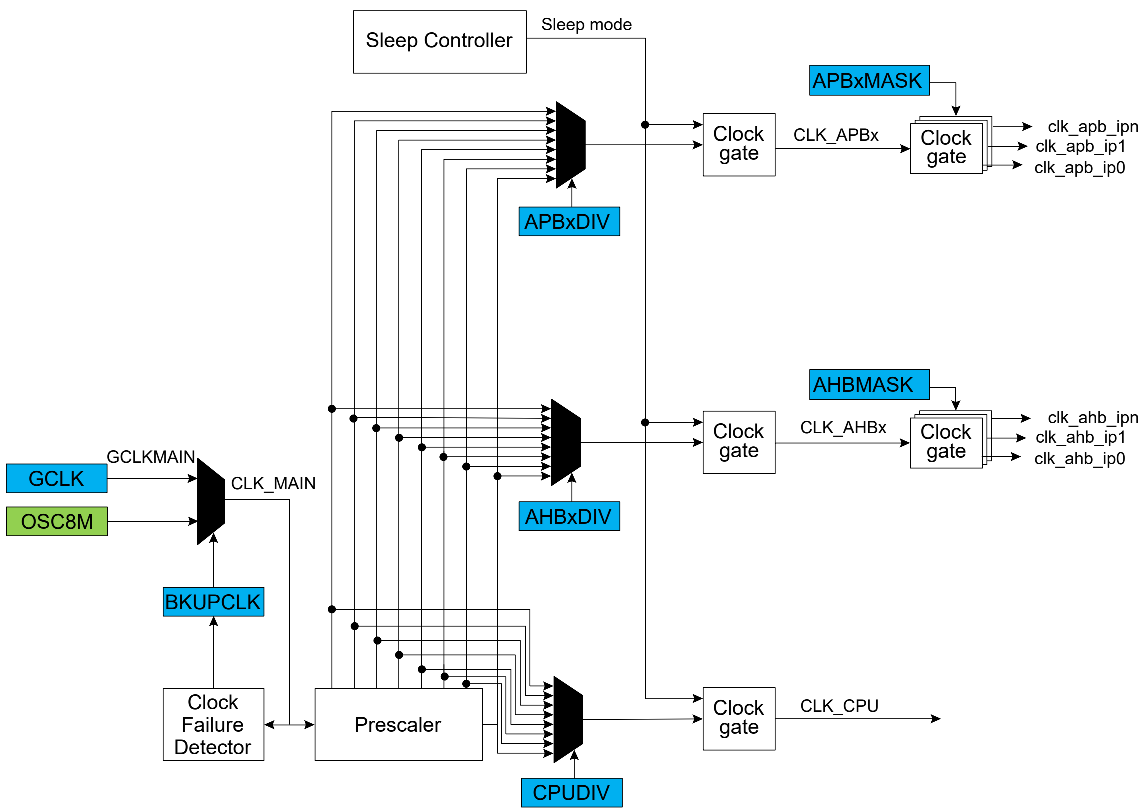
If the application attempts to write forbidden values in CPUDIV, LPDIV, or BUPDIV registers, then these bad values are not used and a violation is reported to the PAC module.

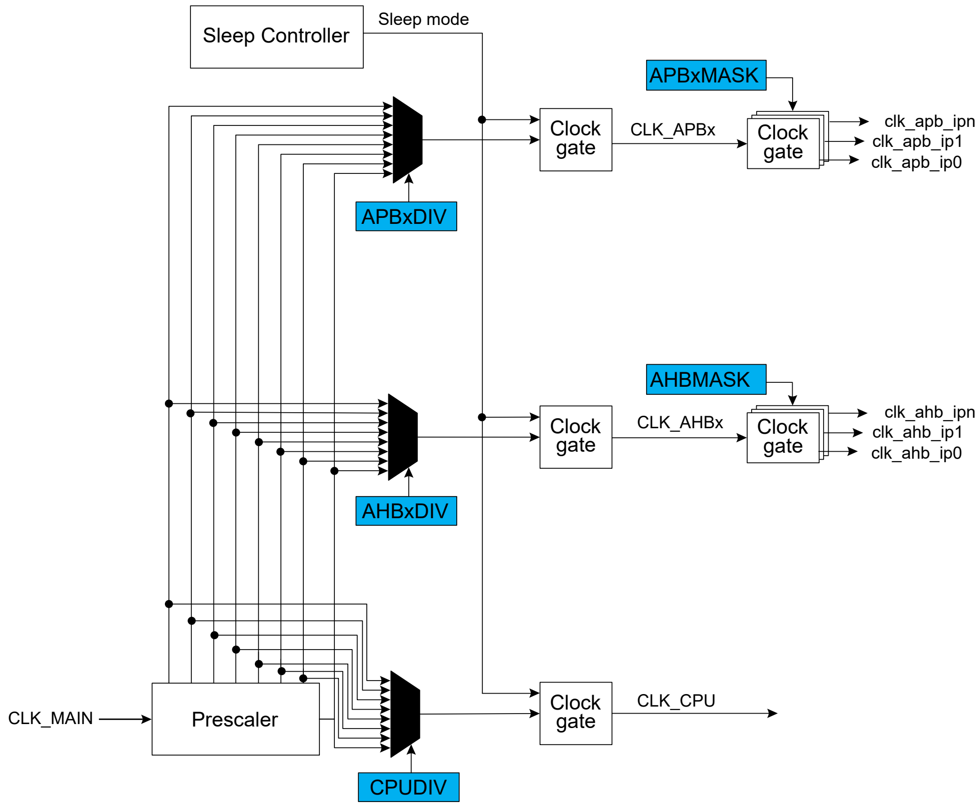
Division bits (DIV) can be written without halting or disabling peripheral modules. Writing DIV bits allows a new clock setting to be written to all synchronous clocks belonging to the corresponding clock domain at the same time. Each clock domain can be changed without changing others. This way, it is possible to, for example, scale the CPU clock domain speed according to the required performance, while keeping the Backup Clock Domain frequency constant.

Figure 21-2. Synchronous Clock Selection and Prescaler




Sleep Controller
Sleep mode
Backup
Clock Domain: fBUP
PERIPHERALS
MASK
Clock gate
CLK_APBx
Clock gate
clk_apb_ipn
clk_apb_ip1
clk_apb_ip0
BUPDIV
MASK
Clock gate
CLK_APB_HS
Clock gate
clk_apb_ipn
clk_apb_ip1
clk_apb_ip0
CPU
Clock Domain: fCPU
PERIPHERALS
MASK
Clock gate
CLK_AHB_HS
Clock gate
clk_ahb_ipn
clk_ahb_ip1
clk_ahb_ip0
CPU
GCLK
GCLK_MAIN
Prescaler
Clock gate
CLK_CPU
CPUDIV


Sleep Controller
Sleep mode
APBxMASK
Clock gate
CLK_APBx
Clock gate
clk_apb_ipn
clk_apb_ip1
clk_apb_ip0
APBxDIV
AHBMASK
Clock gate
CLK_AHBx
Clock gate
clk_ahb_ipn
clk_ahb_ip1
clk_ahb_ip0
GCLKMAIN
GCLK
OSC8M
CLK_MAIN
AHBxDIV
BKUPCLK
Clock Failure Detector
Prescaler
Clock gate
CLK_CPU
CPUDIV



21.5.2.5. Clock Ready Flag

There is a slight delay between writing to CPUDIV, LPDIV, and BUPDIV 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 clock settings (CLKCFG) 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.

21.5.2.6. Peripheral Clock Masking

It is possible to disable/enable the AHB or APB clock for a peripheral by writing the corresponding bit in the Clock Mask registers MSKx to '0'/'1'. The default state of the peripheral clocks is shown here.

Table 21-1. Peripheral Clock Masking

Clock	Clock Index	SFR Field	POR Default
CLK_AC_APB	105	CLKMSK3.MSK9	Enabled
CLK_ADC_APB	104	CLKMSK3.MSK8	Enabled
CLK_BSDAP_APB	115	CLKMSK3.MSK19	Enabled
CLK_CAN0_AHB	13	CLKMSK0.MSK13	Enabled
CLK_CAN1_AHB	14	CLKMSK0.MSK14	Enabled
CLK_CCL_APB	109	CLKMSK3.MSK13	Enabled
CLK_DMA0_AHB	8	CLKMSK0.MSK8	Enabled
CLK_DMA0_APB	65	CLKMSK2.MSK1	Enabled
CLK_DMA1_AHB	9	CLKMSK0.MSK9	Enabled
CLK_DMA1_APB	66	CLKMSK2.MSK2	Enabled

Table 21-1. Peripheral Clock Masking (continued)

Clock	Clock Index	SFR Field	POR Default
CLK_DSU_AHB	3	CLKMSK0.MSK3	Enabled
CLK_DSU_APB	32	CLKMSK1.MSK0	Enabled
CLK_EBI_AHB	21	CLKMSK0.MSK21	Enabled
CLK_EBI_APB	114	CLKMSK3.MSK18	Enabled
CLK_EIC_APB	45	CLKMSK1.MSK13	Enabled
CLK_ETH_AHB	15	CLKMSK0.MSK15	Enabled
CLK_ETH_APB	111	CLKMSK3.MSK15	Enabled
CLK_EVSYS_APB	70	CLKMSK2.MSK6	Enabled
CLK_FCR_AHB	4	CLKMSK0.MSK4	Enabled
CLK_FCR_APB	33	CLKMSK1.MSK1	Enabled
CLK_FCW_AHB	5	CLKMSK0.MSK5	Enabled
CLK_FCW_APB	34	CLKMSK1.MSK2	Enabled
CLK_FREQM_APB	42	CLKMSK1.MSK10	Enabled
CLK_HSM_AHB	22	CLKMSK0.MSK22	Enabled
CLK_IDAU_APB	69	CLKMSK2.MSK5	Enabled
CLK_MBISTINTF_APB	50	CLKMSK1.MSK18	Enabled
CLK_OSC32KCTRL_APB	39	CLKMSK1.MSK7	Enabled
CLK_OSCCTRL_APB	38	CLKMSK1.MSK6	Enabled
CLK_PAC_AHB	6	CLKMSK0.MSK6	Enabled
CLK_PAC_APB	46	CLKMSK1.MSK14	Enabled
CLK_PCC_APB	108	CLKMSK3.MSK12	Enabled
CLK_PDEC_APB	110	CLKMSK3.MSK14	Enabled
CLK_PORT_APB	64	CLKMSK2.MSK0	Enabled
CLK_PRM_AHB	12	CLKMSK0.MSK12	Enabled
CLK_PRM_APB	68	CLKMSK2.MSK4	Enabled
CLK_PTC_APB	106	CLKMSK3.MSK10	Enabled
CLK_RSTC_APB	37	CLKMSK1.MSK5	Enabled
CLK_RTC_APB	44	CLKMSK1.MSK12	Enabled
CLK_SDMMC0_AHB	17	CLKMSK0.MSK17	Enabled
CLK_SDMMC1_AHB	18	CLKMSK0.MSK18	Enabled
CLK_SERCOM0_APB	71	CLKMSK2.MSK7	Enabled
CLK_SERCOM1_APB	72	CLKMSK2.MSK8	Enabled
CLK_SERCOM2_APB	73	CLKMSK2.MSK9	Enabled
CLK_SERCOM3_APB	74	CLKMSK2.MSK10	Enabled
CLK_SERCOM4_APB	96	CLKMSK3.MSK0	Enabled
CLK_SERCOM5_APB	97	CLKMSK3.MSK1	Enabled
CLK_SERCOM6_APB	98	CLKMSK3.MSK2	Enabled
CLK_SERCOM7_APB	99	CLKMSK3.MSK3	Enabled
CLK_SPI_IXS_APB	107	CLKMSK3.MSK11	Enabled
CLK_SQI_AHB	16	CLKMSK0.MSK16	Enabled
CLK_TCC0_APB	75	CLKMSK2.MSK11	Enabled
CLK_TCC1_APB	76	CLKMSK2.MSK12	Enabled
CLK_TCC2_APB	77	CLKMSK2.MSK13	Enabled
CLK_TCC3_APB	78	CLKMSK2.MSK14	Enabled
CLK_TCC4_APB	100	CLKMSK3.MSK4	Enabled

Table 21-1. Peripheral Clock Masking (continued)

Clock	Clock Index	SFR Field	POR Default
CLK_TCC5_APB	101	CLKMSK3.MSK5	Enabled
CLK_TCC6_APB	102	CLKMSK3.MSK6	Enabled
CLK_TCC7_APB	103	CLKMSK3.MSK7	Enabled
CLK_TDM_APB	51	CLKMSK1.MSK19	Enabled
CLK_TRAM_APB	47	CLKMSK1.MSK15	Enabled
CLK_TRNG_APB	112	CLKMSK3.MSK16	Enabled
CLK_USB_AHB	19	CLKMSK0.MSK19	Enabled
CLK_USB_APB	113	CLKMSK3.MSK17	Enabled
CLK_USBHS_AHB	20	CLKMSK0.MSK20	Enabled
CLK_WDT_APB	43	CLKMSK1.MSK11	Enabled

When the APB clock is not provided to a module, its registers cannot be read or written. The module can be re-enabled later by writing the corresponding mask bit to '1'.

A module may be connected to several clock domains (for instance, AHB and APB), in which case it will have several mask bits.

Note that clocks should only be switched off if it is certain that the module will not be used: Switching off the clock for the NVM Controller (NVMCTRL) will cause a problem if the CPU needs to read from the Flash Memory. Switching off the clock to the MCLK module (which contains the mask registers) or the corresponding APBx bridge, will make it impossible to write the mask registers again. In this case, they can only be re-enabled by a system reset.

21.5.3. DMA Operation

Not applicable.

21.5.4. Interrupts

The peripheral has the following interrupt sources:

- Clock Ready (CKRDY): indicates that CPU, LP, and BUP 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.

21.5.5. Sleep Mode Operation

In all IDLE sleep modes, the MCLK is still running on the selected main clock.

In STANDBY sleep mode, the MCLK is frozen if no synchronous clock is required.

21.6. Register Summary

For descriptions and definitions of both Register and bitfield properties, refer to Register Properties.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	INTENCLR	31:24								
		23:16								
		15:8								
		7:0								CKRDY
0x04	INTENSET	31:24								
		23:16								
		15:8								
		7:0								CKRDY
0x08	INTFLAG	31:24								
		23:16								
		15:8								
		7:0								CKRDY
0x0C	CLKDIV0	31:24								
		23:16								
		15:8								
		7:0	DIV[7:0]							
0x10 ... 0x13	Reserved									
0x14	CLKDIV1	31:24								
		23:16								
		15:8								
		7:0	DIV[7:0]							
0x18 ... 0x3B	Reserved									
0x3C	CLKMSK0	31:24								
		23:16		MSK22	MSK21	MSK20	MSK19	MSK18	MSK17	MSK16
		15:8	MSK15	MSK14	MSK13	MSK12			MSK9	MSK8
		7:0					MSK6	MSK5	MSK4	MSK3
0x40	CLKMSK1	31:24								
		23:16					MSK19	MSK18		
		15:8	MSK15	MSK14	MSK13	MSK12	MSK11	MSK10		
		7:0	MSK7	MSK6	MSK5			MSK2	MSK1	MSK0
0x44	CLKMSK2	31:24								
		23:16								
		15:8		MSK14	MSK13	MSK12	MSK11	MSK10	MSK9	MSK8
		7:0	MSK7	MSK6	MSK5	MSK4		MSK2	MSK1	MSK0
0x48	CLKMSK3	31:24								
		23:16					MSK19	MSK18	MSK17	MSK16
		15:8	MSK15	MSK14	MSK13	MSK12	MSK11	MSK10	MSK9	MSK8
		7:0	MSK7	MSK6	MSK5	MSK4	MSK3	MSK2	MSK1	MSK0
0x4C ... 0x5F	Reserved									
0x60	ODOFF	31:24								
		23:16								
		15:8								
		7:0								ODMSK

21.6.1. Interrupt Enable Clear Register

Name: INTENCLR
Offset: 0x00
Reset: 0x00000000
Property: PAC Write-Protection

Table 21-2. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								CKRDY
Access								R/K
Reset								0

Bit 0 – CKRDY Clock Ready Interrupt Enable Clear

Note: 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.

21.6.2. Interrupt Enable Set

Name: INTENSET
Offset: 0x04
Reset: 0x00000000
Property: PAC Write-Protection

Table 21-3. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								CKRDY
Access								R/S
Reset								0

Bit 0 – CKRDY Clock Ready Interrupt Enable Set

Value	Description
0	The Clock Ready interrupt is disabled.
1	The Clock Ready interrupt is enabled.

21.6.3. Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x08
Reset: 0x00000000
Property: PAC Write-Protection

Note: Subsequent to an interrupt flag being cleared, the flag must be read back to verify the clear before exiting the ISR. Failure to do this can result in duplicate interrupts.

Table 21-4. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								CKRDY
Access								HS/K
Reset								0

Bit 0 – CKRDY Clock Ready Interrupt Flag
Notes:
1. Writing a '1' to this bit will clear the Clock Ready Interrupt Enable bit and the corresponding interrupt request.
2. This flag is **set by hardware** when the system clocks have frequencies as indicated in the CLKDIVx registers and will generate an interrupt if CKRDY interrupt enable is set to '1'.

Value	Description
0	The Clock Ready interrupt is disabled.
1	The Clock Ready interrupt is enabled.

21.6.4. Clock Divide n Register

Name: CLKDIVn
Offset: 0x0C + n*0x08 [n=0..1]
Reset: 0x00000001 (0x00000000 for CLKDIV0)
Property: PAC Write-Protection

Important: To facilitate the use of sleep modes, the following conditions must be met:

1. PLL0 must be dedicated to the CPU.
2. PLL0 must be stepped down in <= 75 MHz increments to <= 75 MHz output when entering sleep modes.
3. PLL0 must be stepped up to the operating frequency in <= 75 MHz increments after exiting sleep modes.
4. The step delay for both of these processes needs to be >= 1 us.

Notes:

1. The CLKDIV0.DIV bit field is write protected.
2. To ensure correct operation, frequencies must be selected so that CLKDIV0.DIV $\leq$ CLKDIV1.DIV.
3. Frequencies must never exceed the specified maximum frequency for each clock domain.
4. The user updates to this register may not take effect immediately. The MCLK module logic will wait for the falling edge of the previous clock and the new clock to coincide before switching. The INTFLAG.CKRDY can be used to determine when MCLK has made the switch.

Table 21-5. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	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	1

Bits 7:0 – DIV[7:0] CPU Clock Domain Division Factor

These bits define the division ratio of the main clock (MCLK) prescaler related to the CPU Clock Domain controlled by the CLKDIVn register.

Note: All other values are reserved or invalid.

Value	Description
`0x01`	Divide by 1
`0x02`	Divide by 2
`0x04`	Divide by 4
`0x08`	Divide by 8
`0x10`	Divide by 16
`0x20`	Divide by 32
`0x40`	Divide by 64
`0x80`	Divide by 128

21.6.5. Peripheral BUS Clock Enable Mask0 Register

Name: CLKMSK0
Offset: 0x3C
Reset: 0x007F_FFFF
Property: PAC Write-Protection

Note: AHB = Advanced High-performance Bus
APB = Advanced Peripheral Bus

Table 21-6. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
		MSK22	MSK21	MSK20	MSK19	MSK18	MSK17	MSK16
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	1	0	0

Bit	15	14	13	12	11	10	9	8
	MSK15	MSK14	MSK13	MSK12			MSK9	MSK8
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
					MSK6	MSK5	MSK4	MSK3
Access					R/W	R/W	R/W	R/W
Reset					1	1	1	1

Bits 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22 – MSKn Clock Enable Mask

Bits 8, 9 – MSKn Clock Enable Mask

Bits 0, 1, 2, 3 – MSKn Clock Enable Mask

Bit Number	Module
3	DSU_AHB
4	FCR
5	FCW
6	PAC
8	DMA0
9	DMA1
12	PRM
13	CAN0
14	CAN1
15	ETH
16	SQI

MSKn (continued)

Bit Number	Module
17	SDMMC0
18	SDMMC1
19	USBFS
20	USBHS
21	EBI
22	HSM

21.6.6. Peripheral BUS Clock Enable Mask1 Register

Name: CLKMSK1
Offset: 0x40
Reset: 0x0006_FFFF
Property: PAC Write-Protection

Note: AHB = Advanced High-performance Bus
APB = Advanced Peripheral Bus

Table 21-7. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					MSK19	MSK18		
Access					R/W	R/W		
Reset					1	0		

Bit	15	14	13	12	11	10	9	8
	MSK15	MSK14	MSK13	MSK12	MSK11	MSK10		
Access	R/W	R/W	R/W	R/W	R/W	R/W		
Reset	0	0	1	1	1	1		

Bit	7	6	5	4	3	2	1	0
	MSK7	MSK6	MSK5			MSK2	MSK1	MSK0
Access	R/W	R/W	R/W			R/W	R/W	R/W
Reset	1	1	1			1	1	1

Bits 18, 19 – MSKn Clock Enable Mask n

Bits 10, 11, 12, 13, 14, 15 – MSKn Clock Enable Mask n

Bits 5, 6, 7 – MSKn Clock Enable Mask n

Bits 0, 1, 2 – MSKn Clock Enable Mask n

Bit Number	Module
0	DSU_APB
1	FCR
2	FCW
5	RSTC
6	OSCCTRL
7	OSC32KCTRL
10	FREQM
11	WDT
12	RTC

MSKn (continued)

Bit Number	Module
13	EIC
14	PAC
15	TRAM
18	MBISTINTF
19	TDM

21.6.7. Peripheral BUS Clock Enable Mask2 Register

Name: CLKMSK2
Offset: 0x44
Reset: 0x0000_7FFF
Property: PAC Write-Protection

Note: AHB = Advanced High-performance Bus
APB = Advanced Peripheral Bus

Table 21-8. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
		MSK14	MSK13	MSK12	MSK11	MSK10	MSK9	MSK8
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
	MSK7	MSK6	MSK5	MSK4		MSK2	MSK1	MSK0
Access	R/W	R/W	R/W	R/W		R/W	R/W	R/W
Reset	1	1	1	1		1	1	1

Bits 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14 – MSKn Clock Enable Mask n

Bits 0, 1, 2 – MSKn Clock Enable Mask n

Bit Number	Module
0	PORT
1	DMA0
2	DMA1
4	PRM
5	IDAU
6	EVSYS
7	SERCOM0
8	SERCOM1
9	SERCOM2
10	SERCOM3
11	TCC0
12	TCC1
13	TCC2

MSKn (continued)

Bit Number	Module
14	TCC3

21.6.8. Peripheral BUS Clock Enable Mask3 Register

Name: CLKMSK3
Offset: 0x48
Reset: 0x0007_FFFF
Property: PAC Write-Protection

Note: AHB = Advanced High-performance Bus
APB = Advanced Peripheral Bus

Table 21-9. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					MSK19	MSK18	MSK17	MSK16
Access					R/W	R/W	R/W	R/W
Reset					1	1	1	1

Bit	15	14	13	12	11	10	9	8
	MSK15	MSK14	MSK13	MSK12	MSK11	MSK10	MSK9	MSK8
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
	MSK7	MSK6	MSK5	MSK4	MSK3	MSK2	MSK1	MSK0
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 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19 – MSKn Clock Enable Mask n

Bit Number	Module
0	SERCOM4
1	SERCOM5
2	SERCOM6
3	SERCOM7
4	TCC4
5	TCC5
6	TCC6
7	TCC7
8	ADC
9	AC
10	PTC
11	I2S
12	PCC
13	CCL
14	PDEC

MSKn (continued)

Bit Number	Module
15	ETH
16	TRNG
17	USB
18	EBI
19	BSDAP

21.6.9. On Demand Clock OFF Register

Name: ODOFF
Offset: 0x60
Reset: 0x00000000
Property: PAC Write-Protection

Table 21-10. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

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

Bit 0 – ODMSK On Demand Clock Control

Value	Description
0	On demand clock feature is available for peripheral.
1	On demand clock feature is disabled for peripheral.

22. 32 KHz Oscillators Controller (OSC32KCTRL)

22.1. Overview

The 32 KHz Oscillators Controller (OSC32KCTRL) provides a user interface to the 32.768kHz oscillators:

- A 32.768 kHz crystal oscillator (XOSC32K)
- A 32.768 kHz ultra low-power internal RC oscillator (OSCULP32K)

The OSC32KCTRL sub-peripherals can be enabled, disabled, calibrated, and monitored through interface registers. All sub-peripheral statuses are collected in the Status register (STATUS). They can additionally trigger interrupts upon status changes through the INTENSET, INTENCLR, and INTFLAG registers. The OSC32KCTRL provides clock sources to the Generic Clock Controller (GCLK), Real-Time Counter (RTC), Watchdog Timer (WDT), Hardware Security Module (HSM), and Reset Power Management Unit (RPMU).

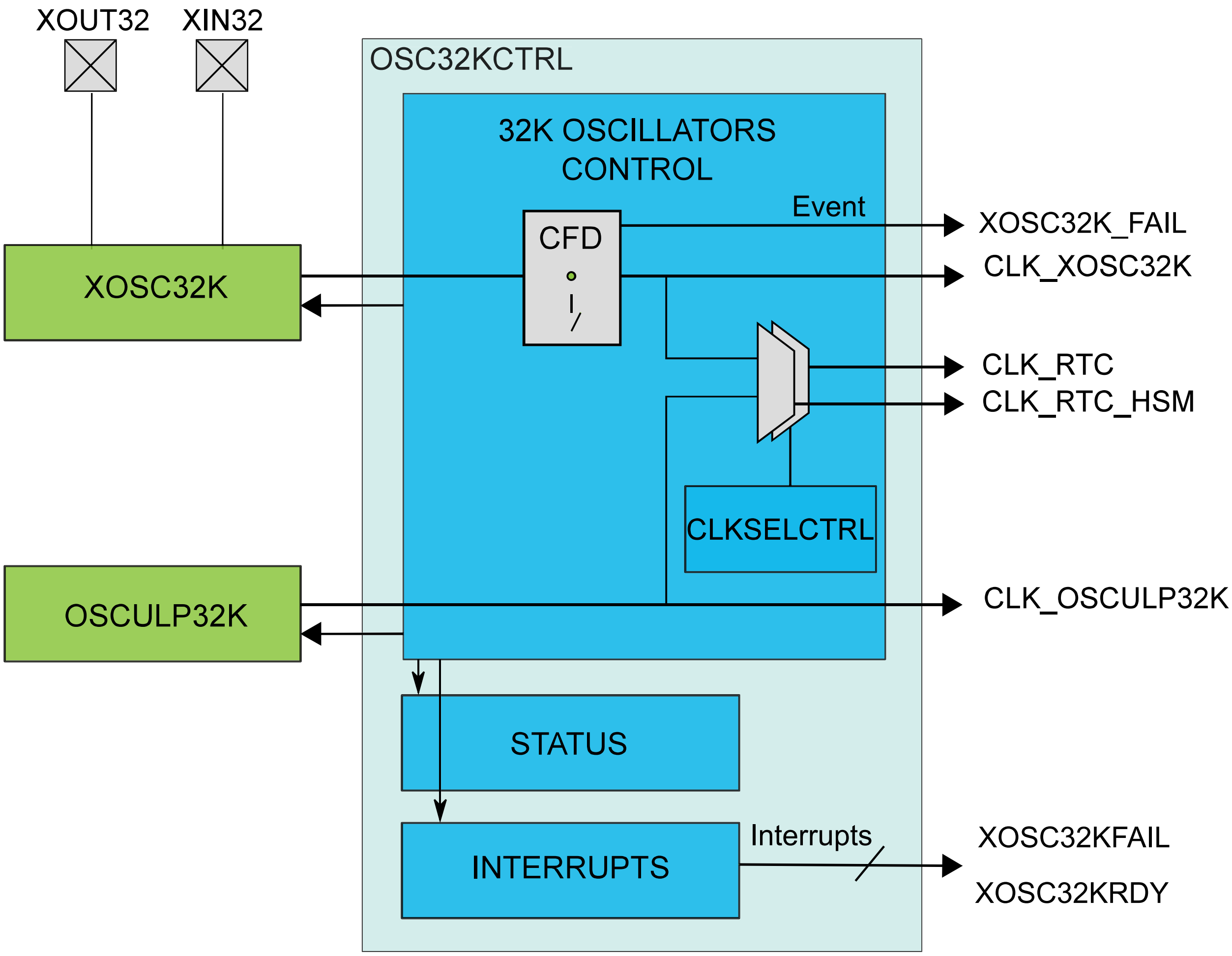
22.2. Features

The following are key features of the OSC32KCTRL module:

- 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
- 1.024 kHz clock outputs available

22.3. Block Diagram

Figure 22-1. OSC32KCTRL Block Diagram



22.4. 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 and do not need to be configured by the user. The external 32.768 kHz crystal must be connected between the XIN32 and XOUT32 pins, along with any required load capacitors.

22.5. Peripheral Dependencies

Table 22-1. OSC32KCTRL Configuration Summary

Peripheral name	Base address	NVIC IRQ Index	MCLK AHBx/APBx Clock Enable Mask Bit	PAC Peripheral ID (PAC.WRCTRL.PERIDx)	EVSYS Generators (EVSYS.CHANNELn.EVGENx)	Power Domain
OSC32KCTRL	0x4400_E000	8: XOSC32KRDY XOSC32KFAIL	MCLK.CLKMSK1[7]	7: INTFLAG[7]	3: XOSC32K_FAIL	AVDD

22.6. Functional Description

22.6.1. Principle of Operation

The XOSC32K and OSCULP32K are configured through the OSC32KCTRL control registers. Through this interface, the sub-peripherals are enabled, disabled, or have their calibration values updated.

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.

22.6.2. 32 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 (XOSC32K.XTALEN = 1), the XIN32 and XOUT32 pins are controlled by XOSC32K.ENABLE, and the GPIO functions are overridden on both pins. When in external clock mode (XOSC32K.XTALEN = 0), only the XIN32 pin will be overridden and controlled by XOSC32K.ENABLE, while the XOUT32 pin can still be used as a GPIO pin.

Enabling, Disabling: The XOSC32K is enabled by writing a '1' to the Enable bit in the 32 kHz External Crystal Oscillator Control register (XOSC32K.ENABLE = 1). It is necessary to wait for STATUS.XOSC32KRDY bit to set, before disabling the XOSC32K. The XOSC32K is disabled by writing a '0' to the Enable bit in the 32 kHz External Crystal Oscillator Control register (XOSC32K.ENABLE = 0). It is necessary to wait for STATUS.XOSC32KRDY to clear before enabling the XOSC32K.

Mode Selection: To enable the XOSC32K in Crystal Oscillator mode, the XTALEN bit in the 32 kHz External Crystal Oscillator Control register must be written (XOSC32K.XTALEN = 1). If XOSC32K.XTALEN is '0', the External Clock Input mode will be enabled. This bit must be written when XOSC32K.ENABLE = 0. Otherwise, the write on this bit is ignored.

Gain Selection: When a crystal oscillator is selected, a user controllable gain is provided through the CGM control bits. The user must review the electrical specification. This bit must be written when XOSC32K.ENABLE = 0. Otherwise, the write on this bit is ignored.

Startup: XOSC32K.STARTUP[3:0] selects the startup time. It is configurable and applicable for crystal mode or external clock mode. These bits must be written when XOSC32K.ENABLE = 0. Otherwise, the write on these bits is ignored.

The XOSC32K will behave differently in different sleep modes based on the settings of XOSC32K.ONDEMAND and XOSC32K.ENABLE bits. The XOSC32K.ONDEMAND bit must be written when XOSC32K.ENABLE = 0. Otherwise, the write on this bit is ignored. If XOSC32K.ENABLE = 0, the XOSC32K will be always stopped. For XOS32K.ENABLE = 1, following table is valid:

XOSC32K.ONDEMAND	Sleep Behavior
0	Always run
1	Run if requested by a peripheral

As a crystal oscillator usually requires a long start-up time, the 32 kHz External Crystal Oscillator will keep running across resets, 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 start-up time can be configured by changing the Oscillator Start-Up Time bit group (XOSC32K.STARTUP[3:0]) in the 32 kHz External Crystal Oscillator Control register. During the start-up 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).

22.6.3. 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 pre-scaler. 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.

The Oscillator’s registers allow to enable, disable, and configure the CFD. The Status register provides status flags on failure and clock switch conditions. The CFD can be configured to trigger an interrupt or an event when a failure is detected.

Clock Failure Detection

The CFD is reset only at Power-on Reset (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). After starting or restarting the XOSC32K, the CFD does not detect failure until the start-up time has elapsed. The start-up time is configured by the Oscillator Start-Up Time in the External Multipurpose Crystal Oscillator Control register (XOSC32K.STARTUP). Once the XOSC32K Start-Up Time is elapsed, the XOSC32K clock is constantly monitored.

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.XOSC32KFAIL) and the Clock Failure Detector interrupt flag bit in the Interrupt Flag register (INTFLAG.XOSC32KFAIL) are set. If the XOSC32KFAIL bit in the Interrupt Enable Set register (INTENSET.XOSC32KFAIL) is set, an interrupt is generated. If the Event Output enable bit in the Event Control register (EVCTRL.CFDEO) is set, an output event is generated.

After a clock failure is issued, the CFD continuously monitors of the XOSC32K clock and the Clock Failure Detector status bit in the Status register (STATUS.XOSC32KFAIL) reflects the current XOSC32K activity.

Clock Switch

When a clock failure is detected, the XOSC32K clock 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. Both 32KHz and 1KHz outputs of the XOSC32K are replaced by the OSCULP32K, 32KHz and 1KHz outputs, respectively. The safe clock source can be scaled down by a configurable pre-scaler 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.XOSC32KSW) is set.

When the CFD switches 32KHz clock from XOSC32K to the safe clock, the XOSC32K is still enabled. If required, 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 Switch Back Enable bit in the Clock Failure Control register (CFDCTRL.SWBACK). Once the XOSC32K clock is switched back, the Switch Back bit (CFDCTRL.SWBACK) is cleared by hardware.

Note: If an application can recover the XOSC32K from a Clock Failure Detection, the application can switch back to the XOSC32K clock (CFDCTRL.SWBACK = 1). Entering in Low-Power Sleep mode will not be possible if the clock switch back can not complete due to a persistent failure. Therefore, the user should disable the Clock Failure Detector (CFDCTRL.CFDEN = 0) until the clock switch back has completed (CFDCTRL.SWBACK = 0).

Prescaler

The CFD has an internal configurable prescaler to generate the safe clock from the OSCULP32K oscillator. The prescaler allows users to scale down the OSCULP32K oscillator clock to half so the safe clock frequency is not higher than the XOSC32K clock frequency monitored by the CFD.

The prescaler is applied on both outputs (32 kHz and 1 kHz) of the safe clock.

Note: For an external crystal oscillator at 32 kHz and the OSCULP32K frequency is 32 kHz, the XOSC32K.CFDPRESC must be set to 0 for a safe clock of equal frequency.

Sleep Mode

The CFD is halted depending on configuration of the XOSC32K and the peripheral clock request. In sleep modes, CFD operation is automatically disabled when the external oscillator is not requested to run by a peripheral. For further details, refer to the XOSC32K Sleep mode Behavior table. The CFD interrupt can be used to wake up the device from sleep modes.

22.6.4. 32 kHz Ultra Low-Power Internal Oscillator (OSCULP32K) Operation

The OSCULP32K provides a tunable, low-speed, and ultra low-power clock source. The OSCULP32K is factory-calibrated oscillator 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.

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.

22.6.5. Watchdog Timer Clock Selection

The Watchdog Timer (WDT) uses the internal 1.024 kHz OSCULP32K output clock. This clock is running all the time and internally enabled when requested by the WDT module.

22.6.6. RTC Clock Selection Control

Before enabling the system RTC module, the RTC clock must be selected first. All outputs from this oscillator source are valid RTC clock sources. The RTC Clock Selection is available in the Selection Control register (CLKSELCTRL.RTCSEL). To ensure proper operation, it is highly recommended to disable the RTC module before changing its clock source.

22.6.7. Interrupts

The OSC32KCTRL has the following interrupt sources:

- **XOSC32KRDY - 32 kHz Crystal Oscillator Ready:** A 0-to-1 transition on the STATUS.XOSC32KRDY bit is detected and XOSC32K ready interrupt request or XOSC32K ready interrupt wake up request are set.
- **XOSC32KFAIL - Clock Failure Detector:** A 0-to-1 transition on the STATUS.XOSC32KFAIL bit is detected and XOSC32K clock fail detector interrupt request or XOSC32K clock fail detector interrupt wake up request are set.

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 two interrupt request line each for XOSC32KRDY and XOSC32KFAIL interrupt sources.

Note: Interrupts must be globally enabled for interrupt requests to be generated.

22.6.8. Events

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 and event line will go high. When the CFD is switched to the safe clock, the CFD clock failure will not be output on the Event Output.

The CFD can generate the following output event:

- Clock Failure Detector (XOSC32KFAIL): Generated when the Clock Failure Detector status bit is set in the Status register (STATUS.XOSC32KFAIL). The CFD event is not generated when the Clock Switch bit (STATUS.XOSC32KSW) 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.

22.7. Register Summary

For descriptions and definitions of both Register and bitfield properties, refer to Register Properties.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	INTENCLR	31:24								
		23:16								
		15:8								
		7:0						XOSC32KFAIL		XOSC32KRDY
0x04	INTENSET	31:24								
		23:16								
		15:8								
		7:0						XOSC32KFAIL		XOSC32KRDY
0x08	INTFLAG	31:24								
		23:16								
		15:8								
		7:0						XOSC32KFAIL		XOSC32KRDY
0x0C	STATUS	31:24								
		23:16								
		15:8								
		7:0					XOSC32KSW	XOSC32KFAIL		XOSC32KRDY
0x10	CLKSELCTRL	31:24								
		23:16								
		15:8								
		7:0			HSMSEL[1:0]				RTCSEL[1:0]	
0x14	CFDCTRL	31:24								
		23:16								
		15:8								
		7:0						CFDPRESC	SWBACK	CFDEN
0x18	EVCTRL	31:24								
		23:16								
		15:8								
		7:0								CFDEO
0x1C	XOSC32K	31:24								
		23:16			CGM[3:0]					
		15:8					STARTUP[3:0]			
		7:0	ONDEMAND					XTALEN	ENABLE	

22.7.1. 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).

Table 22-2. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						XOSC32KFAIL		XOSC32KRDY
Access						R/W/HC		R/W/HC
Reset						0		0

Bit 2 – XOSC32KFAIL XOSC32K 32.768kHz Clock Failure Detect Interrupt Enable
Note: Writing a '0' to this bit has no effect.

This bit is cleared under the following conditions:
- Writing a '1' to this bit will clear the XOSC32K Clock Fail Interrupt Enable bit, (i.e. XOSC32KFAIL), which disables the XOSC32K Clock Failure interrupt
- Writing a one to the same corresponding bit in the INTENSET register

Value	Description
0	The XOSC32K Clock Fail Detect Interrupt is disabled.
1	The XOSC32K Clock Fail Detect Interrupt is enabled. An interrupt request will be generated when the XOSC32K Clock Failure Detection interrupt flag is set.

Bit 0 – XOSC32KRDY XOSC32K 32.768kHz Ready Interrupt Enable
Note: Writing a '0' to this bit has no effect.

This bit is cleared under the following conditions:
- Writing a '1' to this bit, (XOSC32KRDY), will clear the XOSC32K Ready Interrupt Enable bit, which disables the XOSC32K Ready interrupt

- Writing a one to the same corresponding bit in the INTENSET register

Value	Description
0	The XOSC32K Ready interrupt is disabled.
1	The XOSC32K Ready interrupt is enabled.

22.7.2. 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).

Table 22-3. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						XOSC32KFAIL		XOSC32KRDY
Access						R/W/HC		R/W/HC
Reset						0		0

Bit 2 – XOSC32KFAIL XOSC32K 32.768kHz Clock Failure Detect Interrupt Enable
Note: Writing a '0' to this bit has no effect.

This bit is cleared under the following conditions:

- Writing a '1' to this bit will clear the XOSC32K Clock Fail Interrupt Enable bit, (i.e. XOSC32KFAIL), which disables the XOSC32K Clock Failure interrupt
- Writing a one to the same corresponding bit in the INTENCLR register

Value	Description
0	The XOSC32K Clock Fail Detect Interrupt is disabled.
1	The XOSC32K Clock Fail Detect Interrupt is enabled. An interrupt request will be generated when the XOSC32K Clock Failure Detection interrupt flag is set.

Bit 0 – XOSC32KRDY XOSC32K 32.768kHz Ready Interrupt Enable
Note: Writing a '0' to this bit has no effect.

This bit is cleared under the following conditions:

- Writing a '1' to this bit, (XOSC32KRDY), will clear the XOSC32K Ready Interrupt Enable bit, which disables the XOSC32K Ready interrupt

- Writing a one to the same corresponding bit in the INTENCLR register

Value	Description
0	The XOSC32K Ready interrupt is disabled.
1	The XOSC32K Ready interrupt is enabled.

22.7.3. Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x08
Reset: 0x00000000
Property: –

Note: Subsequent to an interrupt flag being cleared, the flag must be read back to verify the clear before exiting the ISR. Failure to do this can result in duplicate interrupts.

Table 22-4. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						XOSC32KFAIL		XOSC32KRDY
Access						R/W/HS/HC		R/W/HS/HC
Reset						0		0

Bit 2 – XOSC32KFAIL XOSC32K Clock Failure Detection

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

This bit is set by hardware under the following conditions:

- This flag is set by hardware on a zero-to-one transition of the XOSC32K Clock Failure Detection bit in the (STATUS.XOSC32KFAIL) register and will generate an Interrupt request if INTENSET.XOSC32KFAIL is '1'

This bit is cleared under the following conditions:

- Writing a '1' to this bit will clear the XOSC32K Clock Failure Detection flag

Value	Description
0	No XOSC32K 32.768kHz Clock Fail Detected.
1	XOSC32K 32.768kHz Clock Fail Detected.

Bit 0 – XOSC32KRDY XOSC32K Ready

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

This bit is set by hardware under the following conditions:

- 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

This bit is cleared under the following conditions:

- Writing a '1' to this bit will clear the XOSC32K Ready interrupt flag

Value	Description
0	The XOSC32K Ready interrupt is disabled.
1	The XOSC32K Ready interrupt is enabled.

22.7.4. Status

Name: STATUS
Offset: 0x0C
Reset: 0x00000000
Property: –

Table 22-5. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
					XOSC32KSW	XOSC32KFAIL		XOSC32KRDY
Access					R/HS/HC	R/HS/HC		R/HS/HC
Reset					0	0		0

Bit 3 – XOSC32KSW XOSC32K 32.768kHz Clock Switch

Note: This bit is set by hardware if a 32.768kHz clock fail detect occurs and clock fail detect is enabled, (i.e. CFDCTRL.CFDEN =1 and INTFLAG.XOSC32KFAIL=1 or STATUS.XOSC32KFAIL=1).

Value	Description
0	XOSC32K is not switched and the clock source is provided by the external 32.768kHz oscillator source.
1	XOSC32K is switched to the internal OSCULP32K oscillator clock. Both 32.768kHz and 1.024kHz outputs of the XOSC32K are replaced by the respective OSCULP32K 32 kHz and 1 kHz outputs to the respective logic modules.

Bit 2 – XOSC32KFAIL XOSC32K 32.768kHz Clock Failure Detect

Note: This bit is set by hardware if a 32.768kHz clock fail detect occurs and clock fail detect is enabled, (i.e. CFDCTRL.CFDEN =1 and INTFLAG.XOSC32KFAIL=1).

Value	Description
0	No XOSC32K 32.768kHz clock fail detection.
1	XOSC32K 32.768kHz external crystal/clock fail detect. Hardware clock switch will be initiated to OSCULP32K oscillator clocks.

Bit 0 – XOSC32KRDY XOSC32K Ready

Note: This bit is set and cleared by hardware based on the status of the of the active XOSC32K 32.768kHz clock source.

Value	Description
0	XOSC32K is not ready.
1	XOSC32K is stable and ready to be used as a clock source.

22.7.5. Clock Selection Control

Name: CLKSELCTRL
Offset: 0x10
Reset: 0x00000000
Property: PAC Write-Protection

Table 22-6. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

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

Bits 5:4 – HSMSEL[1:0] HASM - Hardware Security Module, (HSM), Clock Select
These bits select the HSM clock source.

Value	Name	Description
0x0	ULP32K	Clock from 32 kHz internal ULP oscillator
0x1	-	Reserved, (defaults to ULP32K)
0x2	XOSC32K	32.768 kHz from 32 kHz external crystal oscillator
0x3	-	Reserved, (defaults to ULP32K)

Note: If a reserved value is written, the ULP32K internal ULP oscillator is selected as source for the HSM clock by default.

Bits 1:0 – RTCSEL[1:0] RTC Clock Selection
These bits select the RTC clock source.

Value	Name	Description
0x0	ULP32K	32 kHz from 32 kHz internal ULP oscillator
0x1	ULP1K	1 kHz from 32 kHz internal ULP oscillator
0x2	XOSC32K	32.768 kHz from 32 kHz external crystal oscillator
0x3	XOSC1K	1.024 kHz from 32 kHz external oscillator

22.7.6. Clock Failure Detector Control

Name: CFDCTRL
Offset: 0x14
Reset: 0x00000000
Property: PAC Write-Protection

Table 22-7. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

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
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 is disabled.
1	The clock switch 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.
Note: After setting CFDEN to enable clock failure detection, STATUS.XOSC32KFAIL will always be set. This first detection must be ignored. Subsequent setting of this bit will indicate actual clock failure events.

Value	Description
0	The CFD is disabled.
1	The CFD is enabled.

22.7.7. Event Control

Name: EVCTRL
Offset: 0x18
Reset: 0x00000000
Property: PAC Write-Protection

Table 22-8. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

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.

Note: To prevent false event generation, the CFDEO bit must be set or cleared *only* when the XOSC32K is Disabled, (i.e. XOSC32K.ENABLE=0).

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.

22.7.8. 32 kHz External Crystal Oscillator (XOSC32K) Control

Name: XOSC32K
Offset: 0x1C
Reset: 0x00200080
Property: PAC Write-Protection

Table 22-9. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
			CGM[3:0]					
Access			R/W	R/W	R/W	R/W		
Reset			1	0	0	0		

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
	ONDEMAND					XTALEN	ENABLE	
Access	R/W					R/W	R/W	
Reset	1					0	0	

Bits 21:18 – CGM[3:0] Control Gain Mode
These bits control the gain of the external crystal oscillator.
These bits are enable-protected.

Value	Name	Description
0x0	CGM0	The lower Control Gain Mode value
0x1	CGM1	A higher Control Gain Mode value than CGM0
0x2	CGM2	A higher Control Gain Mode value than CGM1
0x3	CGM3	A higher Control Gain Mode value than CGM2
0x4	CGM4	A higher Control Gain Mode value than CGM3
0x5	CGM5	A higher Control Gain Mode value than CGM4: • Min Recommended for SF=3, ESR ≤ 100K • Min Recommended for SF=5, ESR ≤ 60K
0x6	CGM6	A higher Control Gain Mode value than CGM5
0x7	CGM7	A higher Control Gain Mode value than CGM6
0x8	CGM8	A higher Control Gain Mode value than CGM7
0x9	CGM9	A higher Control Gain Mode value than CGM8
0xA	CGM10	A higher Control Gain Mode value than CGM9, (Min Recommended for SF=5, ESR ≤ 100K)
0xB	CGM11	A higher Control Gain Mode value than CGM10
0xC	CGM12	A higher Control Gain Mode value than CGM11

CGM (continued)

Value	Name	Description
0xD	CGM13	A higher Control Gain Mode value than CGM12
0xE	CGM14	A higher Control Gain Mode value than CGM13
0xF	CGM15	The highest Control Gain Mode value

Note:

1. These bits are enable-protected. They cannot be written to if XOSC32K.ENABLE = 1.

Bits 11:8 – STARTUP[3:0] Oscillator Start-Up Time

These bits select the start-up time for the XOSC32K oscillator. The OSCULP32K oscillator is used to clock the start-up counter. The given time assumes an XOSC32K crystal frequency of 32.768 kHz. After the XOSC32K.STARTUP time has expired, the XOSC32K clock is released internally after the selected programmable startup clock cycles plus 3 additional XOSC32 periods. The Clock Fail Detect (CFD) monitoring also starts when the clock is released for internal use. The user selected start-up time must equal or exceed the start-up time defined in the electrical characteristics.

STARTUP[2:0]	OSCULP32K Clock Cycles	Plus	XOSC32K Clock Cycles	Equal	Approximate Equivalent Time
0x0	1	+	3	=	~122 µs
0x1	16	+	3	=	~580 µs
0x2	32	+	3	=	~1.07 ms
0x3	2048	+	3	=	~62.6 ms
0x4	4096	+	3	=	~125.1 ms
0x5	8192	+	3	=	~250.1 ms
0x6	16384	+	3	=	~500.1 ms
0x7	32768	+	3	=	~1s
0x8	65536	+	3	=	~2s
0x9	131072	+	3	=	~4s
0xA	262144	+	3	=	~8s
0xB – 0xF	---	---	---	---	Reserved

Note: These bits are enable-protected. They cannot be written to if XOSC32K.ENABLE = 1.
These bits are valid only when XOSC32K.XTALEN = 1, crystal XOSC32K selected.

Bit 7 – ONDEMAND On Demand Control

This bit controls how the XOSC32K behaves when a peripheral clock request is detected.
Note: This bit is enable-protected. It cannot be written to if XOSC32K.ENABLE = 1.

Value	Description
0	XOSC32K always run
1	Only run if requested by a peripheral

Bit 2 – XTALEN Crystal Oscillator Enable

This bit controls the connections between the I/O pads and the external clock or crystal oscillator.
Note: This bit is enable-protected. It cannot be written to if XOSC32K.ENABLE = 1.

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

Note: It is necessary to wait for STATUS.XOSC32KRDY = 0 before enabling the XOSC32K.

Value	Description
0	The 32K oscillator is disabled.

Value	Description
1	The 32K oscillator is enabled.

23. Watchdog Timer (WDT)

23.1. 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.

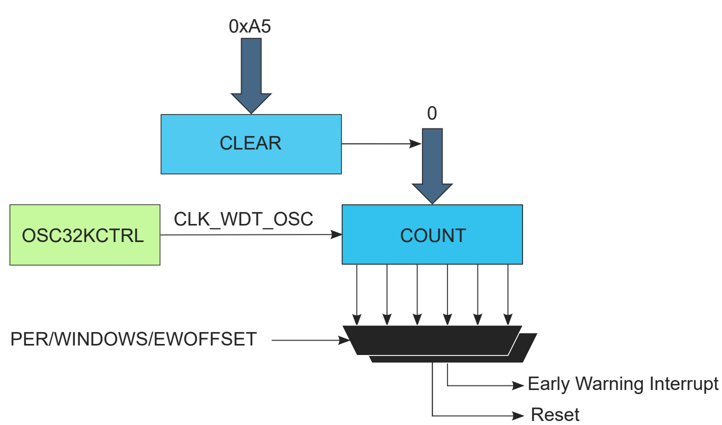
23.2. Features

The following are key features of the WDT module:

- 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

23.3. Block Diagram

Figure 23-1. WDT Block Diagram



23.4. Peripheral Dependencies

Peripheral Name	Base Address	NVIC IRQ Index: Source	MCLK AHBx/APBx Clock Enable Mask Bit	PAC Peripheral ID (PAC.WRCTRL.PERIDx)	Power Domain
WDT	0x4401_6000	11 : EW	MCLK.CLKMSK1[11]	11	VDDREG

23.5. Functional Description

23.5.1. 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 23-1. 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

23.5.2. Basic Operation

23.5.2.1. 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)

The 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.

23.5.2.2. Configurable Reset Values

After a Power-on Reset, some registers will be loaded with initial values from the *USER CFG page*.

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

23.5.2.3. 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'.

23.5.2.4. 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.

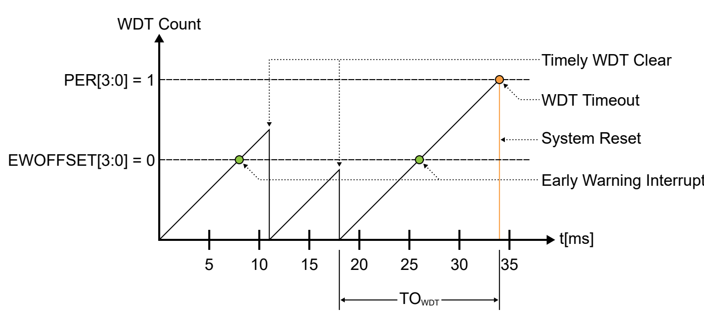
The WDT is cleared and a new WDT time-out period is started by writing 0xA5 to the Clear register (CLEAR). Writing any other value than 0xA5 to CLEAR will issue an immediate system reset.

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 23-2. Normal-Mode Operation



23.5.2.5. 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.

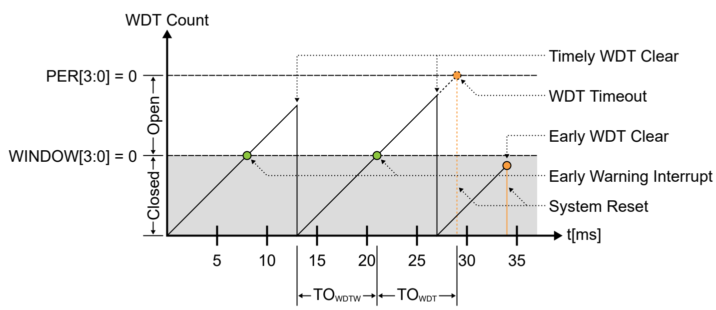
Both parameters TO_{WDTW} and TO_{WDT} are periods in a range from 8ms to 16s, so the total duration of the WDT time-out period is the sum of the two parameters.

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 SET 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 23-3. Window-Mode Operation



23.5.3. 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.

Note: The FREQM module can be used to calibrate the output of the OSCULP32K clock against a refererence clock, thereby reducing the uncertainty in setting WDT windows.

23.5.4. 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 INTFLAG register description for details on how to clear interrupt flags. All interrupt requests from the peripheral are ORed together at the system level to generate one combined interrupt request to the *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.

23.5.5. 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.

23.5.6. Debug Operation

When the CPU is halted in debug mode the WDT will halt normal operation.

23.5.7. 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*.

23.5.8. Additional Features

23.5.8.1. 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. Therefore, 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 23-2. 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

23.5.8.2. 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.

23.6. Register Summary

For descriptions and definitions of both Register and bitfield properties, refer to Register Properties.

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]							

23.6.1. Control A

Name: CTRLA
Offset: 0x00
Reset: x initially determined from USER CFG page after reset
Property: PAC Write-Protection, Write-Synchronized Bits, Enable-Protected Bits

Table 23-3. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
	ALWAYSON	RUNSTDBY				WEN	ENABLE	
Access	R/W/CFG	R/W/CFG				R/W/CFG	R/W/CFG	
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.
These bit is loaded from User Configuration FUCFG0 at startup.
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 bit is loaded from User Configuration FUCFG0 at start up.

Value	Description
0	The WDT is disabled during Standby sleep mode.
1	The WDT is enabled continues to operate during Standby sleep mode.

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.

These bit is loaded from User Configuration FUCFG0 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.

These bit is loaded from User Configuration FUCFG0 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.

23.6.2. Configuration

Name: CONFIG
Offset: 0x01
Reset: x initially determined from USER CFG page after reset
Property: PAC Write-Protection, Enable-Protected

Table 23-4. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
	WINDOW[3:0]				PER[3:0]			
Access	R/W/CFG	R/W/CFG	R/W/CFG	R/W/CFG	R/W/CFG	R/W/CFG	R/W/CFG	R/W/CFG
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 *User Configuration FUCFG0* at start-up.

Value	Description
`0x0`	8 1kHz clock cycles
`0x1`	16 1kHz clock cycles
`0x2`	32 1kHz clock cycles
`0x3`	64 1kHz clock cycles
`0x4`	128 1kHz clock cycles
`0x5`	256 1kHz clock cycles
`0x6`	512 1kHz clock cycles
`0x7`	1024 1kHz clock cycles
`0x8`	2048 1kHz clock cycles
`0x9`	4096 1kHz clock cycles
`0xA`	8192 1kHz clock cycles
`0xB`	16384 1kHz clock cycles
`0xC-0xF`	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 *User Configuration FUCFG0* at startup.

Value	Description
`0x0`	8 1kHz clock cycles
`0x1`	16 1kHz clock cycles
`0x2`	32 1kHz clock cycles
`0x3`	64 1kHz clock cycles
`0x4`	128 1kHz clock cycles
`0x5`	256 1kHz clock cycles
`0x6`	512 1kHz clock cycles
`0x7`	1024 1kHz clock cycles
`0x8`	2048 1kHz clock cycles
`0x9`	4096 1kHz clock cycles
`0xA`	8192 1kHz clock cycles

Value	Description
`0xB`	16384 1kHz clock cycles
`0xC - 0xF`	Reserved

23.6.3. Early Warning Control

Name: EWCTRL
Offset: 0x02
Reset: x initially determined from USER CFG page after reset
Property: PAC Write-Protection, Enable-Protected

Table 23-5. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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

Bits 3:0 – EWOFFSET[3:0] Early Warning Interrupt Time Offset
These bits determine the number of CLK_WDT_OSC clock cycles between the start of the watchdog time-out period and the generation of the Early Warning interrupt. These bits are loaded from *User Configuration FUCFG0* at start-up.

Value	Description
`0x0`	8 CLK_WDT_OSC clock cycles
`0x1`	16 CLK_WDT_OSC clock cycles
`0x2`	32 CLK_WDT_OSC clock cycles
`0x3`	64 CLK_WDT_OSC clock cycles
`0x4`	128 CLK_WDT_OSC clock cycles
`0x5`	256 CLK_WDT_OSC clock cycles
`0x6`	512 CLK_WDT_OSC clock cycles
`0x7`	1024 CLK_WDT_OSC clock cycles
`0x8`	2048 CLK_WDT_OSC clock cycles
`0x9`	4096 CLK_WDT_OSC clock cycles
`0xA`	8192 CLK_WDT_OSC clock cycles
`0xB`	16384 CLK_WDT_OSC clock cycles
`0xC - 0xF`	Reserved

23.6.4. 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.

Table 23-6. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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

Bit 0 – EW Early Warning Interrupt Disable
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.

23.6.5. 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.

Table 23-7. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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.

23.6.6. Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x06
Reset: 0x00
Property: -

Note: Subsequent to an interrupt flag being cleared, the flag must be read back to verify the clear before exiting the ISR. Failure to do this can result in duplicate interrupts.

Table 23-8. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
								EW
Access								R/W/HS
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.

23.6.7. Synchronization Busy

Name: SYNCBUSY
Offset: 0x08
Reset: 0x00000000
Property: -

Table 23-9. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
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.

Value	Description
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.

23.6.8. 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.

Table 23-10. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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.

24. Frequency Meter (FREQM)

24.1. Overview

The Frequency Meter (FREQM) can be used to accurately measure the frequency of a clock source, GCLK_FREQM_MSR, by comparing it to a known reference clock. It 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*GCLK_FREQM_REF and stored in the Value register (VALUE.VALUE). REFNUM is the number of Reference clock cycles selected in the Configuration A register (CFGA.REFNUM) over which the duration of a measurement is to be done.

The frequency of the measured clock, f_{CLK_MSR}, is calculated by:

$$f_{GCLK\_FREQM\_MSR} = \left(\frac{VALUE + 1 + Error}{REFNUM + 1}\right) f_{GCLK\_FREQM\_REF}$$

Where,

Error represents the error introduced by the synchronization mechanism. The error can be a maximum of two measured clock cycles.

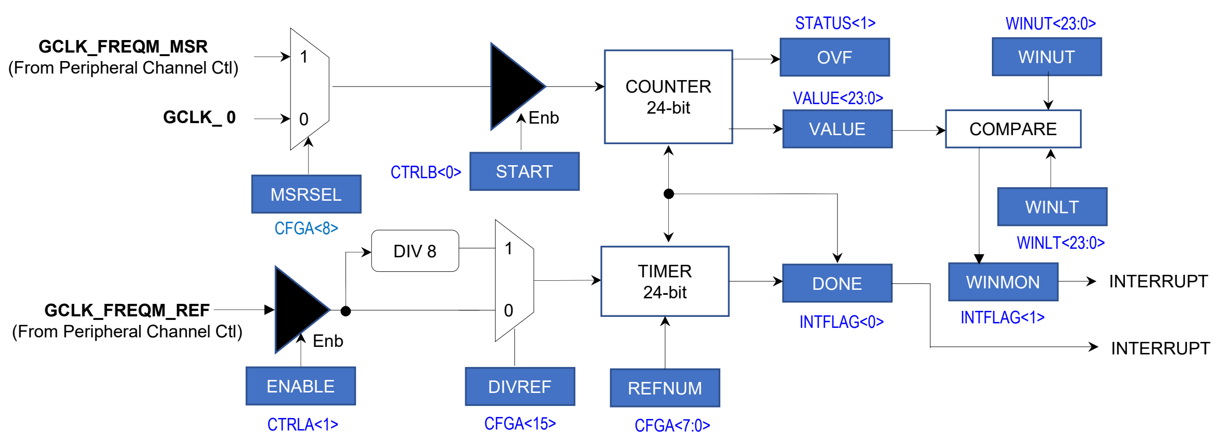
24.2. Features

The following are key features of the FREQM module:

- Ratio can be measured with 24-bit accuracy
- Accurately measures the frequency of an input clock source from GCLK_FREQM_MSR 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
- Free-running support enabling continuous clock monitor with flexible window threshold selection

24.3. Block Diagram

Figure 24-1. FREQM Block Diagram



24.4. Signal Description

Signal Name	Type	Description
GCLK_FREQM_MSR	Digital Clock Input	Measured clock source
GCLK_FREQM_REF	Digital Clock Input	Reference measurement clock source from peripheral channel
GCLK0	Digital Clock Input	Internal GCLK0

24.5. Peripheral Dependencies

Peripheral Name	Base Address	NVIC IRQ Index: Source	MCLK AHBx/APBx Clock Enable Mask Bit	GCLK Peripheral Channel Clock Name:Register	PAC Peripheral Identifier (PAC.WRCTRL.PERIDx)	EVSYS Users (EVSYS.USERm)	EVSYS Generator (EVSYS.CHANNELn.EVGENx)	Power Domain
FREQM	0x4401_4000	10 : DONE, WINMON	MCLK.CLKMSK1[10]	GCLK_FREQM_MSR0:GCLK.PCHCTRL[2] GCLK_FREQM_MSR1:PCHCTRL[3] GCLK_FREQM_REF:PCHCTRL[4]	10	0 : START	4 : DONE 5 : WINMON	VDDREG

24.6. Clocks

The clock for the FREQM bus interface (CLK_APB_FREQM) is enabled and disabled by the Main Clock Controller, the default state of CLK_APB_FREQM 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.

The FREQM has also the capability to use other internal clocks as source for frequency measurement. These clocks are chip specific and must be configured and enabled before using the FREQM.

24.7. Functional Description

24.7.1. Basic Operation

24.7.1.1. Initialization

Before enabling FREQM, the device and peripheral must be configured:

- The generic reference clock (GCLK_FREQM_REF) 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).

The following registers are enable-protected, meaning that they can only be written when the FREQM is disabled (CTRLA.ENABLE = 0):

- Configuration A register (CFGA)
- Control C register (CTRLC)
- Event Control register (EVCTRL)
- Window Monitor Lower Threshold register (WINLT)
- Window Monitor Upper Threshold register (WINUT)

Enable-protection is denoted by the "Enable-Protected" property in the register description.

The following register bits are enable-protected, meaning that they can only be written when the FREQM is disabled (CTRLA.ENABLE = 0):

- On Demand Control bit in Control A register (CTRLA.ONDEMAND)
- Run During Standby bit in Control A register (CTRLA.RUNSTDBY)
- Free Running Mode bit in Control A register (CTRLA.FREERUN)

The 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'.

24.7.1.2. 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.

24.7.1.3. 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:

1. 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 the Status register (STATUS.BUSY) is set when the measurement starts, and cleared when the measurement is complete.

Note: Writing a '1' to the START bit in the Control B register during a measurement will be ignored.

The Measurement Done interrupt provides an alternative way to detect measurement completion. 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:

$$f_{CLK\_MSR} = \left(\frac{VALUE\ +\ 1\ +\ Error}{REFNUM\ +\ 1}\right) f_{GCLK\_FREQM\_REF}$$

Where,

Error represents the error introduced by the synchronization mechanism. The error can be a maximum of two CLK_MSR cycles.

Notes:

1. To make sure the measurement result (VALUE.VALUE[23:0]) is valid, the overflow status (STATUS.OVF) must be checked.
2. Due to asynchronous operation, 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.

Note: The REFNUM field must be written before the FREQM is enabled.

24.7.1.4. Window Monitor Mode

Window Monitor Mode causes the FREQM to compare each measurement result to pre-configured thresholds. It can compare the value to see if it is below a lower, above an upper, within the upper and lower, or outside of the upper and lower thresholds. The Window Monitor Mode is selected by setting the Window Mode bits in the Control C register (CTRLC.WINMODE). Threshold values must be written in the Window Monitor Lower Threshold register (WINLT.WINLT) and Window Monitor Upper Threshold register (WINUT.WINUT).

The FREQM also supports an interrupt request for Window Monitor operation: When the Window Monitor bit in Interrupt Enable Set register (INTENSET.WINMON) is '1' and a measurement meets the criteria defined in the Window Mode bits in the Control C register (CTRLC.WINMODE), the Window Monitor bit in the Interrupt Flag Status and Clear register (INTFLAG.WINMON) will be set and an interrupt request is generated.

24.7.1.5. Free Running Mode

Free Running Mode causes the FREQM to automatically start a new measurement after each measurement is complete. The Free Running Mode can be disabled by disabling the module, or by setting the Software Reset bit in the Control A register (CTRLA.SWRST = 1).

Important: In free running mode, when a new measurement starts, the VALUE register will follow the MSR counter value. It is not recommended to read the VALUE register when the free running operating mode is enabled.

Notes:

1. In free running mode, the event detection is disabled. As consequence, all incoming events are ignored and no actions will be triggered inside the FREQM module.
2. Writing to CTRLB.START when a measurement is in progress is ignored. The bit is always read zero.

24.7.2. Interrupts

The FREQM has two interrupt sources:

- DONE: A frequency measurement is done.
- WINMON: A window compare valid condition is detected.

The interrupt flag in the Interrupt Flag Status and Clear (INTFLAG) register is set when the interrupt condition occurs. The interrupt can be enabled by writing a '1' to the corresponding bit in the Interrupt Enable Set (INTENSET) register, and disabled by writing a '1' to the corresponding bit in the Interrupt Enable Clear (INTENCLR) register.

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 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. The user must read the INTFLAG register to determine which interrupt condition is present.

This interrupt is a synchronous wake-up source.

Note: Interrupts must be globally enabled for interrupt requests to be generated.

24.7.3. Events

The FREQM can generate the following output events:

- Done (DONE): Measurement Done.

- Window Monitor (WINMON): Generated when the window monitor condition matches.

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 FREQM can take the following actions on an input event:

- **Start measurement (START)**: Start a new measurement. If a new START event is detected during an ongoing measurement, the event is ignored.

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 FREQM will detect a rising edge on the incoming event. If the FREQM 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.STARTINV).

24.7.4. Sleep Mode Operation

During sleep modes, the ONDEMAND and RUNSTDBY bits in the Control A register (CTRLA) controls the behavior of the FREQM clock requests, in cases where the FREQM is enabled (CTRLA.ENABLE = 1) during a Sleep mode.

Table 24-1. FREQM Sleep Behavior

CTRLA.RUNSTDBY	CTRLA.ONDEMAND	CTRLA.ENABLE	Description
X	x	0	Disabled
0	0	1	Always request the clocks in all sleep modes, except in Standby mode.
0	1	1	On-demand clocks request in all sleep modes, except in Standby mode.
1	0	1	Always request the clocks in all sleep modes.
1	1	1	On-demand clocks request in all sleep modes.

Note: By default, if RUNSTDBY = 0, an ongoing measurement is stopped when the device is going to Standby Sleep mode. A new measurement will be restarted when the device exits the Standby Sleep mode.

24.8. Register Summary

For descriptions and definitions of both Register and bitfield properties, refer to Register Properties.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRLA	7:0	ONDEMAND	RUNSTDBY				FREERUN	ENABLE	SWRST
0x01	CTRLB	7:0								START
0x02	CFGA	15:8	DIVREF							MSREL
		7:0	REFNUM[7:0]							
0x04	CTRLC	7:0						WINMODE[2:0]		
0x05	Reserved									
0x06	EVCTRL	7:0			WINMONEO	DONEEO			STARTINV	STARTEI
0x07	Reserved									
0x08	INTENCLR	7:0							WINMON	DONE
0x09	INTENSET	7:0							WINMON	DONE
0x0A	INTFLAG	7:0							WINMON	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]							
0x14 ... 0x1F	Reserved									
0x20	WINLT	31:24								
		23:16	WINLT[23:16]							
		15:8	WINLT[15:8]							
		7:0	WINLT[7:0]							
0x24	WINUT	31:24								
		23:16	WINUT[23:16]							
		15:8	WINUT[15:8]							
		7:0	WINUT[7:0]							

24.8.1. Control A

Name: CTRLA
Offset: 0x00
Reset: 0x80
Property: PAC Write-Protection

Table 24-2. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
	ONDEMAND	RUNSTDBY				FREERUN	ENABLE	SWRST
Access	R/W	R/W				R/W	R/W	R/W
Reset	1	0				0	0	0

Bit 7 – ONDEMAND On Demand Control

During Sleep mode, the On Demand mode causes the FREQM clocks to be running when a measurement is requested by a peripheral or if free running operation mode is enabled. If there is no peripheral requesting the FREQM measurement, the module will not request any clock.

This bit is enable-protected.

Note: The ONDEMAND bit should always be set to '0'.

Value	Description
0	The FREQM is always requesting the clocks when CTRLA.ENABLE is set.
1	The FREQM is requesting the clocks only when a peripheral is requesting a measurement, or if free running mode is enabled.

Bit 6 – RUNSTDBY Run in Standby

This bit controls how the FREQM clocks will be requested during Standby S leep mode.

This bit is enable-protected.

Value	Description
0	The FREQM clocks are not requested during Standby Sleep mode. If CTRLA.ONDEMAND = 1, the FREQM will request the clocks only if a measurement is requested. If CTRLA.ONDEMAND = 0, the FREQM clocks will be requested as long as CTRLA.ENABLE is set.
1	The FREQM clocks are requested in standby sleep mode. If CTRLA.ONDEMAND = 1, the FREQM will request the clocks only if a measurement is requested. If CTRLA.ONDEMAND = 0, the FREQM clocks will be requested as long as CTRLA.ENABLE is set.

Bit 2 – FREERUN Free Running Mode

This bit controls the free running operation mode.

This bit is enable-protected.

Value	Description
0	The free running operating mode is disabled.
1	The free running operating mode is enabled.

Bit 1 – ENABLE Enable

Due to synchronization there is delay from writing CTRLA.ENABLE until the peripheral is enabled/disabled. The value written to CTRLA.ENABLE will read back immediately and the ENABLE bit in the Synchronization Busy register (SYNCBUSY.ENABLE) will be set. SYNCBUSY.ENABLE will be cleared when the operation is complete.

This bit is not enable-protected.

Value	Description
0	The peripheral is disabled.
1	The peripheral is enabled.

Bit 0 – SWRST Software Reset

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

Writing a '1' to this bit resets all registers in the FREQM to their initial state, and the FREQM 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.

Due to synchronization there is a delay from writing CTRLA.SWRST until the Reset is complete. CTRLA.SWRST and SYNCBUSY.SWRST will both be cleared when the Reset is complete.

This bit is not enable-protected.

Notes:

1. When the CTRLA.SWRST is written, the user should poll SYNCB.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.

24.8.2. Control B

Name: CTRLB
Offset: 0x01
Reset: 0x00
Property: –

Table 24-3. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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

Bit 0 – START Start Measurement

Value	Description
0	Writing a '0' has no effect.
1	Writing a '1' starts a single measurement.

24.8.3. Configuration A

Name: CFGA
Offset: 0x02
Reset: 0x0000
Property: PAC Write-Protection, Enable-protected

Table 24-4. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	15	14	13	12	11	10	9	8
	DIVREF							MSREL
Access	R/W							R/W
Reset	0							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.

Bit 8 – MSREL Frequency Meter Clock Measure Selection

Value	Description
0	Select GCLK_FREQM_MSR as FREQM clock to measure
1	Select GCLK_ 0 as FREQM clock to measure

Bits 7:0 – REFNUM[7:0] Number of Reference Clock Cycles
Selects the duration of a measurement in number of CLK_FREQM_REF cycles.
Note: The measurement reference period must be longer than 4 APB clock periods.

24.8.4. Control C

Name: CTRLC
Offset: 0x04
Reset: 0x0000
Property: PAC Write-Protection, Enable-Protected

Table 24-5. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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

Bits 2:0 – 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	VALUE > WINLT
0x2	MODE2	VALUE < WINUT
0x3	MODE3	WINLT < VALUE < WINUT
0x4	MODE4	!(WINLT < VALUE < WINUT)
0x5 - 0x7		Reserved

24.8.5. Event Control

Name: EVCTRL
Offset: 0x06
Reset: 0x00
Property: PAC Write-Protection, Enable-Protected

Table 24-6. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
			WINMONEO	DONEEO			STARTINV	STARTEI
Access			R/W	R/W			R/W	R/W
Reset			0	0			0	0

Bit 5 – WINMONEO Window Monitor Event Out

Value	Description
0	Window Monitor event output is disabled and an event will not be generated.
1	Window Monitor event output is disabled and an event will be generated.

Bit 4 – DONEEO Measurement Done Event Out

Value	Description
0	Measurement Done event output is disabled and an event will not be generated.
1	Measurement Done event output is disabled and an event will be generated.

Bit 1 – STARTINV Start Measurement Event Invert Enable

Value	Description
0	Start event input source is not inverted.
1	Start event input source is inverted.

Bit 0 – STARTEI Start Measurement Event Input Enable

Value	Description
0	A new measurement will not be triggered on any incoming event.
1	A new measurement will be triggered on any incoming event.

24.8.6. Interrupt Enable Clear

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

Table 24-7. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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

Bit 1 – WINMON Window Monitor Interrupt Disable
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 Window Monitor interrupt.

Value	Description
0	The Window Monitor interrupt is disabled.
1	The Window Monitor interrupt is enabled.

Bit 0 – DONE Measurement Done Interrupt Disable
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.

24.8.7. Interrupt Enable Set

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

Table 24-8. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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

Bit 1 – WINMON Measurement Done Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Window Monitor Interrupt Enable bit, which enables the Window Monitor interrupt.

Value	Description
0	The Window Monitor interrupt is disabled.
1	The Window Monitor interrupt is enabled.

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.

24.8.8. Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x0A
Reset: 0x00
Property: –

Note: Subsequent to an interrupt flag being cleared, the flag must be read back to verify the clear before exiting the ISR. Failure to do this can result in duplicate interrupts.

Table 24-9. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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

Bit 1 – WINMON Window Monitor

This flag is set on the next clock cycle after a match with the window monitor condition, and an interrupt request will be generated if INTENCLR/SET.WINMON is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the WINMON interrupt flag.
Note: This Flag will set after the TIMER has expired and a match with the window monitor condition occurred. There will be some synchronization delay from the GCLK to APB clock domain after the TIMER has expired. To avoid the user confusion, the INTFLAG.DONE and the INTFLAG.WINMON should be set on the same APB clock edge.

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.

24.8.9. Status

Name: STATUS
Offset: 0x0B
Reset: 0x00
Property: –

Table 24-10. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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 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. This bit is set only when the measurements are triggered by a software start (CTRLB.START = 1).

24.8.10. Synchronization Busy

Name: SYNCBUSY
Offset: 0x0C
Reset: 0x00000000
Property: –

Table 24-11. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

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

Bit 1 – ENABLE Enable Synchronization Busy Flag
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 Software Reset Synchronization Busy Flag
This bit is cleared when the synchronization of CTRLA.SWRST is complete.
This bit is set when the synchronization of CTRLA.SWRST is started.

24.8.11. Value

Name: VALUE
Offset: 0x10
Reset: 0x00000000
Property: –

Table 24-12. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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.
Note: The register value is refreshed each time a start event is detected, and as long as a measurement is ongoing. It is not recommended to read the register if the start of measurements are triggered by peripheral events or it is in Free Running Mode.

24.8.12. Window Monitor Lower Threshold

Name: WINLT
Offset: 0x20
Reset: 0x0000
Property: PAC Write-Protection, Enable-Protected

Table 24-13. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	WINLT[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
	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 23:0 – WINLT[23:0] Window Lower Threshold

When the Window Monitor bit in Interrupt Enable Set register, INTENSET.WINMON= 1, and the VALUE register result of the measurement depends on CTRLC.WINMON, the Window Monitor bit in the Interrupt Flag Status and Clear register, INTFLAG.WINMON, will be set and an interrupt request is generated.

24.8.13. Window Monitor Upper Threshold

Name: WINUT
Offset: 0x24
Reset: 0x0000
Property: PAC Write-Protection, Enable-Protected

Table 24-14. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	WINUT[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
	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 23:0 – WINUT[23:0] Window Upper Threshold

When the Window Monitor bit in Interrupt Enable Set register, INTENSET.WINMON= 1, and the VALUE register result of the measurement depends on CTRLC.WINMON, the Window Monitor bit in the Interrupt Flag Status and Clear register, INTFLAG.WINMON, will be set and an interrupt request is generated.

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, 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

The following are key features of the RTC module:

- 32-bit counter with 10-bit prescaler
- Multiple clock sources
- 32-bit or 16-bit counter mode
- Two 32-bit or four 16-bit compare values
- Clock/Calendar mode
  - Time in seconds, minutes, and hours (12/24)
  - Date in day of month, month, and year
  - Leap year correction
- Digital prescaler correction/tuning for increased accuracy
- Overflow, alarm/compare match and prescaler interrupts and events
  - Optional clear on alarm/compare match
- Four general purpose registers
- Tamper Detection
  - Up to 8 static tamper inputs with programmable level detection
  - Up to 8 inputs and 8 outputs dynamic tamper (Active layer protection)
  - Support for any kinds of static and dynamic combinations
  - Timestamp on tampers event

25.3. Block Diagram

Figure 25-1. RTC Block Diagram (Mode 0 — 32-Bit Counter)

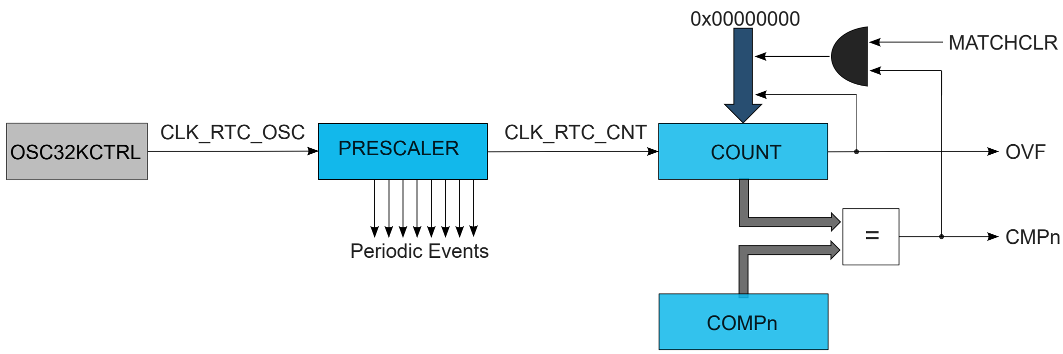


Figure 25-2. RTC Block Diagram (Mode 1 — 16-Bit Counter)

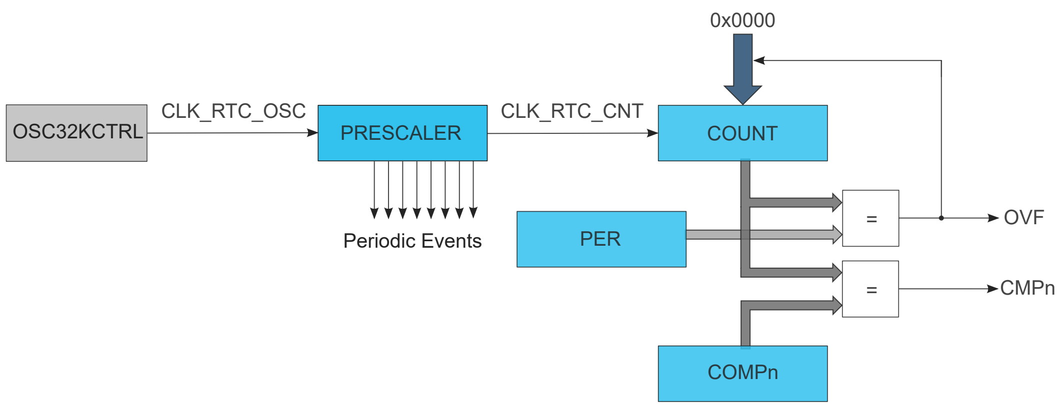


Figure 25-3. RTC Block Diagram (Mode 2 — Clock/Calendar)

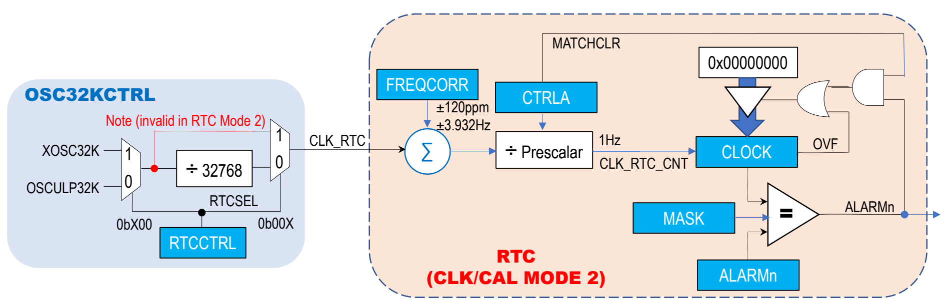
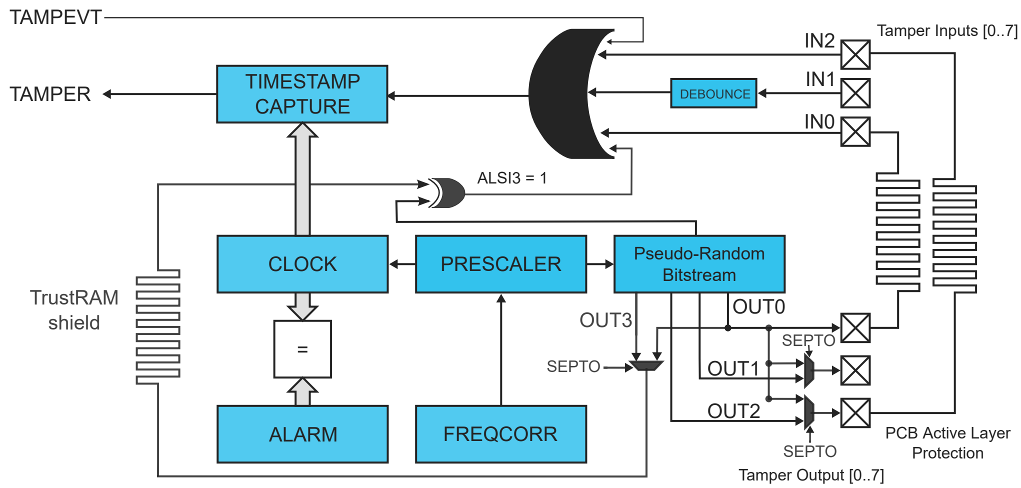


Figure 25-4. RTC Block Diagram (Tamper Detection Use Case)



25.4. Signal Description

Table 25-1. Signal Description

Signal	Description	Type
RTC_IN[7:0] (1)	Tamper Detection Input	Digital input
RTC_OUT[7:0] (1)	Tamper Detection Output	Digital output

Notes:

1. The RTC_IN[5:7] and RTC_OUT[5:7] are not available on 100-pin packages .
2. Only the IO pins in the VDDCORE_BU power domain can be used to wakeup the device from Hibernate or Backup sleep modes, when using the RTC Tamper Detection features (Wake/Capture/Active Layer Protection). These pins are RTC_IN0, RTC_IN2, RTC_IN3, RTC_IN4, RTC_OUT0.

25.5. Peripheral Dependencies

Peripheral Name	Base Address	NVIC IRQ Index: Source	MCLK AHBx/APBx Clock Enable Mask Bit	PAC Peripheral Identifier (PAC.WRCTRL.PERIDx)	EVSYS Users (EVSYS.USERm)	EVSYS Generator (EVSYS.CHANNELn.EVGENx)	DMA Trigger Index:Source (DMAC.CHCTRLBk.TRIGx)	Power Domain
RTC	0X4401_8000	12 : PER_0, PER_1, PER_2, PER_3, PER_4, PER_5, PER_6, PER_7, CMP_0, CMP_1, CMP_2, CMP_3, TAMPER, OVF	MCLK.CLKMSK1[12]	12	1 : TAMPER	6-13 : Periodic Interval n (RTC_PERn), n=0,1...7 14 : Daily Period (RTC_PERD) 15-18 : Compare n (RTC_CMPn), n=0,1,2,3 19 : Tamper 20 : Overflow (OVF)	4 : RTC Timestamp (TIMESTAMP)	VDDCORE_BU

25.5.1. 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 Peripheral Clock Masking section.

A 32KHz or 1KHz oscillator clock (CLK_RTC_OSC) is required to clock the RTC. This clock must be configured and enabled in the 32KHz 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.

25.5.2. 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 DBGCTRL for details.

25.5.3. Register Access Protection

All registers with write-access are optionally write-protected by the peripheral access controller (PAC), except the following registers:

- Interrupt Flag Status and Clear (INTFLAG) register

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

Write-protection does not apply to accesses through an external debugger. Refer to the *PAC - Peripheral Access Controller* for details.

25.5.4. Analog Connections

A 32.768kHz crystal can be connected to the XIN32 and XOUT32 pins, along with any required load capacitors. See the *Electrical Characteristics* for details on recommended crystal characteristics and load capacitors.

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:

$$f_{\mathrm{CLK\_RTC\_CNT}} = \frac{f_{\mathrm{CLK\_RTC\_OSC}}}{2^{\mathrm{PRESCALER}}}$$

The frequency of the oscillator clock, CLK_RTC_OSC, is given by $f_{CLK\_RTC\_OSC}$, and $f_{CLK\_RTC\_CNT}$ is the frequency of the internal prescaled RTC clock, CLK_RTC_CNT.

25.6.2.2. Enabling, Disabling, and Resetting

The RTC is enabled by setting the Enable bit in the Control A register (CTRLA.ENABLE=1). The RTC is disabled by writing CTRLA.ENABLE=0.

The RTC is reset by setting the Software Reset bit in the Control A register (CTRLA.SWRST=1). All registers in the RTC, except DEBUG, will be reset to their initial state, and the RTC will be disabled. The RTC must be disabled before resetting it.

25.6.2.3. 32-Bit Counter (Mode 0)

When the RTC Operating Mode bits in the Control A register (CTRLA.MODE) are written to 0x0, the counter operates in 32-bit Counter mode. The block diagram of this mode is shown in the Block Diagram. 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 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.

If the Clear on Match bit in the Control A register (CTRLA.MATCHCLR) is '1', the counter is cleared on the next counter cycle when a compare match with COMPn occurs. This allows the RTC to generate periodic interrupts or events with longer periods than the prescaler events. Note that when CTRLA.MATCHCLR is '1', INTFLAG.CMPn and INTFLAG.OVF will both be set simultaneously on a compare match with COMPn.

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 the Block Diagram. 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..3). When a compare match occurs, the Compare n Interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.CMPn, n=0..3) 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 the Block Diagram. 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 registers (ALARMn, n=0,1). When an alarm match occurs, the Alarm n Interrupt flag in the Interrupt Flag Status and Clear registers (INTFLAG.ALARMn) is set on the next 0-to-1 transition of CLK_RTC_CNT. E.g. For a 1Hz clock counter, it means the Alarm n 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 n Mask register (MASKn.SEL, n=0,1). These bits determine which time/date fields of the clock and alarm values are valid for comparison and which are ignored.

If the Clear on Match bit in the Control A register (CTRLA.MATCHCLR) is set, the counter is cleared on the next counter cycle when an alarm match with ALARMn occurs. This allows the RTC to generate periodic interrupts or events with longer periods than it would be possible with the prescaler events only (see Periodic Intervals).
Note: When CTRLA.MATCHCLR is 1, INTFLAG.ALARMn and INTFLAG.OVF will both be set simultaneously on an alarm match with ALARMn.

25.6.3. DMA Operation

The RTC generates the following DMA request:

- Tamper (TAMPER): The request is set on capture of the timestamp. The request is cleared when the Timestamp register is read.

If the CPU accesses the registers which are source for DMA request set/clear condition, the DMA request can be lost or the DMA transfer can be corrupted, if enabled.

25.6.4. Interrupts

The RTC has the following interrupt sources:

- Overflow (OVF): Indicates that the counter has reached its top value and wrapped to zero.
- Tamper (TAMPER): Indicates detection of valid signal on a tamper input pin or tamper event input.
- Compare (CMPn): Indicates a match between the counter value and the compare register.
- Alarm (ALARMn): Indicates a match between the clock value and the alarm register.
- Period n (PERn): The corresponding bit in the prescaler has toggled. Refer to *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 or more 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.

25.6.5. 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 (CMPn): Indicates a match between the counter value and the compare register.
- Alarm (ALARMn): Indicates a match between the clock value and the alarm register.
- Period n (PERn): The corresponding bit in the prescaler has toggled. Refer to *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 Enable bit ((EVCTRL.TAMPEVEI)) into the Event Control register (EVCTRL) enables the action on input events. Writing a zero to this bit disables the action on input events.

25.6.6. Sleep Mode Operation

The RTC will continue to operate in any sleep mode where the source clock is active. The RTC *interrupts* can be used to wake up the device from a sleep mode. RTC *events* can trigger other operations in the system without exiting the sleep mode.

An interrupt request will be generated after the wake-up if the Interrupt Controller is configured accordingly. Otherwise the CPU will wake up directly, without triggering any interrupt. In this case, the CPU 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.

25.6.7. Synchronization

Due to asynchronicity between the main clock domain and the peripheral clock domains, some registers need to be synchronized when written or read.

The following bits are synchronized when written:

- Software Reset bit in Control A register, CTRLA.SWRST
- Enable bit in Control A register, CTRLA.ENABLE
- Count Read Synchronization bit in Control A register (CTRLA.COUNTSYNC)
- Clock Read Synchronization bit in Control A register (CTRLA.COUNTSYNC)

The following registers are synchronized when written:

- Counter Value register, COUNT
- Clock Value register, CLOCK
- Counter Period register, PER
- Compare n Value registers, COMPn
- Alarm n Value registers, ALARMn
- Frequency Correction register, FREQCORR
- Alarm n Mask register, MASKn
- The General Purpose n registers (GPn)

The following registers are synchronized when read:

- The Counter Value register, COUNT, if the Counter Read Sync Enable bit in CTRLA (CTRLA.COUNTSYNC) is '1'
- The Clock Value register, CLOCK, if the Clock Read Sync Enable bit in CTRLA (CTRLA.CLOCKSYNC) is '1'
- The Timestamp Value register (TIMESTAMP)

Required write synchronization is denoted by the "Write-Synchronized" property in the register description.

Required read synchronization is denoted by the "Read-Synchronized" property in the register description.

25.6.8. Additional Features

25.6.8.1. Periodic Intervals

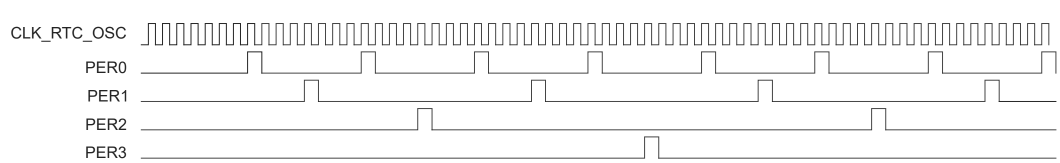
The RTC prescaler can generate interrupts and events at periodic intervals, allowing flexible system tick creation. Any of the upper eight bits of the prescaler (bits 2 to 9) can be the source of an interrupt/event. When one of the eight Periodic Event Output bits in the Event Control register (EVCTRL.PEREO[n=0..7]) is '1', an event is generated on the 0-to-1 transition of the related bit in the prescaler, resulting in a periodic event frequency of:

$$f_{\mathrm{PERIODIC(n)}} = \frac{f_{\mathrm{CLK\_RTC\_OSC}}}{2^{\mathrm{n}+3}}$$

$f_{CLK\_RTC\_OSC}$ 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



25.6.8.2. Frequency Correction

The RTC Frequency Correction module employs periodic counter corrections to compensate for a too-slow or too-fast oscillator. Frequency correction requires that CTRLA.PRESCALER is greater than 1.

The digital correction circuit adds or subtracts cycles from the RTC prescaler to adjust the frequency in approximately 1ppm steps. Digital correction is achieved by adding or skipping a single count in the prescaler once every 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.

25.6.8.3. 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 complete (bit SYNCBUSY.COMPn going to 0). After the write is completed the write buffer can be used as a odd general purpose register without affecting the COMPARE/ALARM function.

If the COMPARE/ALARM function is not used, the read buffer can be used as an even general purpose register. In this case writing the even GP will temporarily use the write buffer until the synchronization is complete (bit SYNCBUSY.GPn going to 0). Therefore an even GP must be written before writing the odd GP. Changing or writing an even GP needs to temporarily save the value of the odd GP.

Before using an even GP, the associated COMPARE/ALARM feature must be disabled by writing a '1' to the General Purpose Enable bit in the Control B register (CTRLB.GPnEN). To re-enable the compare/alarm, CTRLB.GPnEN must be written to zero and the associated COMPn/ALARMn must be written with the correct value.

An example procedure to write the general purpose registers GP0 and GP1 is:

1. Wait for any ongoing write to COMP0 to complete (SYNCBUSY.COMP0 = 0). If the RTC is operating in Mode 1, wait for any ongoing write to COMP1 to complete as well (SYNCBUSY.COMP1 = 0).
2. Write CTRLB.GP0EN = 1 if GP0 is needed.
3. Write GP0 if needed.
4. Wait for any ongoing write to GP0 to complete (SYNCBUSY.GP0 = 0). Note that GP1 will also show as busy when GP0 is busy.
5. Write GP1 if needed.

The following table provides the correspondence of General Purpose Registers and the COMPARE/ALARM read or write buffer in all RTC modes.

Table 25-2. General Purpose Registers Versus Compare/Alarm Registers: n in 0, 2, 4, 6...

Register	Mode 0	Mode 1	Mode 2	Write Before
GPn	COMPn/2 write buffer	(COMPn , COMPn+1) write buffer	ALARM0 write buffer	GPn+1
GPn+1	COMPn/2 read buffer	(COMPn , COMPn+1) read buffer	ALARM0 read buffer	-

25.6.8.4. 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 Layer Protection: A mismatch of an internal RTC signal routed between INn and OUTn pins will be detected and the tamper interrupt flag (INTFLAG.TAMPER) will be set. The RTC value will be captured in the TIMESTAMP register.

In order to determine which tamper source caused a tamper event, the Tamper ID register (TAMPID) provides the detection status of each tamper channel. These bits remain active until cleared by software.

A single interrupt request (TAMPER) is available for all tamper channels.

The RTC also supports an input event (TAMPEVT) for generating a tamper condition within the Event System. The tamper input event is enabled by the Tamper Input Event Enable bit in the Event Control register (EVCTRL.TAMPEVTEI).

Up to eight 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 *Edge Detection with Debouncer Enabled* 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 *Edge Detection with Synchronous Stability Debouncing* 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 Edge Detection with Asynchronous Stability Debouncing 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 *Edge Detection with Majority Debouncing* for an example.

Table 25-3. Debouncer Configuration

TAMPCTRL. DEBNCn	CTRLB. DEBMAJ	CTRLB. DEBASYNC	Description
0	X	X	Detect edge on INn with no debouncing. Every edge detected is immediately triggered.
1	0	0	Detect edge on INn with synchronous stability debouncing. Edge detected is only triggered when INn is stable for 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

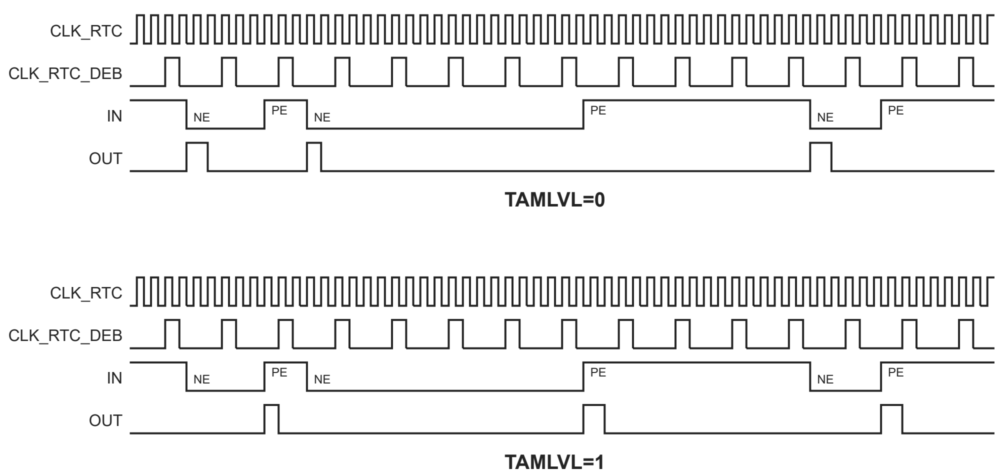
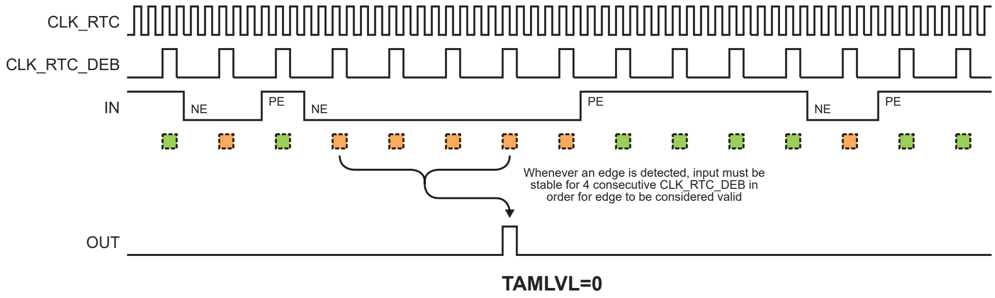


Figure 25-7. Edge Detection with Synchronous Stability Debouncing



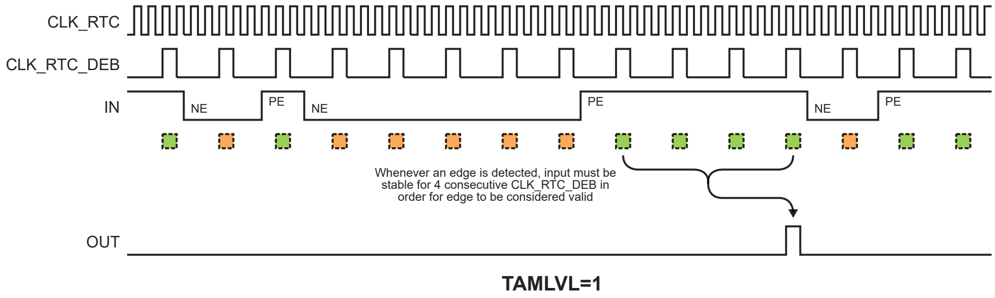
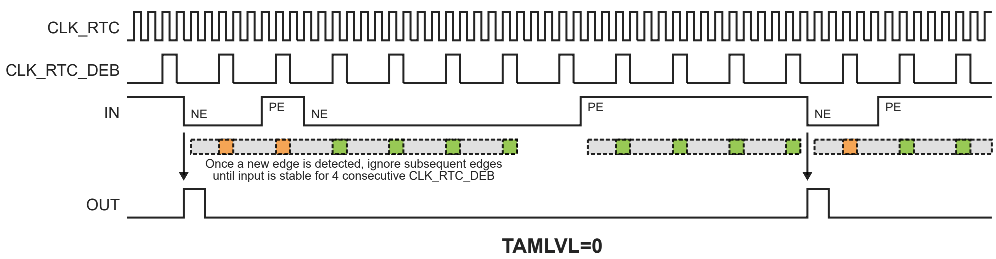


Figure 25-8. Edge Detection with Asynchronous Stability Debouncing



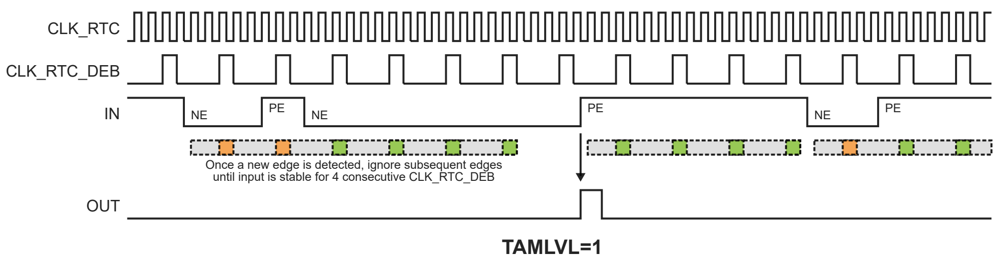
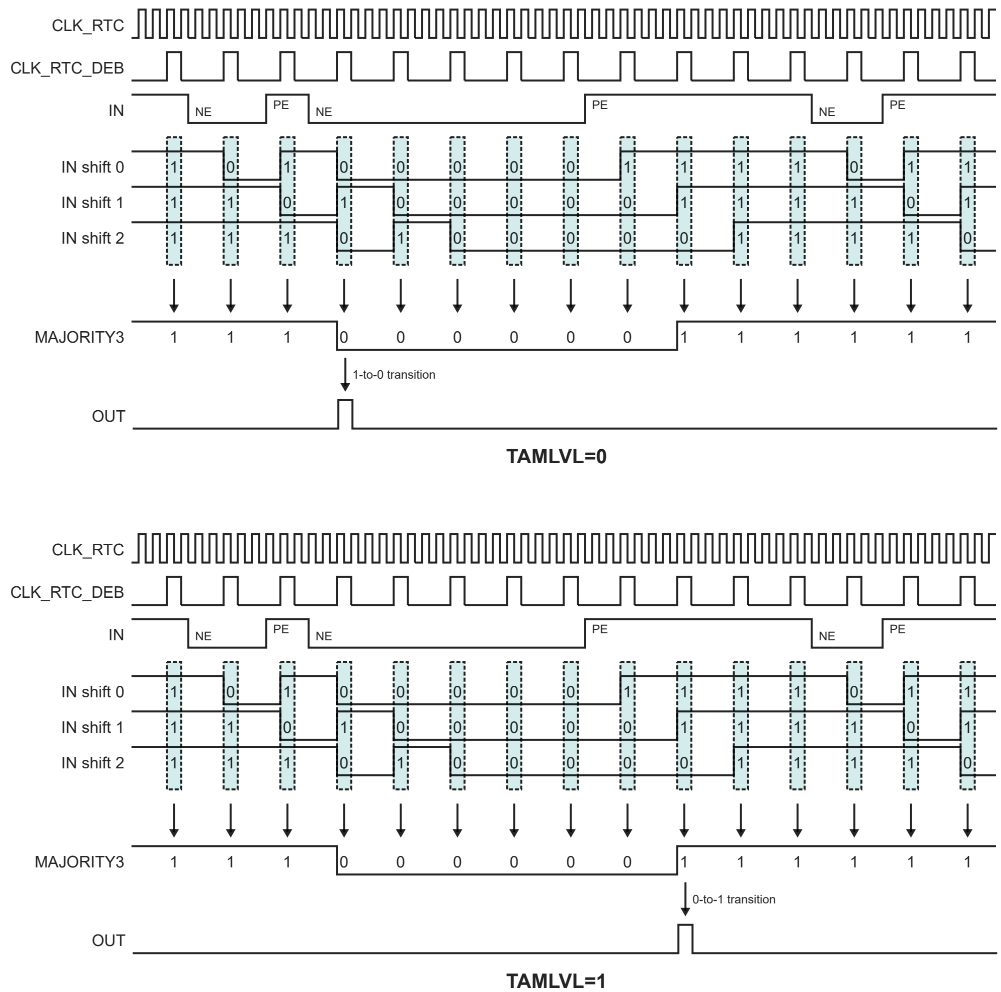


Figure 25-9. Edge Detection with Majority Debouncing



25.6.8.4.1. 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 until the Tamper flag is cleared, either by reading the timestamp or by writing a '1' to INTFLAG.TAMPER. If several tamper conditions occur in a short window before the flag is cleared, only the first timestamp may be logged. However, the detection of each tamper will still be recorded in TAMPID.

The Tamper Input Event (TAMPEVT) will always perform a timestamp capture. To capture on the external inputs (INn), the corresponding Input Action field in the Tamper Control register (TAMPCTRL.INnACT) must be written to '1'. If an input is set for wake functionality it does not capture the timestamp; however the Tamper flag and TAMPID will still be updated.

25.6.8.4.2. Active Layer Protection

The RTC provides a mean of detecting broken traces on the PCB, also known as Active layer Protection. In this mode, a generated internal RTC signal can be directly routed over critical components on the board using RTC OUT output pin to one RTC INn input pin. A tamper condition is detected if there is a mismatch on the generated RTC signal.

The Active Layer Protection mode and the generation of the RTC signal is enabled by setting the RTCOUT bit in the Control B register (CTRLB.RTCOUT).

Enabling active layer protection 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 Layer 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 layer mode by writing 3 to the corresponding Input Action field in the Tamper Control register (TAMPCTRL.INnACT). When active layer protection is enabled and INn and OUTn pin are used, the value of INn is sampled on the rising edge of CLK_RTC and compared to the expected value of OUTn. Therefore up to one CLK_RTC period is available for propagation delay through the trace.
- Select Active Layer Monitoring Source (TrustRAM or INn/OUTn tamper pins) using the ALSIn bit of the TAMPCTRLB register.
- Enable Active Layer Protection by setting the CTRLB.RTCOUT bit.

25.7. Register Summary - 32-bit Counter Mode

For descriptions and definitions of both Register and bitfield properties, refer to Register Properties.

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			GP2EN	GP0EN
0x04	EVCTRL	31:24								PERDEO
		23:16								TAMPEVEI
		15:8	OVFEO	TAMPEREO					CMPEO0[1:0]	
		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					GP3	GP2	GP1	GP0
		15:8	COUNTSYNC							
		7:0		COMP1	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	COMP1	31:24	COMP[31:24]							
		23:16	COMP[23:16]							
		15:8	COMP[15:8]							
		7:0	COMP[7:0]							
0x28 ... 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	GP2	31:24	GP[31:24]							
		23:16	GP[23:16]							
		15:8	GP[15:8]							
		7:0	GP[7:0]							
0x4C	GP3	31:24	GP[31:24]							
		23:16	GP[23:16]							
		15:8	GP[15:8]							
		7:0	GP[7:0]							

Register Summary - 32-bit Counter Mode (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x50 ... 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]	
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, Write-Synchronized

Table 25-4. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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.
This bit is not enable-protected.

Value	Description
0	COUNT read synchronization is disabled
1	COUNT read synchronization is enabled

Bit 14 – GPTRST GP Registers Reset On Tamper Enable
Only GP registers enabled by the CTRLB.GPnEN bits are affected. This bit can be written only when the peripheral is disabled.
This bit is not synchronized.
Note: Note: If CTRA.GPRST is set, the user is required to read the GPx register multiple times until same value is confirmed for two consecutive reads. Otherwise, discard the false GPx register value.

Bits 11:8 – PRESCALER[3:0] Prescaler
These bits define the prescaling factor for the RTC clock source (GCLK_RTC) to generate the counter clock (CLK_RTC_CNT). Periodic events and interrupts are not available when the prescaler is off.
These bits are not synchronized.

Value	Name	Description
0x0	OFF	CLK_RTC_CNT = GCLK_RTC/1
0x1	DIV1	CLK_RTC_CNT = GCLK_RTC/1
0x2	DIV2	CLK_RTC_CNT = GCLK_RTC/2
0x3	DIV4	CLK_RTC_CNT = GCLK_RTC/4
0x4	DIV8	CLK_RTC_CNT = GCLK_RTC/8
0x5	DIV16	CLK_RTC_CNT = GCLK_RTC/16
0x6	DIV32	CLK_RTC_CNT = GCLK_RTC/32
0x7	DIV64	CLK_RTC_CNT = GCLK_RTC/64
0x8	DIV128	CLK_RTC_CNT = GCLK_RTC/128

Value	Name	Description
0x9	DIV256	CLK_RTC_CNT = GCLK_RTC/256
0xA	DIV512	CLK_RTC_CNT = GCLK_RTC/512
0xB	DIV1024	CLK_RTC_CNT = GCLK_RTC/1024
0xC-0xF	-	Reserved

Bit 7 – MATCHCLR Clear on Match

This bit defines if the counter is cleared or not on a match.
This bit is not synchronized.

Value	Description
0	The counter is not cleared on a Compare/Alarm 0 match
1	The counter is cleared on a Compare/Alarm 0 match

Bits 3:2 – MODE[1:0] Operating Mode

This bit group defines the operating mode of the RTC.
This bit is not synchronized.

Value	Name	Description
0x0	COUNT32	Mode 0: 32-bit counter
0x1	COUNT16	Mode 1: 16-bit counter
0x2	CLOCK	Mode 2: Clock/calendar
0x3	-	Reserved

Bit 1 – ENABLE Enable

Due to synchronization there is a delay between writing CTRLA.ENABLE and until the peripheral is enabled/disabled. The value written to CTRLA.ENABLE will read back immediately and the Enable bit in the Synchronization The Busy register (SYNCBUSY.ENABLE) will be set. SYNCBUSY.ENABLE will be cleared when the operation is complete.
This bit is not enable-protected.

Value	Description
0	The peripheral is disabled
1	The peripheral is enabled

Bit 0 – SWRST Software Reset

Writing a ‘0’ to this bit has no effect.
Writing a '1' to this bit resets all registers in the RTC (except DBGCTRL) to their initial state, and the RTC will be disabled.
Writing a '1' to CTRLA.SWRST will always take precedence, meaning that all other writes in the same write-operation will be discarded.
Due to synchronization there is a delay between writing CTRLA.SWRST and until the reset is complete. CTRLA.SWRST will be cleared when the reset is complete.
This bit is not enable-protected.

Notes:

1. When the CTRLA.SWRST is written, the user must poll the SYNCB.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 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

Table 25-5. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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			GP2EN	GP0EN
Access	R/W	R/W	R/W	R/W			R/W	R/W
Reset	0	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 Layer Frequency
These bits define the prescaling factor for the RTC clock output (OUT) used during active layer protection in terms of the CLK_RTC.

Value	Name	Description
0x0	DIV2	CLK_RTC_OUT = CLK_RTC / 2
0x1	DIV4	CLK_RTC_OUT = CLK_RTC / 4
0x2	DIV8	CLK_RTC_OUT = CLK_RTC / 8
0x3	DIV16	CLK_RTC_OUT = CLK_RTC / 16
0x4	DIV32	CLK_RTC_OUT = CLK_RTC / 32
0x5	DIV64	CLK_RTC_OUT = CLK_RTC / 64
0x6	DIV128	CLK_RTC_OUT = CLK_RTC / 128
0x7	DIV256	CLK_RTC_OUT = CLK_RTC / 256

Bits 10:8 – DEBF[2:0] Debounce Frequency
These bits define the prescaling factor for the input debouncers in terms of the CLK_RTC.

Value	Name	Description
0x0	DIV2	CLK_RTC_DEB = CLK_RTC / 2
0x1	DIV4	CLK_RTC_DEB = CLK_RTC / 4
0x2	DIV8	CLK_RTC_DEB = CLK_RTC / 8
0x3	DIV16	CLK_RTC_DEB = CLK_RTC / 16
0x4	DIV32	CLK_RTC_DEB = CLK_RTC / 32
0x5	DIV64	CLK_RTC_DEB = CLK_RTC / 64
0x6	DIV128	CLK_RTC_DEB = CLK_RTC / 128
0x7	DIV256	CLK_RTC_DEB = CLK_RTC / 256

Bit 7 – DMAEN DMA Enable

The RTC can trigger a DMA request when the timestamp is ready in the TIMESTAMP register.

Value	Description
0	Tamper DMA request is disabled. Reading TIMESTAMP has no effect on INTFLAG.TAMPER.
1	Tamper DMA request is enabled. Reading TIMESTAMP will clear INTFLAG.TAMPER.

Bit 6 – RTCOUT RTC Output Enable

Value	Description
0	The RTC active layer output is disabled.
1	The RTC active layer output is enabled.

Bit 5 – DEBASYNC Debouncer Asynchronous Enable

Value	Description
0	The tamper input debouncers operate synchronously.
1	The tamper input debouncers operate asynchronously.

Bit 4 – DEBMAJ Debouncer Majority Enable

Value	Description
0	The tamper input debouncers match three equal values.
1	The tamper input debouncers match majority two of three values.

Bit 1 – GP2EN General Purpose 2 Enable

Value	Description
0	COMP1 compare function enabled. GP2/GP3 disabled.
1	COMP1 compare function disabled. GP2/GP3 enabled.

Bit 0 – GP0EN General Purpose 0 Enable

Value	Description
0	COMP0/1 compare function enabled. GP0/GP1 disabled.
1	COMP0/1 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

Table 25-6. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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					CMPEOn[1:0]	
Access	R/W	R/W					R/W	R/W
Reset	0	0					0	0

Bit	7	6	5	4	3	2	1	0
	PEREO7	PEREO6	PEREO5	PEREO4	PEREO3	PEREO2	PEREO1	PEREO0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 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 9:8 – CMPEOn[1:0] Compare n Event Output Enable [n = 1..0]

Value	Description
0	Compare n event is disabled and will not be generated.
1	Compare n event is enabled and will be generated for every compare match.

Bits 0, 1, 2, 3, 4, 5, 6, 7 – PEREOn Periodic Interval n Event Output Enable [n = 7..0]

Note: The PEREO5, PEREO6, and PEREO7 bits are not available on 100-pin packages (RTC_IN[5:7] and RTC_OUT[5:7]).

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.

Table 25-7. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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 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.

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.

Note: The PER5, PER6, and PER7 bits are not available on 100-pin packages (RTC_IN[5:7] and RTC_OUT[5:7]).

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.

Table 25-8. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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 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.
Note: The PER5, PER6, and PER7 bits are not available on 100-pin packages (RTC_IN[5:7] and RTC_OUT[5:7]).

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: -

Note: Interrupt flags must be cleared and then read back to confirm the clear before exiting the ISR to avoid double interrupts.

Table 25-9. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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 INTENCLR/SET.OVF is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Overflow interrupt flag.

Bit 14 – TAMPER Tamper event
This flag is set after a tamper condition occurs, and an interrupt request will be generated if INTENCLR.TAMPER/INTENSET.TAMPER is '1'. Writing a '0' to this bit has no effect. Writing a '1' to this bit clears the Tamper interrupt flag.

Bits 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 INTENCLR.CMPn or INTENSET.CMPn is one.
Writing a ‘0’ to this bit has no effect.
Writing a '1' to this bit clears the Compare n Interrupt flag.

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 [n = 7..0]
This flag is cleared by writing a '1' to the flag.
This flag is set on the 0-to-1 transition of prescaler bit [n+2], and an interrupt request will be generated if INTENCLR/SET.PERn is one.
Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the Periodic Interval n interrupt flag.
Note: The PER5, PER6, and PER7 bits are not available on 100-pin packages (RTC_IN[5:7] and RTC_OUT[5:7]).

25.7.7. Debug Control

Name: DBGCTRL
Offset: 0x0E
Reset: 0x00
Property: PAC Write-Protection

Table 25-10. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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: -

Table 25-11. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					GP3	GP2	GP1	GP0
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
	COUNTSYNC							
Access	R							
Reset	0							

Bit	7	6	5	4	3	2	1	0
		COMP1	COMP0		COUNT	FREQCORR	ENABLE	SWRST
Access		R/W	R/W		R	R	R	R
Reset		0	0		0	0	0	0

Bits 16, 17, 18, 19 – GPn General Purpose n Synchronization Busy Status [n = 3..0]

Value	Description
0	Write synchronization for the GPn register is complete.
1	Write synchronization for the GPn register is ongoing.

Bit 15 – COUNTSYNC Count Read Sync Enable Synchronization Busy Status

Value	Description
0	Write synchronization for the CTRLA.COUNTSYNC bit is complete.
1	Write synchronization for the CTRLA.COUNTSYNC bit is ongoing.

Bits 5, 6 – COMPn Compare n Synchronization Busy Status [n = 1..0]

Value	Description
0	Write synchronization for the COMPn register is complete.
1	Write synchronization for the COMPn register is ongoing.

Bit 3 – COUNT Count Value Synchronization Busy Status

Value	Description
0	Read/write synchronization for the COUNT register is complete.

Value	Description
1	Read/write synchronization for the COUNT register is ongoing.

Bit 2 – FREQCORR Frequency Correction Synchronization Busy Status

Value	Description
0	Write synchronization for the FREQCORR register is complete.
1	Write synchronization for the FREQCORR register is ongoing.

Bit 1 – ENABLE Enable Synchronization Busy Status

Value	Description
0	Write synchronization for the CTRLA.ENABLE bit is complete.
1	Write synchronization for the CTRLA.ENABLE bit is ongoing.

Bit 0 – SWRST Software Reset Synchronization Busy Status

Value	Description
0	Write synchronization for the CTRLA.SWRST bit is complete.
1	Write synchronization for the CTRLA.SWRST bit is ongoing.

25.7.9. Frequency Correction

Name: FREQCORR
Offset: 0x14
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized

Table 25-12. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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

Bit 7 – SIGN Correction Sign

Value	Description
0	The correction value is positive, i.e., frequency will be decreased.
1	The correction value is negative, i.e., frequency will be increased.

Bits 6:0 – VALUE[6:0] Correction Value
These bits define the amount of correction applied to the RTC prescaler.

Value	Description
0	Correction is disabled and the RTC frequency is unchanged.
1 - 127	The RTC frequency is adjusted according to the value.

25.7.10. Counter Value in COUNT32 mode (CTRLA.MODE=0)

Name: COUNT
Offset: 0x18
Reset: 0x00000000
Property: PAC Write-Protection, Write-Synchronized, Read-Synchronized

Table 25-13. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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 0 Value in COUNT32 mode (CTRLA.MODE=0)

Name: COMPn
Offset: 0x20 + n*0x04 [n=0..1]
Reset: 0x00000000
Property: PAC Write-Protection, Write-Synchronized

Table 25-14. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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 COMP0 is continuously compared with the 32-bit COUNT value. When a match occurs, the Compare 0 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: GPn
Offset: 0x40 + n*0x04 [n=0..3]
Reset: 0x00000000
Property: PAC Write-Protection, Write-Synchronized

Table 25-15. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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

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

Bit	15	14	13	12	11	10	9	8
	GP[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

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

Bits 31:0 – GP[31:0] General Purpose
These bits are for user-defined general purpose use, see *General Purpose Registers*.

25.7.13. Tamper Control

Name: TAMPCTRL
Offset: 0x60
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Table 25-16. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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 Layer Protection mode (TAMPCTRL.INACT = ACTL).

Note: The DEBNC5, DEBNC6, and DEBNC7 bits are not available on 100-pin packages (RTC_IN[5:7] and RTC_OUT[5:7]).

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 Layer Protection mode (TAMPCTRL.INACT = ACTL).

Note: The TAMLVL5, TAMLVL6, and TAMLVL7 bits are not available on 100-pin packages (RTC_IN[5:7] and RTC_OUT[5:7]).

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.

Note: The IN5ACT, IN6ACT, and IN7ACT bits are not available on 100-pin packages (RTC_IN[5:7] and RTC_OUT[5:7]).

Value	Name	Description
0x0	OFF	Off (Disabled)
0x1	WAKE	Wake and set Tamper flag
0x2	CAPTURE	Capture timestamp and set Tamper flag
0x3	ACTL	Compare RTC signal routed between INn and OUT pins. When a mismatch occurs, capture timestamp and set Tamper flag

25.7.14. Timestamp

Name: TIMESTAMP
Offset: 0x64
Reset: 0x0
Property: -

Table 25-17. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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

Important: The DMA reads the TAMPID register with an outdated value when triggered through the EVSYS. A secondary read of the TAMPID register is required to get accurate data.

Table 25-18. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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 on 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

Table 25-19. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
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 Layer Internal Select n [n=0..7]

Note: Only one ALSI bit must be set to enable Active Layer Protection on the TrustRAM.

Note: The ALSI5, ALSI6, and ALSI7 bits are not available on 100 pin packages (RTC_IN[5:7] and RTC_OUT[5:7]).

Value	Description
0	Active layer Protection is monitoring the RTC signal using INn and OUTn tamper pins
1	Active layer Protection is monitoring the RTC signal on the TrustRAM shield

25.8. Register Summary - 16-bit Counter Mode

For descriptions and definitions of both Register and bitfield properties, refer to Register Properties.

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			GP2EN	GP0EN
0x04	EVCTRL	31:24								PERDEO
		23:16								TAMPEVEI
		15:8	OVFEO	TAMPEREO			CMPEO3	CMPEO2	CMPEO1	CMPEO0
		7:0	PEREO7	PEREO6	PEREO5	PEREO4	PEREO3	PEREO2	PEREO1	PEREO0
0x08	INTENCLR	15:8	OVF	TAMPER			CMP3	CMP2	CMP1	CMP0
		7:0	PER7	PER6	PER5	PER4	PER3	PER2	PER1	PER0
0x0A	INTENSET	15:8	OVF	TAMPER			CMP3	CMP2	CMP1	CMP0
		7:0	PER7	PER6	PER5	PER4	PER3	PER2	PER1	PER0
0x0C	INTFLAG	15:8	OVF	TAMPER			CMP3	CMP2	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					GP3	GP2	GP1	GP0
		15:8	COUNTSYNC							COMP3
		7:0	COMP2	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	COMP2	15:8	COMP[15:8]							
		7:0	COMP[7:0]							
0x26	COMP3	15:8	COMP[15:8]							
		7:0	COMP[7:0]							
0x28 ... 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	GP2	31:24	GP[31:24]							
		23:16	GP[23:16]							
		15:8	GP[15:8]							
		7:0	GP[7:0]							

Register Summary - 16-bit Counter Mode (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x4C	GP3	31:24	GP[31:24]							
		23:16	GP[23:16]							
		15:8	GP[15:8]							
		7:0	GP[7:0]							
0x50 ... 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]	
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, Write-Synchronized

Important: When the RTC is disabled (writing CTRLA.ENABLE=0), the SYNCBUSY.ENABLE will be set to '0' before the TAMPER detection is disabled. Changing the tamper configuration (TAMCTRL,TAMPCTRLB,CTRLB,EVCTRL) during that time can produce a false tamper detection. After the fall of SYNCBUSY.ENABLE, the firmware must wait for at least 1 RTC clock period before changing the tamper configuration.

Table 25-20. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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.
This bit is not enable-protected.

Value	Description
0	COUNT read synchronization is disabled
1	COUNT read synchronization is enabled

Bit 14 – GPTRST GP Registers Reset On Tamper Enable
Only GP registers enabled by the CTRLB.GPnEN bits are affected. This bit can be written only when the peripheral is disabled.
This bit is not synchronized.
Note: :If CTRA.GPRST is set, the user is required to read the GPx register multiple times until same value is confirmed for two consecutive reads. Otherwise, discard the false GPx register value.

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.
These bits are not synchronized.

Value	Name	Description
0x0	OFF	CLK_RTC_CNT = GCLK_RTC/1
0x1	DIV1	CLK_RTC_CNT = GCLK_RTC/1
0x2	DIV2	CLK_RTC_CNT = GCLK_RTC/2
0x3	DIV4	CLK_RTC_CNT = GCLK_RTC/4
0x4	DIV8	CLK_RTC_CNT = GCLK_RTC/8
0x5	DIV16	CLK_RTC_CNT = GCLK_RTC/16
0x6	DIV32	CLK_RTC_CNT = GCLK_RTC/32
0x7	DIV64	CLK_RTC_CNT = GCLK_RTC/64
0x8	DIV128	CLK_RTC_CNT = GCLK_RTC/128
0x9	DIV256	CLK_RTC_CNT = GCLK_RTC/256
0xA	DIV512	CLK_RTC_CNT = GCLK_RTC/512
0xB	DIV1024	CLK_RTC_CNT = GCLK_RTC/1024
0xC-0xF	-	Reserved

Bits 3:2 – MODE[1:0] Operating Mode

This field defines the operating mode of the RTC. This bit is not synchronized.

Value	Name	Description
0x0	COUNT32	Mode 0: 32-bit counter
0x1	COUNT16	Mode 1: 16-bit counter
0x2	CLOCK	Mode 2: Clock/calendar
0x3	-	Reserved

Bit 1 – ENABLE Enable

Due to synchronization there is delay from writing CTRLA.ENABLE until the peripheral is enabled/disabled. The value written to CTRLA.ENABLE will read back immediately and the Enable bit in the Synchronization Busy register (SYNCBUSY.ENABLE) will be set. SYNCBUSY.ENABLE will be cleared when the operation is complete.
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.
Due to synchronization there is a delay from writing CTRLA.SWRST until the reset is complete. CTRLA.SWRST will be cleared when the reset is complete.
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

Table 25-21. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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			GP2EN	GP0EN
Access	R/W	R/W	R/W	R/W			R/W	R/W
Reset	0	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 Layer Frequency

These bits define the prescaling factor for the RTC clock output (OUT) used during active layer protection in terms of the CLK_RTC.

Value	Name	Description
0x0	DIV2	CLK_RTC_OUT = CLK_RTC / 2
0x1	DIV4	CLK_RTC_OUT = CLK_RTC / 4
0x2	DIV8	CLK_RTC_OUT = CLK_RTC / 8
0x3	DIV16	CLK_RTC_OUT = CLK_RTC / 16
0x4	DIV32	CLK_RTC_OUT = CLK_RTC / 32
0x5	DIV64	CLK_RTC_OUT = CLK_RTC / 64
0x6	DIV128	CLK_RTC_OUT = CLK_RTC / 128
0x7	DIV256	CLK_RTC_OUT = CLK_RTC / 256

Bits 10:8 – DEBF[2:0] Debounce Frequency

These bits define the prescaling factor for the input debouncers in terms of the CLK_RTC.

Value	Name	Description
0x0	DIV2	CLK_RTC_DEB = CLK_RTC / 2
0x1	DIV4	CLK_RTC_DEB = CLK_RTC / 4
0x2	DIV8	CLK_RTC_DEB = CLK_RTC / 8
0x3	DIV16	CLK_RTC_DEB = CLK_RTC / 16
0x4	DIV32	CLK_RTC_DEB = CLK_RTC / 32
0x5	DIV64	CLK_RTC_DEB = CLK_RTC / 64
0x6	DIV128	CLK_RTC_DEB = CLK_RTC / 128
0x7	DIV256	CLK_RTC_DEB = CLK_RTC / 256

Bit 7 – DMAEN DMA Enable

The RTC can trigger a DMA request when the timestamp is ready in the TIMESTAMP register.

Value	Description
0	Tamper DMA request is disabled. Reading TIMESTAMP has no effect on INTFLAG.TAMPER.
1	Tamper DMA request is enabled. Reading TIMESTAMP will clear INTFLAG.TAMPER.

Bit 6 – RTCOUT RTC Output Enable

Value	Description
0	The RTC active layer output is disabled.
1	The RTC active layer output is enabled.

Bit 5 – DEBASYNC Debouncer Asynchronous Enable

Value	Description
0	The tamper input debouncers operate synchronously.
1	The tamper input debouncers operate asynchronously.

Bit 4 – DEBMAJ Debouncer Majority Enable

Value	Description
0	The tamper input debouncers match three equal values.
1	The tamper input debouncers match majority two of three values.

Bit 1 – GP2EN General Purpose 2 Enable

Value	Description
0	COMP1 compare function enabled. GP2/GP3 disabled.
1	COMP1 compare function disabled. GP2/GP3 enabled.

Bit 0 – GP0EN General Purpose 0 Enable

Value	Description
0	COMP0/1 compare function enabled. GP0/GP1 disabled.
1	COMP0/1 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

Table 25-22. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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			CMPEO3	CMPEO2	CMPEO1	CMPEO0
Access	R/W	R/W			R/W	R/W	R/W	R/W
Reset	0	0			0	0	0	0

Bit	7	6	5	4	3	2	1	0
	PEREO7	PEREO6	PEREO5	PEREO4	PEREO3	PEREO2	PEREO1	PEREO0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 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, 10, 11 – CMPEOn Compare n Event Output Enable [n = 3..0]

Value	Description
0	Compare n event is disabled and will not be generated.
1	Compare n event is enabled and will be generated for every compare match.

Bits 0, 1, 2, 3, 4, 5, 6, 7 – PEREOn Periodic Interval n Event Output Enable [n = 7..0]

Note: The PEREO5, PEREO6, and PEREO7 bits are not available on 100 pin packages (RTC_IN[5:7] and RTC_OUT[5:7]).

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.

Table 25-23. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	15	14	13	12	11	10	9	8
	OVF	TAMPER			CMP3	CMP2	CMP1	CMP0
Access	R/W	R/W			R/W	R/W	R/W	R/W
Reset	0	0			0	0	0	0

Bit	7	6	5	4	3	2	1	0
	PER7	PER6	PER5	PER4	PER3	PER2	PER1	PER0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 15 – OVF Overflow Interrupt Enable
Writing a '0' to this bit has no effect. Writing a '1' to this bit will clear the Overflow Interrupt Enable bit, which disables the Overflow interrupt.

Value	Description
0	The Overflow interrupt is disabled.
1	The Overflow interrupt is enabled.

Bit 14 – TAMPER Tamper Interrupt Enable
Writing a '0' to this bit has no effect. Writing a '1' to this bit will clear the Tamper Interrupt Enable bit, which disables the Tamper interrupt.

Value	Description
0	The Tamper interrupt is disabled.
1	The Tamper interrupt is enabled.

Bits 8, 9, 10, 11 – CMPn Compare n Interrupt Enable [n = 3..0]
Writing a '0' to this bit has no effect. Writing a '1' to this bit will clear the Compare n Interrupt Enable bit, which disables the Compare n interrupt.

Value	Description
0	The Compare n interrupt is disabled.
1	The Compare n interrupt is enabled.

Bits 0, 1, 2, 3, 4, 5, 6, 7 – PERn Periodic Interval n Interrupt Enable [n = 7..0]
Writing a '0' to this bit has no effect. Writing a '1' to this bit will clear the Periodic Interval n Interrupt Enable bit, which disables the Periodic Interval n interrupt.
Note: The PER5, PER6, and PER7 bits are not available on 100 pin packages (RTC_IN[5:7] and RTC_OUT[5:7]).

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.

Table 25-24. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	15	14	13	12	11	10	9	8
	OVF	TAMPER			CMP3	CMP2	CMP1	CMP0
Access	R/W	R/W			R/W	R/W	R/W	R/W
Reset	0	0			0	0	0	0

Bit	7	6	5	4	3	2	1	0
	PER7	PER6	PER5	PER4	PER3	PER2	PER1	PER0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 15 – OVF Overflow Interrupt Enable
Writing a '0' to this bit has no effect. Writing a '1' to this bit will set the Overflow Interrupt Enable bit, which enables the Overflow interrupt.

Value	Description
0	The Overflow interrupt is disabled.
1	The Overflow interrupt is enabled.

Bit 14 – TAMPER Tamper Interrupt Enable
Writing a '0' to this bit has no effect. Writing a '1' to this bit will set the Tamper Interrupt Enable bit, which enables the Tamper interrupt.

Value	Description
0	The Tamper interrupt is disabled.
1	The Tamper interrupt is enabled.

Bits 8, 9, 10, 11 – CMPn Compare n Interrupt Enable [n = 3..0]
Writing a '0' to this bit has no effect. Writing a '1' to this bit will set the Compare n Interrupt Enable bit, which 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.
Note: The PER5, PER6, and PER7 bits are not available on 100 pin packages (RTC_IN[5:7] and RTC_OUT[5:7]).

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: -

Note: Interrupt flags must be cleared and then read back to confirm the clear before exiting the ISR to avoid double interrupts.

Table 25-25. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	15	14	13	12	11	10	9	8
	OVF	TAMPER			CMP3	CMP2	CMP1	CMP0
Access	R/W	R/W			R/W	R/W	R/W	R/W
Reset	0	0			0	0	0	0

Bit	7	6	5	4	3	2	1	0
	PER7	PER6	PER5	PER4	PER3	PER2	PER1	PER0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 15 – OVF Overflow
This flag is cleared by writing a '1' to the flag.
This flag is set on the next CLK_RTC_CNT cycle after an overflow condition occurs, and an interrupt request will be generated if INTENCLR/SET.OVF is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Overflow interrupt flag.

Bit 14 – TAMPER Tamper
This flag is set after a tamper condition occurs, and an interrupt request will be generated if INTENCLR.TAMPER/ 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, 10, 11 – CMPn Compare n [n = 3..0]
This flag is cleared by writing a '1' to the flag.
This flag is set on the next CLK_RTC_CNT cycle after a match with the compare condition, and an interrupt request will be generated if INTENCLR/SET.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 INTENCLR/SET.PERn is one.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Periodic Interval n interrupt flag.
Note: The PER5, PER6, and PER7 bits are not available on 100 pin packages (RTC_IN[5:7] and RTC_OUT[5:7]).

25.8.7. Debug Control

Name: DBGCTRL
Offset: 0x0E
Reset: 0x00
Property: PAC Write-Protection

Table 25-26. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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: -

Table 25-27. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					GP3	GP2	GP1	GP0
Access					R	R	R	R
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
	COUNTSYNC							COMP3
Access	R							R/W
Reset	0							0

Bit	7	6	5	4	3	2	1	0
	COMP2	COMP1	COMP0	PER	COUNT	FREQCORR	ENABLE	SWRST
Access	R/W	R/W	R/W	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 16, 17, 18, 19 – GPn General Purpose n Synchronization Busy Status [n = 3..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, 7, 8 – COMPn Compare n Synchronization Busy Status [n = 3..0]

Value	Description
0	Write synchronization for COMPn register is complete.
1	Write synchronization for COMPn register is ongoing.

Bit 4 – PER Period Synchronization Busy Status

Value	Description
0	Write synchronization for PER register is complete.

Value	Description
1	Write synchronization for PER register is ongoing.

Bit 3 – COUNT Count Value Synchronization Busy Status

Value	Description
0	Read/write synchronization for COUNT register is complete.
1	Read/write synchronization for COUNT register is ongoing.

Bit 2 – FREQCORR Frequency Correction Synchronization Busy Status

Value	Description
0	Write synchronization for FREQCORR register is complete.
1	Write synchronization for FREQCORR register is ongoing.

Bit 1 – ENABLE Enable Synchronization Busy Status

Value	Description
0	Write synchronization for CTRLA.ENABLE bit is complete.
1	Write synchronization for CTRLA.ENABLE bit is ongoing.

Bit 0 – SWRST Software Reset Synchronization Busy Status

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

Table 25-28. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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

Bit 7 – SIGN Correction Sign

Value	Description
0	The correction value is positive, i.e., frequency will be decreased.
1	The correction value is negative, i.e., frequency will be increased.

Bits 6:0 – VALUE[6:0] Correction Value
These bits define the amount of correction applied to the RTC prescaler.

Value	Description
0	Correction is disabled and the RTC frequency is unchanged.
1 - 127	The RTC frequency is adjusted according to the value.

25.8.10. Counter Value in COUNT16 mode (CTRLA.MODE=1)

Name: COUNT
Offset: 0x18
Reset: 0x0000
Property: PAC Write-Protection, Write-Synchronized, Read-Synchronized

Table 25-29. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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

Table 25-30. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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: COMPn
Offset: 0x20 + n*0x02 [n=0..3]
Reset: 0x0000
Property: PAC Write-Protection, Write-Synchronized

Table 25-31. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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: GPn
Offset: 0x40 + n*0x04 [n=0..3]
Reset: 0x00000000
Property: PAC Write-Protection, Write-Synchronized

Table 25-32. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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

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

Bit	15	14	13	12	11	10	9	8
	GP[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

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

Bits 31:0 – GP[31:0] General Purpose
These bits are for user-defined general purpose use, see *General Purpose Registers*.

25.8.14. Tamper Control

Name: TAMPCTRL
Offset: 0x60
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Table 25-33. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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 Layer Protection mode (TAMPCTRL.INACT = ACTL).

Value	Description
0	Debouncing is disabled for Tamper input INn
1	Debouncing is enabled for Tamper input INn

Bits 16, 17, 18, 19, 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 Layer Protection mode (TAMPCTRL.INACT = ACTL).

Value	Description
0	A falling edge condition will be detected on Tamper input INn.
1	A rising edge condition will be detected on Tamper input INn.

Bits 0:1, 2:3, 4:5, 6:7, 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

Value	Name	Description
0x2	CAPTURE	Capture timestamp and set Tamper flag
0x3	ACTL	Compare RTC signal routed between INn and OUT pins . When a mismatch occurs, capture timestamp and set Tamper flag

25.8.15. Timestamp

Name: TIMESTAMP
Offset: 0x64
Reset: 0x0000
Property: -

Table 25-34. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
	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

Table 25-35. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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 on 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

Table 25-36. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
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 Layer Internal Select n [n=0..7]

Note: Only one ALSI bit must be set to enable Active Layer Protection on the TrustRAM.

Value	Description
0	Active layer Protection is monitoring the RTC signal using INn and OUTn tamper pins
1	Active layer Protection is monitoring the RTC signal on the TrustRAM shield

25.9. Register Summary Clock/Calendar Mode

For descriptions and definitions of both Register and bitfield properties, refer to Register Properties.

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			GP2EN	GP0EN
0x04	EVCTRL	31:24								PERDEO
		23:16								TAMPEVEI
		15:8	OVFEO	TAMPEREO					ALARMEO1	ALARMEO0
		7:0	PEREO7	PEREO6	PEREO5	PEREO4	PEREO3	PEREO2	PEREO1	PEREO0
0x08	INTENCLR	15:8	OVF	TAMPER					ALARM1	ALARM0
		7:0	PER7	PER6	PER5	PER4	PER3	PER2	PER1	PER0
0x0A	INTENSET	15:8	OVF	TAMPER					ALARM1	ALARM0
		7:0	PER7	PER6	PER5	PER4	PER3	PER2	PER1	PER0
0x0C	INTFLAG	15:8	OVF	TAMPER					ALARM1	ALARM0
		7:0	PER7	PER6	PER5	PER4	PER3	PER2	PER1	PER0
0x0E	DBGCTRL	7:0								DBGRUN
0x0F	Reserved									
0x10	SYNCBUSY	31:24								
		23:16					GP3	GP2	GP1	GP0
		15:8	CLOCKSYNC			MASK1	MASK0			
		7:0		ALARM1	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	MASK	7:0						SEL[2:0]		
0x25 ... 0x27	Reserved									
0x28	ALARM1	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]					
0x2C ... 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	GP2	31:24	GP[31:24]							
		23:16	GP[23:16]							
		15:8	GP[15:8]							
		7:0	GP[7:0]							

Register Summary Clock/Calendar Mode (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x4C	GP3	31:24	GP[31:24]							
		23:16	GP[23:16]							
		15:8	GP[15:8]							
		7:0	GP[7:0]							
0x50 ... 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]	
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, Write-Synchronized

Table 25-37. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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.
This bit is not enable-protected.

Value	Description
0	CLOCK read synchronization is disabled
1	CLOCK read synchronization is enabled

Bit 14 – GPTRST GP Registers Reset On Tamper Enable

Only GP registers enabled by the CTRLB.GPnEN bits are affected. This bit can be written only when the peripheral is disabled.
This bit is no synchronized.
Note: If CTRA.GPRST is set, the user is required to read the GPx register multiple times until same value is confirmed for two consecutive reads. Otherwise, discard the false GPx register value.

Bits 11:8 – PRESCALER[3:0] Prescaler

These bits define the prescaling factor for the RTC clock source (GCLK_RTC) to generate the counter clock (CLK_RTC_CNT). Periodic events and interrupts are not available when the prescaler is off.
These bits are not synchronized.

Value	Name	Description
0x0	OFF	CLK_RTC_CNT = GCLK_RTC/1
0x1	DIV1	CLK_RTC_CNT = GCLK_RTC/1
0x2	DIV2	CLK_RTC_CNT = GCLK_RTC/2
0x3	DIV4	CLK_RTC_CNT = GCLK_RTC/4
0x4	DIV8	CLK_RTC_CNT = GCLK_RTC/8
0x5	DIV16	CLK_RTC_CNT = GCLK_RTC/16
0x6	DIV32	CLK_RTC_CNT = GCLK_RTC/32
0x7	DIV64	CLK_RTC_CNT = GCLK_RTC/64
0x8	DIV128	CLK_RTC_CNT = GCLK_RTC/128

Value	Name	Description
0x9	DIV256	CLK_RTC_CNT = GCLK_RTC/256
0xA	DIV512	CLK_RTC_CNT = GCLK_RTC/512
0xB	DIV1024	CLK_RTC_CNT = GCLK_RTC/1024
0xC-0xF	-	Reserved

Bit 7 – MATCHCLR Clear on Match

This bit is valid only in Mode 0 (COUNT32) and Mode 2 (CLOCK). This bit can be written only when the peripheral is disabled. This bit is not synchronized.

Value	Description
0	The counter is not cleared on a Compare/Alarm 0 match
1	The counter is cleared on a Compare/Alarm 0 match

Bit 6 – CLKREP Clock Representation

This bit is valid only in Mode 2 and determines how the hours are represented in the Clock Value (CLOCK) register. This bit can be written only when the peripheral is disabled. This bit is not synchronized.

Value	Description
0	24 Hour
1	12 Hour (AM/PM)

Bits 3:2 – MODE[1:0] Operating Mode

This field defines the operating mode of the RTC. This bit is not synchronized.

Value	Name	Description
0x0	COUNT32	Mode 0: 32-bit counter
0x1	COUNT16	Mode 1: 16-bit counter
0x2	CLOCK	Mode 2: Clock/calendar
0x3	-	Reserved

Bit 1 – ENABLE Enable

Due to synchronization there is delay from writing CTRLA.ENABLE until the peripheral is enabled/disabled. The value written to CTRLA.ENABLE will read back immediately and the Enable bit in the Synchronization Busy register (SYNCBUSY.ENABLE) will be set. SYNCBUSY.ENABLE will be cleared when the operation is complete.
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.
Due to synchronization there is a delay from writing CTRLA.SWRST until the reset is complete. CTRLA.SWRST will be cleared when the reset is complete.
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

Table 25-38. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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			GP2EN	GP0EN
Access	R/W	R/W	R/W	R/W			R/W	R/W
Reset	0	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 Layer Frequency
These bits define the prescaling factor for the RTC clock output (OUT) used during active layer protection in terms of the CLK_RTC.

Value	Name	Description
0x0	DIV2	CLK_RTC_OUT = CLK_RTC / 2
0x1	DIV4	CLK_RTC_OUT = CLK_RTC / 4
0x2	DIV8	CLK_RTC_OUT = CLK_RTC / 8
0x3	DIV16	CLK_RTC_OUT = CLK_RTC / 16
0x4	DIV32	CLK_RTC_OUT = CLK_RTC / 32
0x5	DIV64	CLK_RTC_OUT = CLK_RTC / 64
0x6	DIV128	CLK_RTC_OUT = CLK_RTC / 128
0x7	DIV256	CLK_RTC_OUT = CLK_RTC / 256

Bits 10:8 – DEBF[2:0] Debounce Frequency
These bits define the prescaling factor for the input debouncers in terms of the CLK_RTC.

Value	Name	Description
0x0	DIV2	CLK_RTC_DEB = CLK_RTC / 2
0x1	DIV4	CLK_RTC_DEB = CLK_RTC / 4
0x2	DIV8	CLK_RTC_DEB = CLK_RTC / 8
0x3	DIV16	CLK_RTC_DEB = CLK_RTC / 16
0x4	DIV32	CLK_RTC_DEB = CLK_RTC / 32
0x5	DIV64	CLK_RTC_DEB = CLK_RTC / 64
0x6	DIV128	CLK_RTC_DEB = CLK_RTC / 128
0x7	DIV256	CLK_RTC_DEB = CLK_RTC / 256

Bit 7 – DMAEN DMA Enable

The RTC can trigger a DMA request when the timestamp is ready in the TIMESTAMP register.

Value	Description
0	Tamper DMA request is disabled. Reading TIMESTAMP has no effect on INTFLAG.TAMPER.
1	Tamper DMA request is enabled. Reading TIMESTAMP will clear INTFLAG.TAMPER.

Bit 6 – RTCOUT RTC Out Enable

Value	Description
0	The RTC active layer output is disabled.
1	The RTC active layer output is enabled.

Bit 5 – DEBASYNC Debouncer Asynchronous Enable

Value	Description
0	The tamper input debouncers operate synchronously.
1	The tamper input debouncers operate asynchronously.

Bit 4 – DEBMAJ Debouncer Majority Enable

Value	Description
0	The tamper input debouncers match three equal values.
1	The tamper input debouncers match majority two of three values.

Bit 1 – GP2EN General Purpose 2 Enable

Value	Description
0	COMP1 compare function enabled. GP2/GP3 disabled.
1	COMP1 compare function disabled. GP2/GP3 enabled.

Bit 0 – GP0EN General Purpose 0 Enable

Value	Description
0	COMP0/1 compare function enabled. GP0/GP1 disabled.
1	COMP0/1 compare function disabled. GP0/GP1 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

Table 25-39. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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					ALARMEO1	ALARMEO0
Access	R/W	R/W					R/W	R/W
Reset	0	0					0	0

Bit	7	6	5	4	3	2	1	0
	PEREO7	PEREO6	PEREO5	PEREO4	PEREO3	PEREO2	PEREO1	PEREO0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 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.

Bits 8, 9 – ALARMEOn Alarm n Event Output Enable

Value	Description
0	Alarm n event is disabled and will not be generated.
1	Alarm n event is enabled and will be generated for every compare match.

Bits 0, 1, 2, 3, 4, 5, 6, 7 – PEREOn Periodic Interval n Event Output Enable [n = 7..0]

Value	Description
0	Periodic Interval n event is disabled and will not be generated.
1	Periodic Interval n event is enabled and will be generated.

25.9.4. 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.

Table 25-40. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	15	14	13	12	11	10	9	8
	OVF	TAMPER					ALARM1	ALARM0
Access	R/W	R/W					R/W	R/W
Reset	0	0					0	0

Bit	7	6	5	4	3	2	1	0
	PER7	PER6	PER5	PER4	PER3	PER2	PER1	PER0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 15 – OVF Overflow Interrupt Enable
Writing a '0' to this bit has no effect. Writing a '1' to this bit will clear the Overflow Interrupt Enable bit, which disables the Overflow interrupt.

Value	Description
0	The Overflow interrupt is disabled.
1	The Overflow interrupt is enabled.

Bit 14 – TAMPER Tamper Interrupt Enable

Bits 8, 9 – ALARMn Alarm n Interrupt Enable
Writing a '0' to this bit has no effect. Writing a '1' to this bit will clear the Alarm n Interrupt Enable bit, which disables the Alarm interrupt.

Value	Description
0	The Alarm 0 interrupt is disabled.
1	The Alarm 0 interrupt is enabled.

Bits 0, 1, 2, 3, 4, 5, 6, 7 – PERn Periodic Interval n Interrupt Enable [n = 7..0]
Writing a '0' to this bit has no effect. Writing a '1' to this bit 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.

Table 25-41. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	15	14	13	12	11	10	9	8
	OVF	TAMPER					ALARM1	ALARM0
Access	R/W	R/W					R/W	R/W
Reset	0	0					0	0

Bit	7	6	5	4	3	2	1	0
	PER7	PER6	PER5	PER4	PER3	PER2	PER1	PER0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 15 – OVF Overflow Interrupt Enable
Writing a '0' to this bit has no effect. Writing a '1' to this bit will set the Overflow Interrupt Enable bit, which enables the Overflow interrupt.

Value	Description
0	The Overflow interrupt is disabled.
1	The Overflow interrupt is enabled.

Bit 14 – TAMPER Tamper Interrupt Enable
Writing a '0' to this bit has no effect. Writing a '1' to this bit will set the Tamper Interrupt Enable bit, which enables the Tamper interrupt.

Value	Description
0	The Tamper interrupt it disabled.
1	The Tamper interrupt is enabled.

Bits 8, 9 – ALARMn Alarm n Interrupt Enable
Writing a '0' to this bit has no effect. Writing a '1' to this bit will set the Alarm n Interrupt Enable bit, which enables the Alarm n interrupt.

Value	Description
0	The Alarm n interrupt is disabled.
1	The Alarm n interrupt is enabled.

Bits 0, 1, 2, 3, 4, 5, 6, 7 – PERn Periodic Interval n Interrupt Enable [n = 7..0]
Writing a '0' to this bit has no effect. Writing a '1' to this bit will set the Periodic Interval n Interrupt Enable bit, which enables the Periodic Interval n interrupt.

Value	Description
0	Periodic Interval n interrupt is disabled.

Value	Description
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: -

Note: Interrupt flags must be cleared and then read back to confirm the clear before exiting the ISR to avoid double interrupts.

Table 25-42. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	15	14	13	12	11	10	9	8
	OVF	TAMPER					ALARM1	ALARM0
Access	R/W	R/W					R/W	R/W
Reset	0	0					0	0

Bit	7	6	5	4	3	2	1	0
	PER7	PER6	PER5	PER4	PER3	PER2	PER1	PER0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 15 – OVF Overflow
This flag is cleared by writing a '1' to the flag.
This flag is set on the next CLK_RTC_CNT cycle after an overflow condition occurs, and an interrupt request will be generated if INTENCLR/SET.OVF is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Overflow interrupt flag.

Bit 14 – TAMPER Tamper
This flag is set after a tamper condition occurs, and an interrupt request will be generated if INTENCLR.TAMPER/INTENSET.TAMPER is '1'. Writing a '0' to this bit has no effect. Writing a '1' to this bit clears the Tamper interrupt flag.

Bits 8, 9 – ALARMn Alarm n [n=1..0]
This flag is cleared by writing a '1' to the flag.
This flag is set on the next CLK_RTC_CNT cycle after a match with the compare condition, and an interrupt request will be generated if INTENCLR/SET.ALARMn is one.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Alarm 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 INTENCLR/SET.PERn is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Periodic Interval n interrupt flag.

25.9.7. Debug Control

Name: DBGCTRL
Offset: 0x0E
Reset: 0x00
Property: PAC Write-Protection

Table 25-43. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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: -

Table 25-44. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					GP3	GP2	GP1	GP0
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
	CLOCKSYNC			MASK1	MASK0			
Access	R			R	R			
Reset	0			0	0			

Bit	7	6	5	4	3	2	1	0
		ALARM1	ALARM0		CLOCK	FREQCORR	ENABLE	SWRST
Access		R/W	R/W		R	R	R	R
Reset		0	0		0	0	0	0

Bits 16, 17, 18, 19 – GPn General Purpose n Synchronization Busy Status [n = 3..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.

Bits 11, 12 – MASKn Mask n Synchronization Busy Status [n=1..0]

Value	Description
0	Write synchronization for MASKn register is complete.
1	Write synchronization for MASKn register is ongoing.

Bits 5, 6 – ALARMn Alarm n Synchronization Busy Status [n=1..0]

Value	Description
0	Write synchronization for ALARMn register is complete.

Value	Description
1	Write synchronization for ALARMn register is ongoing.

Bit 3 – CLOCK Clock Register Synchronization Busy Status

Value	Description
0	Read/write synchronization for CLOCK register is complete.
1	Read/write synchronization for CLOCK register is ongoing.

Bit 2 – FREQCORR Frequency Correction Synchronization Busy Status

Value	Description
0	Write synchronization for FREQCORR register is complete.
1	Write synchronization for FREQCORR register is ongoing.

Bit 1 – ENABLE Enable Synchronization Busy Status

Value	Description
0	Write synchronization for CTRLA.ENABLE bit is complete.
1	Write synchronization for CTRLA.ENABLE bit is ongoing.

Bit 0 – SWRST Software Reset Synchronization Busy Status

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: CLK_RTC_CNT = [CLK_RTC / 2(PRESCALAR-1)] * (1 ± [((8192*128)+FREQCORR.VALUE) / (8192*128)])

Table 25-45. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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

Bit 7 – SIGN Correction Sign

Value	Description
0	The correction value is positive, i.e., frequency will be decreased.
1	The correction value is negative, i.e., frequency will be increased.

Bits 6:0 – VALUE[6:0] Correction Value
These bits define the amount of correction applied to the RTC prescaler.

Value	Description
0	Correction is disabled and the RTC frequency is unchanged.
1 - 127	The RTC frequency is adjusted according to the value.

25.9.10. Clock Value in Clock/Calendar mode (CTRLA.MODE=2)

Name: CLOCK
Offset: 0x18
Reset: 0x00000000
Property: PAC Write-Protection, Write-Synchronized, Read-Synchronized

Table 25-46. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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: ALARM
Offset: 0x20 + n*0x08 [n=0..1]
Reset: 0x00000000
Property: PAC Write-Protection, Write-Synchronized

The 32-bit value of ALARM is continuously compared with the 32-bit CLOCK value, based on the masking set by MASK.SEL. When a match occurs, the Alarm n interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.ALARM) is set on the next counter cycle, and the counter is cleared if CTRLA.MATCHCLR is '1'.

Table 25-47. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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 MASK.SEL is 6

Bits 25:22 – MONTH[3:0] Month
The alarm month. Months are matched only if MASK.SEL is greater than 4.

Bits 21:17 – DAY[4:0] Day
The alarm day. Days are matched only if MASK.SEL is greater than 3.

Bits 16:12 – HOUR[4:0] Hour
The alarm hour. Hours are matched only if MASK.SEL is greater than 2.

Bits 11:6 – MINUTE[5:0] Minute
The alarm minute. Minutes are matched only if MASK.SEL is greater than 1.

Bits 5:0 – SECOND[5:0] Second
The alarm second. Seconds are matched only if MASK.SEL is greater than 0.

25.9.12. Alarm Mask in Clock/Calendar mode (CTRLA.MODE=2)

Name: MASK
Offset: 0x24
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized

Table 25-48. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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

Bits 2:0 – SEL[2:0] Alarm Mask Selection
These bits define which bit groups of ALARM 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: GPn
Offset: 0x40 + n*0x04 [n=0..3]
Reset: 0x00000000
Property: PAC Write-Protection, Write-Synchronized

Table 25-49. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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

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

Bit	15	14	13	12	11	10	9	8
	GP[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

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

Bits 31:0 – GP[31:0] General Purpose
These bits are for user-defined general purpose use, see *General Purpose Registers*.

25.9.14. Tamper Control

Name: TAMPCTRL
Offset: 0x60
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Table 25-50. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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 Layer Protection mode (TAMPCTRL.INACT = ACTL).

Value	Description
0	Debouncing is disabled for Tamper input INn
1	Debouncing is enabled for Tamper input INn

Bits 16, 17, 18, 19, 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 Layer Protection mode (TAMPCTRL.INACT = ACTL).

Value	Description
0	A falling edge condition will be detected on Tamper input INn.
1	A rising edge condition will be detected on Tamper input INn.

Bits 0:1, 2:3, 4:5, 6:7, 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

Value	Name	Description
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: -

Table 25-51. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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

Table 25-52. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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 on 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

Table 25-53. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
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 Layer Internal Select n [n=0..7]

Note: Only one ALSI bit must be set to enable Active Layer Protection on the TrustRAM.

Value	Description
0	Active layer Protection is monitoring the RTC signal using INn and OUTn tamper pins
1	Active layer Protection is monitoring the RTC signal on the TrustRAM shield

26. Direct Memory Access Controller (DMAC)

26.1. Overview

The Direct Memory Access (DMA) Controller implements data transfers across the system bus without the intervention of the CPU, normally in a stream fashion called a burst of (32-bit) word transfers (beats).

The DMA Controller module 0 has 8 channels, and DMA Controller module 1 has 4 channels. Please see the Block Diagram for a DMA Controller top-level block diagram.

The DMA Controller data path comprises:

- A DMA Read Bus Host (DMAR), common to all DMA Channels
- A DMA Write Bus Host (DMAW), common to all DMA Channels
- Small data FIFOs to handle the data in transit from the Source to Destination, with one FIFO per each DMA Channel (FIFO0 to FIFON-11 in the Block Diagram)

The DMA Controller is useful in one of the four configurations presented in the following figures.

Figure 26-1. Input Configuration

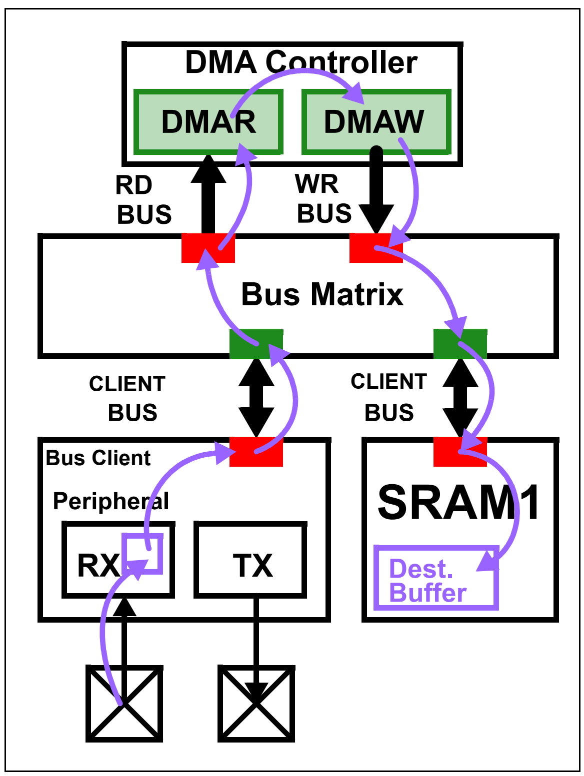


Figure 26-2. Output Configuration

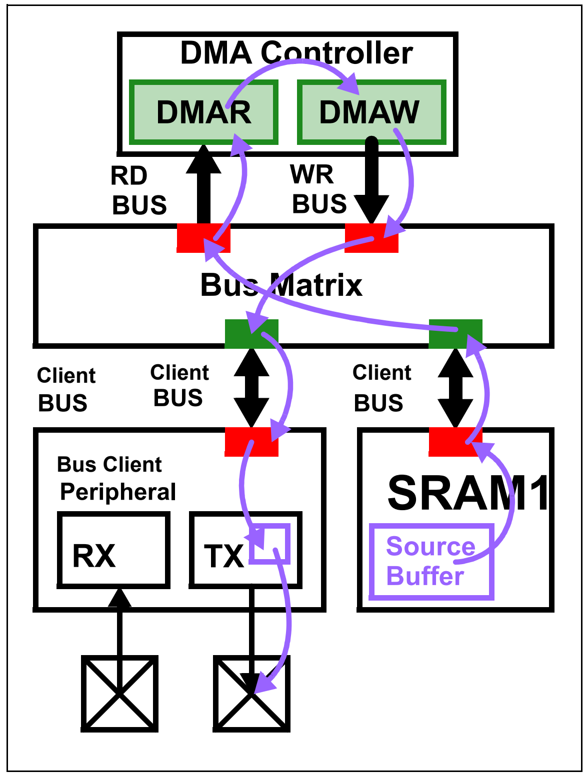


Figure 26-3. Input/Output Configuration

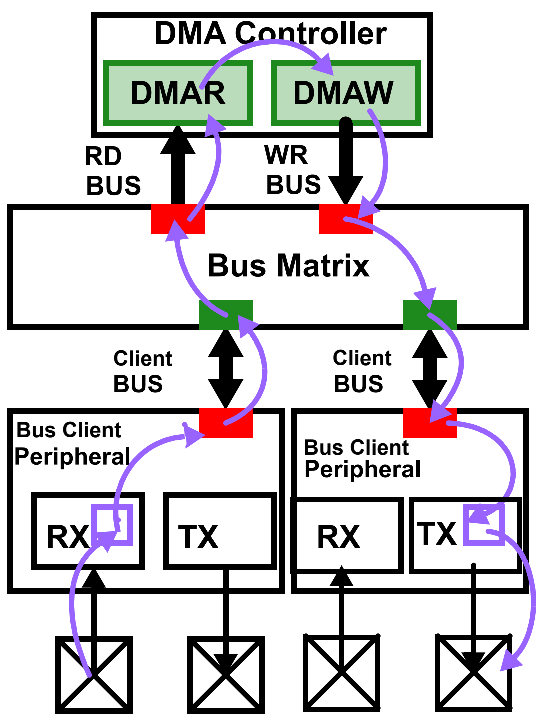
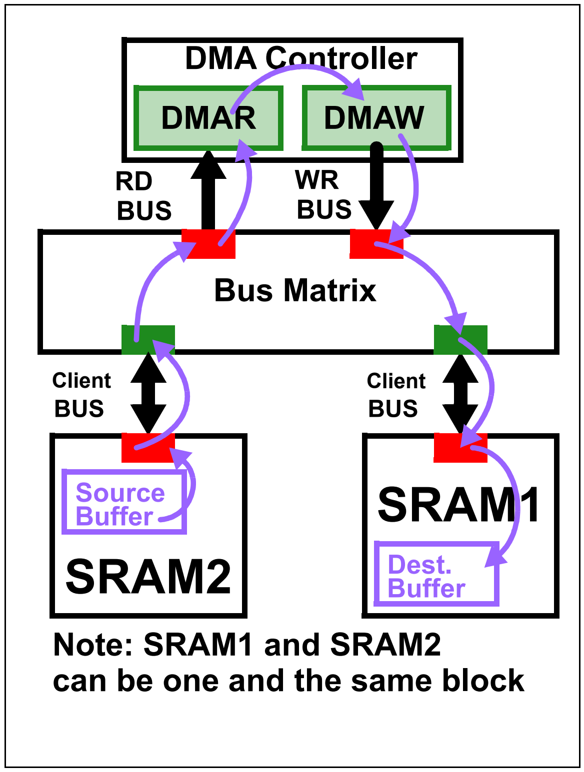


Figure 26-4. Transfer Configuration



26.2. Features

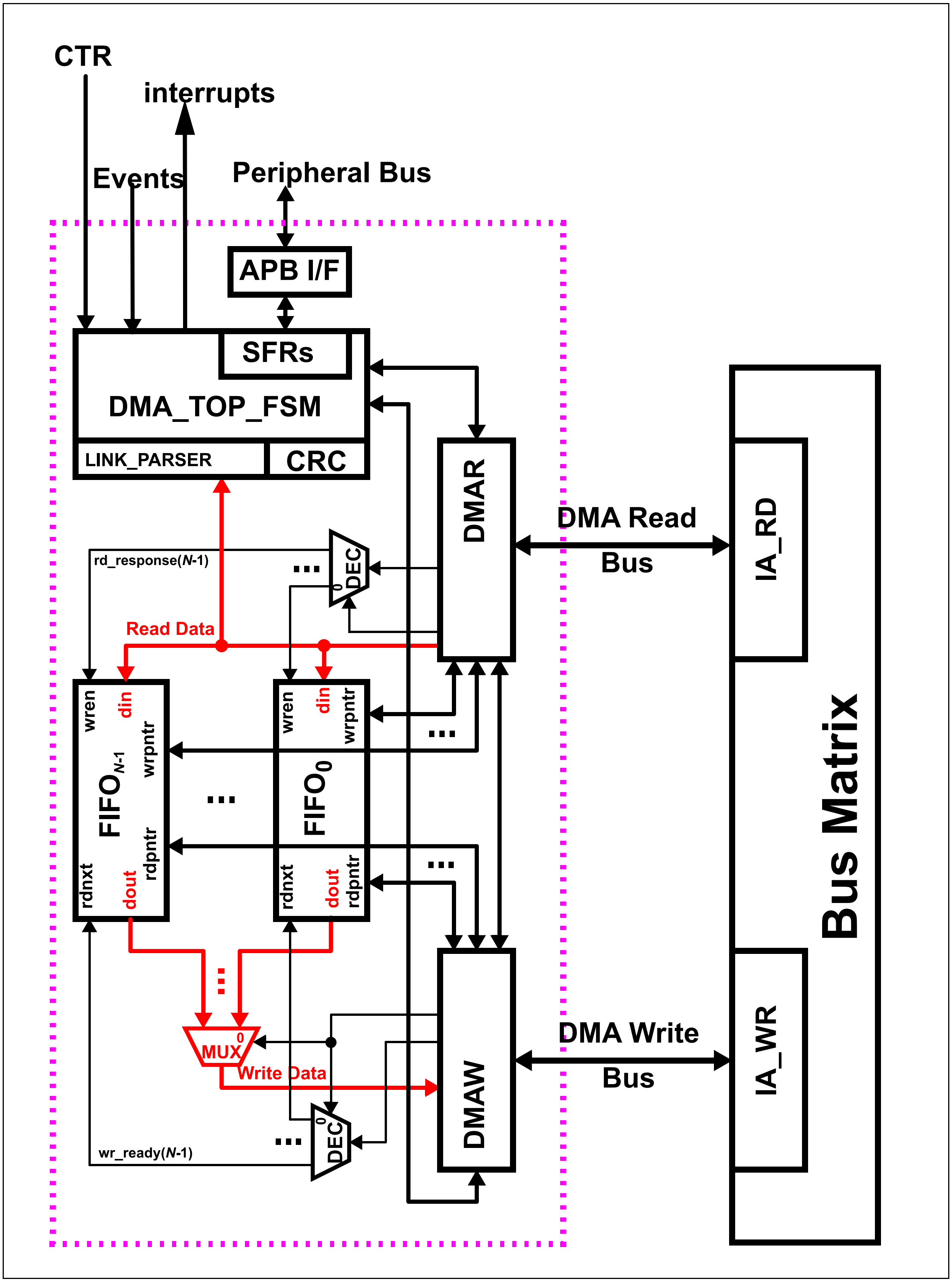
The following are key features of the DMAC module:

- 8 DMA channels on module 0 and 4 DMA channels on module 1
- Linked-List gather and scatter programing model
- Advanced split transaction AHB bus interface with bursting
- Memory-to-Memory transfer capability
- Memory-to-Peripheral transfers capability
- Peripheral-to-Memory transfers capability
- Peripheral-to-Peripheral transfer capability
- Several different block transfer modes affecting the bus transfer protocol and speed with the following characteristics:
  - Transfer sizes of up to 64 KB
  - Byte Transfer mode without burst
  - Half word Transfer mode without burst
  - Auto Transfer mode, where the DMA adjusts Transfer modes to optimize transferred data
  - Fixed or Incrementing address modes
  - 32-bit Source and destination addresses
  - 64 bit Data path Support
- Assignable channel priority level:
  - User assignable channel priorities
  - Fixed natural order priority arbitration

  - Priority levels may be shared.
  - QOS Support per priority group
  - Priority increase on event system trigger
- Channel Start/Abort Triggers:
  - Software start trigger
  - Event system start trigger
  - Conditional start trigger
  - Event system abort trigger
  - Peripheral DMA request trigger
  - Enhanced pattern (data) match transfer termination
  - Bus fault abort
- Multiple DMA channel status interrupts:
  - Transfer complete
  - Transfer started
  - Transfer midpoint reached
  - Transfer aborted
- Single Clock Cycle CRC/Checksum Engine:
  - Provides independent CRC functions to each channel simultaneously (in parallel when activated)
  - Supports preprogrammed CRC-16 (IBM/ANSI), CRC-16 CCITT, CRC-32
  - Provides two user programmable polynomial registers for 32-bit or 16-bit CRCs. These registers are shared amongst the channels.
- Event system output trigger per channel
- APB connectivity for the SFRs

26.3. Block Diagram

Figure 26-5. DMA Controller Top Block Diagram



26.4. Peripheral Dependencies

Peripheral Name	Base Address	NVIC IRQ Index: Source	MCLK AHBx/APBx Clock Enable Mask Bit	PAC Peripheral Identifier (PAC.WRCTRL.PERIDx)	EVSYS Users (EVSYS.USERm)	EVSYS Generator (EVSYS.CHANNELn.EVGENx)	Power Domain
DMA0	0x4480_2000	33 : Priority 0 34 : Priority 1 35 : Priority 2	AHB : MCLK.CLKMSK0[8] APB : MCLK.CLKMSK2[1]	21	6-13 : CHx_Start, x=0,1,...7 14-21 : CHx_Aux, x=0,1...7	38-45 : DMA0_CHx, x=0,1,...7	VDDCORE_SW
DMA1	0x4480_4000	36 : Priority 0 37 : Priority 1	AHB : MCLK.CLKMSK0[9] APB : MCLK.CLKMSK2[2]	22	22-25 : CHx_Start, x=0,1,2,3 26-29 : CHx_Aux, x=0,1,2,3	46-49 : DMA1_CHx, x=0,1,2,3	VDDCORE_SW

26.4.1. Power Management

The DMAC will continue to operate in any sleep mode where the selected source clock is running. The DMAC's interrupts can be used to wake up the device from sleep modes. Events connected to the event system can trigger other operations in the system without exiting sleep modes. On hardware or software reset, all registers are set to their reset value.

26.4.2. 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 DBGCTRL SFR for details.

26.4.3. Register Access Protection

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

- Interrupt Priority Status Registers (INTSTATn)
- Channel Interrupt Registers (CHINTENCLRn, CHINTENSETn, CHINTFn)
- All of the channel status registers

Optional write-protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write-Protection" property in each individual register description. PAC write-protection does not apply to accesses through an external debugger.

26.5. Indexing

There are 8 DMA channels on module 0 and 4 DMA channels on module 1, indexed by k, k = 0,1,...,n. Each channel has 19 dedicated channel-specific registers, which are identified by the channel index as a suffix:

CHCTRLAk - Channel Control Register A

CHCTRLBk - Channel Control Register B

CHEVCTRLk - Channel Event Control Register

CHINTENCLRk - Channel Interrupt Enable Clear Register

CHINTENSETk - Channel Interrupt Enable Set Register

CHINTFk - Channel Interrupt Flag Register

CHSSAk - Channel Source Start Address

CHDSAk - Channel Destination Start Address

CHSSTRDk - Channel Source Cell Stride Size Register

CHDSTRDk - Channel Destination Cell Stride Size Register

CHXSIZk - Channel Transfer Size Register

CHPDATk - Channel Pattern Match Data

CHCTRLCRCk - Channel Control Crc

CHCRCDATk - Channel CRC/Checksum Data Register

CHNXTk - Channel Next Descriptor Address Pointer

CHLLCFGSTATk - Channel Linked List Configuration Status Register

CHSTATBCk - Channel Status Block Count Register

CHSTATCCk - Channel Status Cell Count Register

CHSTATk - Channel Status Register

So, the Channel Control Register A for Channel 5 (k=5) is CHCTRLA5.

26.6. DMA Event/Trigger Mapping

Table 26-1. DMA Event/Trigger Mapping

Instance	Channel	CHCTRLBk.TRIG[index]	Comment
DSU	DCCx	2, 3	DMAC ID for DCC, x=0,1
RTC	TIMESTAMP	4	RTC Timestamp trigger
SERCOM0	RX	5	DMA RX trigger
	TX	6	DMA TX trigger
SERCOM1	RX	7	DMA RX trigger
	TX	8	DMA TX trigger
SERCOM2	RX	9	DMA RX trigger
	TX	10	DMA TX trigger
SERCOM3	RX	11	DMA RX trigger
	TX	12	DMA TX trigger
TCC0	OVF	13	DMA overflow/underflow/retrigger
	MCx	14-19	DMA Match/Compare triggers x=0,1,...,5
TCC1	OVF	20	DMA overflow/underflow/retrigger
	MCx	21-26	DMA Match/Compare triggers x=0,1,...,5
TCC2	OVF	27	DMA overflow/underflow/retrigger
	MCx	28-33	DMA Match/Compare triggers x=0,1,...,5
TCC3	OVF	34	DMA overflow/underflow/retrigger
	MCx	35-40	DMA Match/Compare triggers x=0,1
SERCOM4	RX	41	DMA RX Trigger
	TX	42	DMA TX Trigger
SERCOM5	RX	43	DMA RX Trigger

Table 26-1. DMA Event/Trigger Mapping (continued)

Instance	Channel	CHCTRLBk.TRIG[index]	Comment
	TX	44	DMA TX Trigger
SERCOM6	RX	45	DMA RX Trigger
	TX	46	DMA TX Trigger
SERCOM7	RX	47	DMA RX Trigger
	TX	48	DMA TX Trigger
TCC4	OVF	49	DMA overflow/underflow/retrigger
	MCx	50-51	DMA Match/Compare triggers x=0,1
TCC5	OVF	52	DMA overflow/underflow/retrigger
	MCx	53-54	DMA Match/Compare triggers x=0,1
TCC6	OVF	55	DMA overflow/underflow/retrigger
	MCx	56-57	DMA Match/Compare triggers x=0,1
TCC7	OVF	58	DMA overflow/underflow/retrigger
	MCx	59-60	DMA Match/Compare triggers x=0,1
ADC	PFFRDY	61	ADC DMA PFFRDYT TRIGGER
PTC	SEQ	62	PTC Ready Trigger
	WCOMP	63	PTC Window
	EOC	64	PTC End of Conversion
SPI_IXS0	DREQ_RX	65	Indexes of DMA RX Triggers
SPI_IXS0	DREQ_TX	66	Indexes of DMA TX Triggers
PCC	RX	67	PCC RX Trigger
CAN0	DEBUG	68	DMA CAN Debug Req
CAN1	DEBUG	69	DMA CAN Debug Req

26.7. Applications

26.7.1. Overview

The Direct Memory Access (DMA) Controller can transfer data between memories and peripheral at a high data transfer rate without CPU intervention.

This module provides up to 12 independent DMA channels. Each channel can be uniquely configured to transfer data from a source location to a destination location on a trigger event. Channels share access to the system bus through a Read and Write port as shown in the block diagram.

The block diagram shows the major components of the DMA which include the read port (DMAR), the write port (DMAW), the channel FIFOs, the FSM (DMA_TOP_FSM) which includes the link list parser (LINK_PARSER), the CRC engine, and the SFR registers.

The read port allows the DMA to load descriptors and read source data. Channels arbitrate for access to the read port based on both a natural order priority and a software programmable priority.

The write port allows the DMA to write data to the destination address. As with the read port, channels arbitrate for access to the write port based on natural order priority and software programmable priority.

The channel FIFO provides storage for data that is in transit from source to destination.

The FSM manages configuration updates for each channel, trigger events, arbitration, linked list parsing, and interrupts. The FSM handles configuration of the CRC engine when processing source data from one channel to another.

26.7.2. Operation

The DMA provides two main functions on each channel; transferring data from source to destination called a block transfer, and traversing a linked list of descriptors in system memory, where each descriptor defines a block transfer.

26.7.3. Block Transfer

Each DMA channel performs data transfers on a trigger event. A trigger event is either a hardware signal from the device event system, a hardware signal from a peripheral, or a write to the CHCTRLAk.SWFRC register field. The smallest amount of data transferred on a trigger event is defined as a cell transfer. A block transfer may consist of multiple cell transfers as show in the following figure. The DMA provides flexibility in defining the cell transfer size and block transfer size. However, the block transfer size is always greater than or equal to the cell transfer size. A block transfer size does not have to be an integer multiple of the cell transfer size.

To specify a block transfer, a DMA channel minimally requires the following

- Source Start Address
- Destination Start Address
- Block Transfer Size
- Cell Transfer Size
- Source Cell Stride Size
- Destination Cell Stride Size
- Read Address Sequence Mode
- Write Address Sequence Mode
- Start trigger event source

Once configured a channel will wait for the trigger event to initiate a transfer. On the trigger event, the channel requests access to DMAR. The DMA FSM arbitrates access based on channel and user selected priorities. See *Channel Priorities* for details on how arbitration takes place.

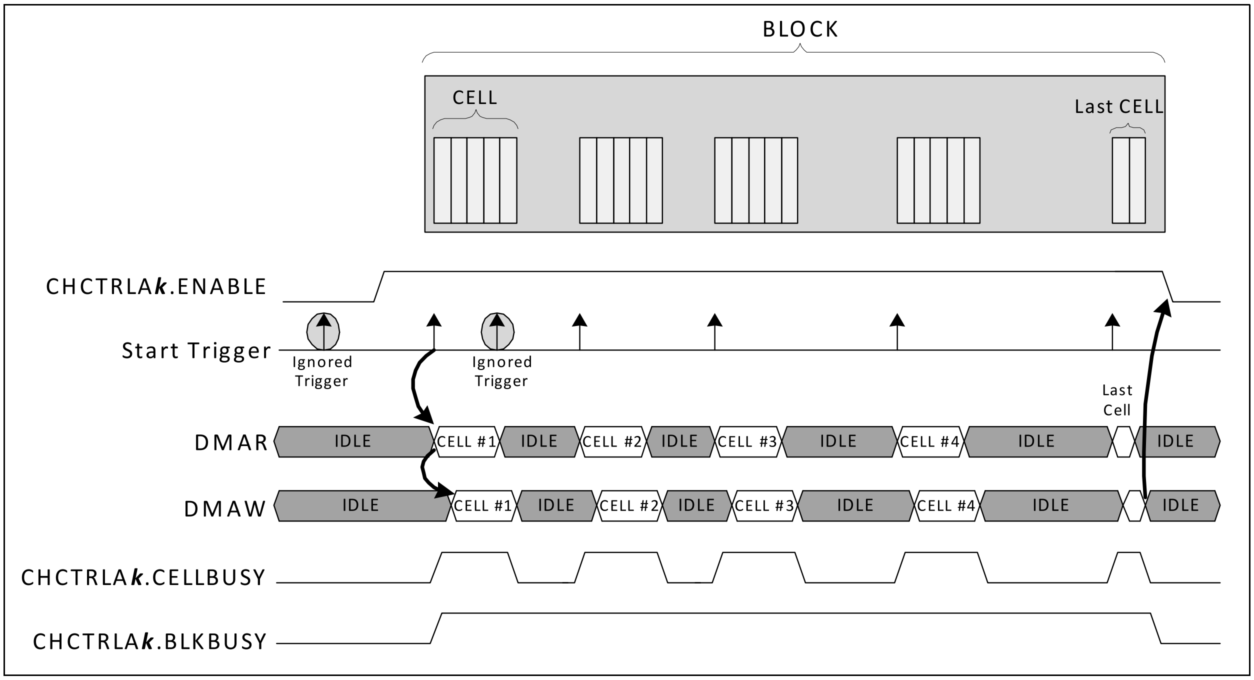
The channel issues read requests of the source data, either in burst fashion or as single transactions until the channel FIFO is full or is not capable of accepting another burst. The DMA makes the determination of using a burst transfer or single requests based on the Read Address Sequence Mode setting and address alignment of the source data. The channel request another read transaction when the channel FIFO has enough room to accept data. The read process continues until the number of bytes read equals the Cell Transfer Size.

On the write size of the DMA, once the channel FIFO has enough data to perform a burst write or single write transaction, the channel requests access to the DMAW. The DMA FSM arbitrates access to the port based on priority settings. The channel writes out data from the FIFO in burst or single transactions based on the Write Address Sequence Mode and destination address alignment. The write process continues until the number of bytes written equals the Cell Transfer Size.

Once a cell transfer completes, the channel waits for another trigger event to start the next cell transfer. The cell transfers repeats until the number of bytes transferred equals a Block Transfer Size. If the Block Transfer Size is not an integer multiple of the Cell Transfer Size, the last cell transfer will truncate to the remaining bytes available in the block transfer.

The following figure depicts the relationship of cells and blocks and how triggers events initiate cell transfers. The DMA performs five cell transfers to complete the block transfer. The last cell transfer is truncated to match the block size.

Figure 26-6. Block Transfer



26.7.3.1. Address Sequence and Transfer Modes

The address sequence mode which is individually programmable for the source and destination, determines how the DMA calculates the read and write addresses and the amount of data that can be transferred in a single bus transaction (beat). The CHCTRLB***k***.RAS[2:0] register field sets the address sequence mode for source reads. The CHCTRLB***k***.WAS[2:0] register field sets the address sequence mode for the destination writes.

There are two address modes that are supported, fixed address and increment address. In fixed address mode, the address does not increment between consecutive transactions. In increment address mode, the address increments based on the number of bytes transferred per beat.

The following address and byte transfer combinations are supported

- Fixed Address / Byte transfer
- Fixed Address / Half word transfer
- Fixed Address / Word transfer
- Increment Address / Byte transfer
- Increment Address / Half word transfer
- Increment Address / Auto transfer size

In Fixed Address/ Byte transfer mode the address is fixed to the start address (see *Source Start Address* and *Destination Start Address* for a start address definition) with no address alignment restrictions. Read or write beats are of byte length. Use this address sequence mode to read or write a peripheral FIFO with byte wide access. Similarly, in Fixed Address / Half word mode, the address is fixed to the start address which must be half word aligned. Read or write beats are two bytes wide. In Fixed Address / Word mode the address is fixed to the start address and must be aligned to a word boundary. Read or write beats are 4 bytes wide. The DMA is capable of bursting up to 4 beats to improve bus traffic efficiency in Fixed Address / Word mode.

In Increment Address / Byte transfer mode the read or write address increments by one byte after each beat, beginning at the start address. In Increment Address / Half word transfer mode the read or write address increments by two bytes after each beat, beginning at the start address. The start address must be half word aligned. In Increment Address/ Auto transfer mode, the beat size and address increment amount are determined by the DMA to optimally read or write data. If the start address is aligned to the interface data width (64 bits) and the block and cell transfer sizes are also aligned to the interface data width, all beats in this mode are the width of interface and bursts of up to 4 beats are possible.

For an example of the Increment Address/ Auto, consider a data packet of 64 bytes in memory starting at address 0x0000_0001, The data packet needs to be copied into a communication peripheral's FIFO which can accept word writes and can hold 16 bytes of data at one time. The Source Start Address is set to 0x0000_0001. The Destination Start Address is set to the FIFO register location which is on a word boundary. The Block Transfer Size is set to the packet size of 64 bytes. The Cell Transfer Size is set to 8 bytes, which is half of the FIFO's capacity. The Read Address Sequence is set to Increment Address / Auto and the Write Address Sequence is set to Fixed Address / Word. The interface data width is 32-bits wide. The channel is programed to trigger a cell transfer when the peripheral FIFO is less than half full.

On the first trigger, the DMA will issue a single byte read starting at 0x0000_0001, a half-word read at 0x0000_0002, followed by a full word read, and finally a single byte read at 0x0000_0008. In the channel FIFO, the DMA realigns data to match the beat size of the data word. Two words are burst out using fixed address set by the Destination Start Address. At this point the DMA will wait for another trigger event from the peripheral before transferring another cell. Cell transfers continue until a block transfer completes.

26.7.3.2. Source Start Address

The Source Start Address is a 32-bit physical address location within the device's memory map, which defines the starting address to perform source reads. Software programs the CHSSA***k***.SSA[31:0] register to the desired start address for channel ***k.*** The start address may be restricted to word or half word address alignment depending on the Read Address Sequence Mode.

26.7.3.3. Destination Start Address

Similarly, the Destination Start Address is a 32-bit physical address location within the device's memory map, which defines the starting address to perform destination writes. Software programs the CHDSA***k***.DSA[31:0] register to the desired start address. The start address may be restricted to word or half-word address alignment depending on the Write Address Sequence Mode.

26.7.3.4. Block Transfer Size

The Block Transfer Size defines the total number of bytes to transfer from source to destination before setting the Block Transfer Complete interrupt status flag, CHINTF***k***.BC and clearing the CHCTRLA***k***.ENABLE bit. See Block Transfer Enable for details on CHCTRLA***k***.ENABLE. The CHXSIZ***k***.BLKSZ register field holds the Block Transfer Size. The size byte alignment (byte, halfword, or word) must match the most restrictive address sequence mode select between Read and Write Address Sequence Mode settings. The following table shows the size restriction.

Table 26-2. Block and Cell Size Restrictions

RAS Transfer Mode	WAS Transfer Mode	Size Alignment Restriction
Fixed or Increment / Byte	Fixed or Increment / Byte	None
Fixed or Increment / Byte	Fixed or Increment / Halfword	Halfword
Fixed or Increment / Byte	Fixed / Word	Word
Fixed or Increment / Byte	Increment /Auto	None
Fixed or Increment / Halfword	Fixed or Increment / Byte	Halfword
Fixed or Increment / Halfword	Fixed or Increment / Halfword	Halfword
Fixed or Increment / Halfword	Fixed / Word	Word
Fixed or Increment / Halfword	Increment /Auto	Halfword
Fixed / Word	Fixed or Increment / Byte	Word
Fixed / Word	Fixed or Increment / Halfword	Word

Table 26-2. Block and Cell Size Restrictions (continued)

RAS Transfer Mode	WAS Transfer Mode	Size Alignment Restriction
Fixed / Word	Fixed / Word	Word
Fixed / Word	Increment /Auto	Word
Increment / Auto	Fixed or Increment / Byte	None
Increment / Auto	Fixed or Increment / Halfword	Halfword
Increment / Auto	Fixed / Word	Word
Increment / Auto	Increment /Auto	None

26.7.3.5. Cell Transfer Size

The CHXSIZ***k***.CSZ register sets the cell transfer size in bytes. This defines the smallest amount of data to transfer on a trigger event. The cell transfer size byte alignment must match the most restrictive address sequence mode as shown in Block and Cell Size Restrictions.

26.7.3.6. Cell Stride Size

The Cell Stride Size provides the increment value between successive cells transfers. The CHSSTRD***k***.SSTRD register field holds the Cell Stride Size for the source. The CHDSTRD***k***.DSTRD register field holds the Cell Stride Size for the destination. The Cell Stride Size is only used in increment address modes. The DMA adds the Cell Stride Size to the last address of the current cell to determine the starting address of the next cell. For block transfers with data located in contiguous memory locations, the Cell Stride Size is set to zero. The Cell Stride Size follows the same alignment restrictions as the Start Address. See Linked-List with Cell Striding Example for an example of how to use cell striding to transfer data that resides in a non-contiguous memory space.

26.7.3.7. Block Transfer Enable

The CHCTRLAk.ENABLE bit controls the channel's ability to perform a block transfer. Setting the bit to 1, either by software or through the linked-list descriptor (see Linked List for details on linked-list operation) notifies DMA FSM that the channel is configured and a start trigger event shall initiate a transfer. On completion or abort of a block transfer, the DMA will always clear CHCTRLAk.ENABLE.

If software clears CHCTRLAk.ENABLE during a block transfer, and a block transfer is in progress (CHSTATk.BLKBUSY=1), the channel suspends the transfer. The CHCTRLAk.ENABLE bit will reflect the written state. When suspended, no new bus requests are issued for the channel and all trigger events are ignored. Setting CHCTRLAk.ENABLE to 1 after suspending resumes the transfer. If CHSTATk.CELLBUSY=1, the current cell transfer resumes. If any of the control registers are modified when suspended, the channel cancels the current block transfer and resets. Channel reset consists of clearing all the status bits, resetting counters to zero, and flushing the channel FIFO. The CHSTATk.BLKBUSY clears at the completion of the channel reset sequence.

26.7.3.8. Start Trigger Source

The DMA provides three types of start triggers which can initiate a cell transfer, a software trigger, an event system trigger, and a peripheral DMA request.

The CHCTRLA*k*.SWFRC bit provides the software trigger. When software or the linked-list descriptor sets CHCTRLAk.SWFRC, the DMA issues a start trigger to the channel.

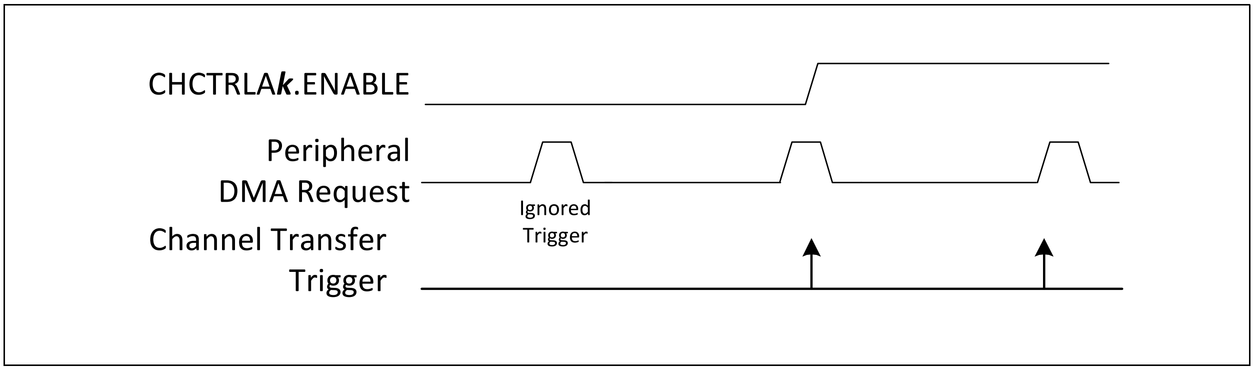
The device event system provides an event trigger for each DMA channel *k* where *k* < 24 (24 start event triggers). Setting CHCTRLB*k*.TRIG=1 configures the channel to use the rising edge of the event trigger as the start trigger.

A channel is also capable of using a peripheral DMA request as the start trigger. Setting CHCTRLB*k*.TRIG to a value greater than one will select one of the connected peripheral's DMA request as the start trigger.

For a channel to take action on a start trigger, the CHCTRLA*k*.ENABLE bit must be set to one, otherwise all trigger events are ignored as shown in Block Transfer.

If a peripheral DMA request is pending when the channel is enabled, the channel will honor the request and initiate a cell transfer. This is shown in the following figure.

Figure 26-7. Peripheral DMA Request at Channel Enable



Start trigger events that occur during a cell transfer, CHSTAT*k*.CELLBUSY=1, are ignored as shown by the second ignored trigger in Block Transfer.

26.7.3.8.1. Conditional Trigger

The DMA can use the auxiliary event system trigger to qualify either the start event system trigger or a peripheral DMA request. When enabled as a conditional trigger, start trigger configured with CHCTRLB*k*.TRIG and the auxiliary event system trigger are logically AND'ed to produce the start trigger on the rising edge of the AND'ed signal. The conditional trigger function is enabled by setting the CHEVCTRL*k*.EVAUXIE=1 and CHEVCTRL*k*.EVAUXACT=2.

26.7.3.9. Abort Trigger Source

There are two abort triggers which can terminate a block transfer, an event system trigger and a pattern match trigger.

The channel auxiliary event system trigger can be configured to provide an abort transfer trigger to the channel. Setting CHEVCTRLk.EVAUXIE=1 and CHEVCTRLk.EVAUXACT=0 or CHEVCTRLk.EVAUXACT=3 enables abort triggering. An event system abort takes effect on the rising edge of the event signal. The CHCTRLAk.ENABLE bit is cleared and all new trigger events are ignored. The channel will stop all new reads. Any reads already issued on the bus are completed normally. Pending writes complete without interruption. Any remaining data in the FIFO is written out to the destination as long as the DMA can honor the address/data alignment set by CHCTRLAk.WAS. Odd bytes remaining in the FIFO are discarded. When all activity concludes, the status bits in CHSTATk are cleared and the BC, CC, and TA interrupt flags in register CHINTFk are set.

The DMA also provides the ability to abort a block transfer on detection a data pattern match. If CHCTRKCk.PATEN=1, a data pattern match can issue an abort trigger to the channel. See Pattern Match Termination for more details on how the pattern match works. On a pattern match abort, the CHCTRLk.ENABLE bit clears and all new trigger events are ignored. The block transfer terminates after writing the matched pattern data to the destination. Data read after the pattern match is ignored. When all activity concludes, the status bits in CHSTATk are cleared and the BC, CC, and TA interrupt flags are set in register CHINTFk.

26.7.3.10. Pattern Match Termination

Pattern match allows the user to end a block transfer if bytes of data read during a transaction matches a specific pattern, as defined by the PDATA register. Upon the detection of the pattern, an abort trigger is sent to the channel. Any bytes following the pattern match will not be written to the destination. The CHCTRLBk.PATEN enables the pattern detection abort function. Pattern matching is independent of where the data pattern starts in the byte stream of the source data. Pattern matching is always performed in the original order in which data was read, therefore CHCTRLBk.BYTORD[1:0] does not affect pattern matching.

Since pattern matching operates on a byte boundary, if the WAS alignment is set to a halfword or fixed word, the DMA may perform a byte wide write on the last transaction depending on where the pattern match is detected.

26.7.3.10.1. Byte Ignore

When two-byte pattern detect is enabled and the ignore byte is also enabled, any read data byte encountered that has the same bit pattern as the CHPDATk.PIGN[7:0] data field will be treated as a don’t care during the pattern matching process. The read data will still be transferred; it just will be ignored when attempting to determine a pattern match. When CHCTRLBk.PIGNEN=0 the CHPDATk.PIGN[7:0] has no effect on pattern matching.

Note: When PIGNEN=1 the user should ensure that the byte held in PIGN for that channel is not the same as any byte(s) specified in PDATA that is participating in the pattern match.

26.7.3.10.2. Pattern Matching

When the Pattern is only a single byte in length the first byte in the read data that matches is a pattern match. Any valid bytes following the pattern match in the read data shall not be written. The read data up to and including the pattern match is placed in the channel FIFO, and further reads for that channel cease. When all the data in the FIFO is written out to the destination, the DMA clears CHCTRLAk.ENABLE and sets the appropriate interrupt flags as described in Abort Trigger Source.

When the Pattern consists of two bytes, a match requires a read byte to match the value in CHPDATk.PDAT[7:0] and the next read byte to match CHPDATk.PDAT[15:8]. If CHCTRLBk.PIGNEN=1 then the two matched bytes may be separated by any number of don’t care bytes as defined by CHPDATk.PIGN[7:0]. Any read bytes following the pattern match are not written out to the destination. The read data up to the final byte of the pattern match is placed in the channel FIFO, further reads for that channel cease, When all the data in the FIFO is written out to the destination, the DMA clears CHCTRLAk.ENABLE and sets the appropriate interrupt flags as described in Abort Trigger Source.

Bytes matching the pattern may be spread over multiple consecutive reads for a channel and must be treated as a contiguous stream.

Consider as a **pattern match example** a two byte pattern that is to match on CR and LF. The option of ignoring null characters could be considered by setting PIGN to zero and enabling the ignore feature with PIGNEN=1. This way, when a CR character occurs followed by 0 or more null characters and then followed by a LF character occurs, the match is detected when the LF character is encountered.

In either case bytes that have been read that follow the matching pattern will not be part of the final data transaction. Pattern matching occurs before any output data reordering.

Note: A pattern match may only occur within the limits of the transfer of data as described by a linked list descriptor. The pattern match may **not** span across descriptor transfers.

26.7.3.10.3. Pattern Match Example

Assuming a system that has a series of messages that are routinely transmitted to an external host through the UART. Assuming a **maximum** message size of 86 characters. The user would set the following parameters on the channel:

- Source Start Address is set to the start of the message
- Destination Start Address is set to the UART transmit FIFO
- Block Transfer Size is set to 86 bytes
- Cell Transfer Size is set to 1/2 the UART transmit FIFO depth.
- Source Cell Stride Size is set to the Cell Transfer Size
- Destination Cell Stride Size is ignored due to the Fixed Address mode setting for the destination.
- Read Address Sequence Mode is set to Increment Address / Auto

- Write Address Sequence Mode is set to Fixed Address / Byte. This assumes the UART transmit FIFO can only accept a single byte per write.
- Start trigger event source is set to the UART DMA transmit request which asserts when the UART transmit FIFO is half full.
- The channel is configured to accept a pattern match abort, CHCTRLBk.PATEN=1, of a single byte length pattern, CHCTRLBk.PATLEN=0, of 0x00, CHPDATk.PDAT[7:0] = 0x00.

Once CHCTRLAk.ENABLE is set to 1, the DMA will transfer data to the UART based on the UART DMA request for data until the 0x00 byte pattern is detected.

Note: The pattern match bytes and don't care bytes are always written to the destination.

26.7.3.11. Single Trigger Block Transfer

The DMA provides the ability to perform a block transfer with a single start trigger as opposed to providing start triggers for each cell transfer. Setting the CHCTRLB***k***.CASTEN to 1 enables this function. This function is useful for memory to memory transfers. Note, the Cell Transfer Size still effects arbitration of port resources between channels in the same priority group. The user should specify a Cell Transfer Size that does not starve lower priority channels from port resources. See Channel Priorities for more details on arbitration.

26.7.3.12. Bus Error Handling

If a channel receives a bus error for a read or write request, the DMA will clear the CHCTRLAk.ENABLE and CHCTRLAk.LLEN fields to disable the channel. The status bits in CHSTATk are cleared when all pending transactions complete. For an error on the DMAR port, the DMA sets the CHINTF***k***.RDE interrupt flag and issues an interrupt. For an error on the DMAW port, the DMA sets the CHINTF***k***.WRE interrupt flag and issues an interrupt. The interrupt flags are set after clearing the status bits in CHSTAT***k***.

26.7.3.13. Data Reordering

Data read from the source can be re-ordered to allow for various byte orders of the source data, such as endianness. The re-ordered source data is written to the channel destination when CHCTRLB***k***.WBOEN = 1'. The unaltered initial source data is written to the destination when CHCTRLB***k***.WBOEN = 0. The following figure shows the different byte order settings and the effect on data reads. BYTORD value of '01' is useful for re-ordering bytes within words. While BYTORD values of '10' and '11' is useful for re-ordering bytes within half-words.

Notes:

1. Data is re-ordered as it is read. This means that data that is not word-aligned may not be re-ordered correctly. Both the block and cell sizes must also be word-aligned.
2. The CRC will process data such that the final result is the same regardless of read address mode (byte, halfword, or word). This is completely transparent to software. This reordering for the CRC block is explained in more detail in CRC and Checksum Engine and is independent of the BYTORD setting for the channel. In this way, this would allow the user, in CRC mode, to byte reorder the data from source to destination using a different swapping than is required for the CRC calculation.

Figure 26-8. CHCTRLB.BYTORD[1:0] Effect on Data

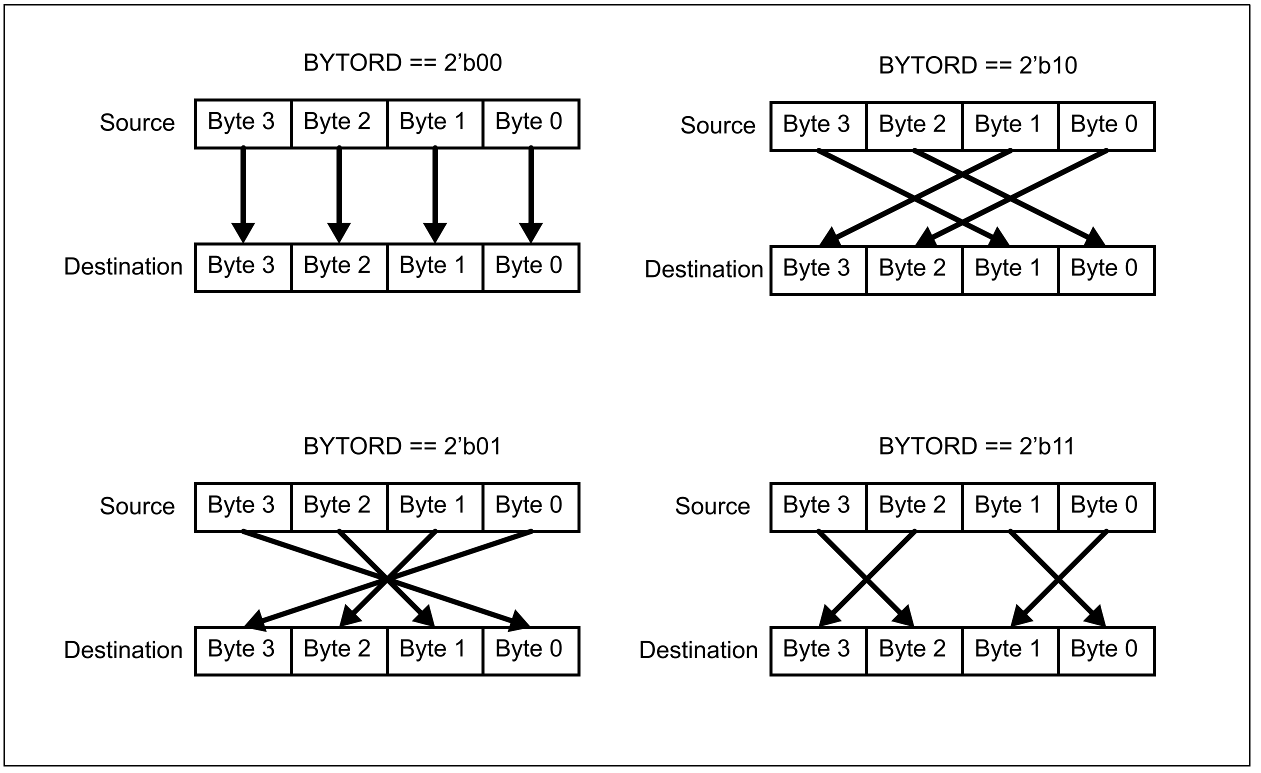


Table 26-3. Example Data Re-order

	Byte Order			
	3	2	1	0
Source Address	0x78	0x56	0x34	0x12
Destination Address	0x12	0x34	0x56	0x78

26.7.4. Linked List

This module provides linked-list operation which allows a DMA channel to perform more than one block transfer without the CPU intervening in-between block transfers. A linked-list is data structure that resides in system memory which consists of descriptors linked together by address pointers contained within the descriptor. Each descriptor can completely define a block transfer. Linked List Block Descriptor describes the contents of a descriptor. The DMA is capable of loading the descriptor, configuring the channel based on the descriptor, performing a block transfer and then loading the next descriptor in the linked-list autonomously.

To enable link list mode, software first programs the CHNXT***k***.NXT register field with a pointer to the first descriptor in the linked-list, then sets the CHCTRLA***k***.LLEN bit to 1 which enables linked list operation. The DMA will then request access to the read port and perform a descriptor load.

The DMA always loads the first and second words of the descriptor. The first word contains the next pointer in the linked list chain. This word is loaded into CHNXT***k***.NXT. The second word contains the BDCFG.RUNSTDBY, BDCFG.SWFRC, BDCFG.LLEN, BDCFB.ENABLE, and controls to optionally load the rest of the descriptor content. The DMA will load all optional parts of the descriptor and configure the channel SFR registers before setting CHCTRLA***k***.LLEN = BDCFG.LLEN, CHCTRLA***k***.ENABLE = BDCFG.ENABLE, and CHCTRLA***k***.SWFRC = BDCFG.SWFRC.

If CHCTRLA***k***.ENABLE=1 after the descriptor is loaded, the DMA will perform a block transfer as described in Block Transfer. If both CHCTRLA***k***.ENABLE=1 and CHCTRLA***k***.SWFRC=1, the DMA issues a start trigger to the channel as soon as the descriptor is loaded and the first cell transfer starts. On completion of the block transfer, the DMA clears CHCTRLA***k***.ENABLE.

At this point the DMA channel is ready to load the next descriptor. If CHCTRLA***k***.LLEN=1, the channel requests access to the DMAR port to load the next descriptor at the location specified by the CHNXT***k***.NXT register.

The link list traversal continues until a descriptor sets CHCTRLA***k***.LLEN=0, or if CHCTRLA***k***.LLEN=1 and CHNXT***k***.NXT was loaded with 0xFFFF_FFFF (NULL pointer) from the descriptor BDNXT.NXT location. If CHNXT***k***.NXT is set to the NULL pointer, the DMA will clear CHCTRLAK.LLEN, set the CHINTF***k***.LL flag and end the link-list operation.

26.7.4.1. Linked List Block Descriptor

Each block transfer descriptor in a linked list is a contiguous region of Memory and consists of ten 32-bit words ordered as follows (see *Block Transfer Descriptor*):

- A word, BDNXT, that always gets loaded into the CHNXT*k*.NXT register in order to access the next descriptor.
- A control word, BDCFG, whose bits[9:0] always get loaded into the CHLLCFGSTAT*k*[9:0] register. These bits are the flags that control loading of the subsequent memory locations of the descriptor into the DMA Channel control registers.
  - The RUNSTDBY bit is loaded into the CHCTRLA*k*.RUNSTDBY bit.
  - The bits LLEN, ENABLE, and SWFRC in the control word BDCFG set the CHCTRLA*k*.LLEN, CHCTRLA*k*.ENABLE, and CHCTRLA*k*.SWFRC bits. These are always loaded into the CHCTRLA*k* register at the **end** of the descriptor load.
- A word, BDCTRLB, that gets loaded into the CHCTRLB*k* register if BDCFG.CTRLB is set
- A half-word, BDEVCTRL, that gets loaded into the CHEVCTRL*k* register if BDCFG.EVCTRL is set and the channel supports events (i.e k < 24). If the channel does not support events, the bit BDCFG.EVCTRL is ignored and BDEVCTRL is never loaded.
- A half-word BDCTRLCRC, that gets loaded into the CHCTRLCRC*k* register if BDCFG.CTRLCRC is set.
- A word, BDSSA, that gets loaded into the CHSSA*k* register if BDCFG.SSA is set.
- A word, BDDSA, that gets loaded into the CHDSA*k* if BDCFG.DSA is set.
- A half-word BDSSTRD, that gets loaded into the CHSSTRD*k* register if BDCFG.SSTRD is set.
- A half-word BDDSTRD, that gets loaded into the CHDSTRD*k* register if BDCFG.DSTRD is set.
- A word BDXSIZ, that gets loaded into the CHXSIZ*k* register if BDCFG.XSIZ is set.
- A word BDPDAT, that gets loaded into the CHPDAT*k* register if BDCFG.PDAT is set.
- A word BDCRCDAT, that gets loaded into the CHCRCDAT*k* register if BDCFG.CRCDAT is set.

Upon loading the CHLLCFGSTAT*k* register from the BDCFG word of a descriptor, the DMA_TOP_FSM may overwrite the contents of the DMA Channel SFRs as instructed by the set bits in CHLLCFGSTAT*k*.

Only the first two words of the descriptor are mandatory. All other words are optional. However, the address offset for each of the descriptor words are fixed. Therefore if BDCFG.XSIZ is set, the descriptor must contain at least eight words to honor the address offset for BDXSIZ. However, the remaining two words are never used by the DMA. The user can decide not to reserve them for the descriptor.

A descriptor, regardless of the size, must reside within a 1KB address boundary. The DMA may perform speculative reads on the descriptor content; therefore the first word of the descriptor must reside a minimum of ten words below a memory boundary to avoid speculative reads into unmapped memory regions.

26.7.4.2. Linked List Enable

The CHCTRLA***k***.LLEN bit controls the channels ability to load a linked-list descriptor and configure the channel SFRs. Setting the bit to 1, either by software or through the linked-list descriptor, indicates to the DMA FSM, the DMA shall initiate a descriptor load from the address location provided by CHNXTk.NXT. The channel can only load a descriptor if the channel is not per-forming a block transfer and is not suspended (i.e., CHCTRLA***k***.ENABLE = 0 and CHSTATk.BLKBUSY = 0). The DMA FSM will wait for this condition before loading the descriptor. If the CHNXTk.NXT is set to the NULL pointer, the DMA clears CHCTRLA**k**.LLEN and sets the CHINTFk.LL interrupt flag to indicate completion of the linked-list.

26.7.4.3. Linked List Step Tracing

The channel will set the SD, CC, BH, BC interrupt flags as described in Block Transfer. The user can track progress through a linked-list by monitoring interrupts based on one of these interrupt flags such as the BC. This is useful if the software needs to be kept up-to-date with the progress of the linked-list execution.

26.7.4.4. Linked List Abort

The DMA provides the ability to abort linked-list operation and block transfer operation on the rising edge of a event system trigger. Setting CHEVCTRL***k***.EVAUXIE=1 and CHEVCTRL***k***.EVAUXACT=3 enables linked-list abort triggering. On the rising edge of the event system trigger, the CHCTRLA***k***.LLEN bit is cleared to prevent further linked-list operation. The associated block transfer abort is covered in detail in Abort Trigger Source.

Table 26-4. Block Transfer Descriptor

Address Offset	Name		Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
0x00	BDNXT	7:0	NXT[31:0]							
0x01		15:8								
0x02		23:16								
0x03		31:24								
0x04	BDCFG	7:0	PDAT	XSIZ	DSTRD	SSTRD	DSA	SSA	EVCTRL	CTRLB
0x05		15:8	—	—	—	—	—	—	CTRLDAT	CTRLCRC
0x06		23:16	—	—	—	—	—	—	—	—
0x07		31:24	—	—	—	—	RUNSTDBY	SWFRC	LLEN	ENABLE
0x08	BDCTRLB	7:0	—	RAS[2:0]			—	WAS[2:0]		
0x09		15:8	BYTORD[1:0]		WBOEN	—	—	—	PRI[1:0]	
0x0A		23:16	TRIG[7:0]							
0x0B		31:24	CRCEN	—	CASTEN	—	—	PATEN	PATLEN	PIGNEN
0x0C	BDEVCTRL	7:0	EVOE	EVSTRIE	EVAUXIE	—	EVOMODE[1:0]		EVAUXACT[1:0]	
0x0D		15:8	—	—	—	—	—	—	—	—
0x0E	BDCTRLCR	7:0	CRCRIN	CRCROUT	CRCXOR	—	CRCAPP	CRCMD[2:0]		
0x0F		15:8	—	—	—	—	—	—	—	—
0x10	BDSSA	7:0	SSA[31:0]							
0x11		15:8								
0x12		23:16								
0x13		31:24								
0x14	BDDSA	7:0	DSA[31:0]							
0x15		15:8								
0x16		23:16								
0x17		31:24								

Table 26-4. Block Transfer Descriptor (continued)

Address Offset	Name		Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
0x18	BDSSTRD	7:0	SSTRD[15:0]							
0x19		15:8								
0x1A	BDDSTRD	7:0	DSTRD[15:0]							
0x1B		15:8								
0x1C	BDXSIZ	7:0	CSZ[7:0]							
0x1D		15:8	—	—	—	—	—	—	CSZ[9:8]	
0x1E		23:16	BLKSZ[15:0]							
0x1F		31:24								
0x20	BDPDAT	7:0	PDAT[15:0]							
0x21		15:8								
0x22		23:16	—	—	—	—	—	—	—	—
0x23		31:24	PIGN[7:0]							
0x24	BDCRCDAT	7:0	CRCDAT[31:0]							
0x25		15:8								
0x26		23:16								
0x27		31:24								

26.7.5. Channel Priorities

26.7.5.1. Arbitration for PORT Resources

The DMA provides arbitration between channels requesting access to the DMAW or DMAR ports. Each port arbitrates between channel access requests based on the channel priority group (PRI) and then between the channel natural order.

The DMA Controller module 0 has 3 priorities available, and DMA Controller module 1 has 2 priorities available. The PRI ranges from 1 to n where 'n' specifies the highest priority available. The priority is set by the CHCTRLBk.PRI register field. Arbitration between priority groups takes place after each transaction on a port. If a channel with a priority set to 0 is actively transferring a cell, a higher priority channel can interrupt the cell transfer to gain access to the port. All channels requests with a higher priority group are processed before considering the next lower priority group.

The DMA defines a fixed natural priority between channels within the same priority group. Fixed natural priority defines a channel with a higher channel number as having a higher priority. For example, channel 8 has a higher priority than channel 0 through 6. Arbitration for resources between these channels always takes place at a cell transfer boundary. Once the current channel completes a cell transfer the DMA will arbitrate access to the channel with largest channel number k in the same priority group that has requested access to the port.

26.7.5.2. Priority Interrupts

The DMA generates 'n' interrupts, one for each priority group. For the DMA Controller module 0, the interrupt status register INTSTAT3 reports which of the channels assigned to priority 3 have pending interrupts. Similarly, INTSTAT2 and INTSTAT1 report which channels have pending interrupts for priority two and one.

26.7.5.3. Event System Priority Increase

The auxiliary event system trigger can be configured to increase a channels priority. When CHEVCTRLk.EVAUXIE=1 and CHEVCTRLk.EVAUXACT=1, on each rising edge of the auxiliary event system trigger, the DMA increments the channel priority by one until the channel is at the highest priority. The value read back from CHCTRLBk.PRI reflects the channels current priority level.

26.7.6. Interrupts

Each channel has the ability to generate an interrupt base on the following events:

- Channel detected a start trigger,CHINTFk.SD. Set on the detection of the start trigger which was configured by CHCTRLBk.TRIG and CHCTRLAk.ENABLE=1.
- Channel transfer aborted due to an abort trigger CHINTFk.TA
- Current cell transfer completed (CHINTFk.CC). Asserts each time a cell transfer completes. Also set at the end of the block transfer.
- Block transfer completed (CHINTFk.BC). Asserts at the completion of a block transfer, or when a block transfer aborts due to an abort trigger.
- Block transfer half completed (CHINTFk.BH). Asserts when the channel has written out more than half of the block based on the Block Transfer Size.
- Linked list complete (CHINTFk.LL). Asserts when the DMA encounters a NULL pointer value in register CHNXTk.NXT when attempting to load a descriptor.
- Read bus error encountered (CHINTFk.RDE). Asserts when the system bus matrix issues a client error response to a read request.
- Write bus error encountered (CHINTFk.WRE). Asserts when the system bus matrix issues a client error response to a write request.

The channel interrupts are logically OR’ed together by channel priority to generate 'n' interrupts.

26.7.7. Event System Output

The DMA provides the ability to generate event triggers based on certain states within a channel. Each channel which is configured to use events can also generate a event system trigger. This output is always active high. The following event system output modes are available:

- Single strobe (1 AHB clock wide pulse) at the end of a block transfer
- Single strobe at the end of each cell transfer
- Output asserted high until cell transfer completes writes
  - Assert high on start trigger detection as con-figured by CHCTRLBk.TRIG or CHCTRLAk.SWFRC
  - Output deasserted low when the channel cell transfer completes writing all data to the destination
- Output asserted high until cell transfer completes reads
  - Assert high on start trigger detection as con-figured by CHCTRLBk.TRIG or CHCTRLAk.SWFRC
  - Output deasserted low when the channel cell transfer completes reading all data from the source

26.7.8. DMA CRC Checksum Module

26.7.8.1. Overview

The DMA provides CRC and Checksum functionality on each channel using a single CRC engine. Channels have their own CRC/Checksum controls and can enable or disable functions and modes independent of other channels. The DMA is capable of supporting CRC/Checksum functions for all channels concurrently.

Only 16-bit or 32-bit CRC polynomials are supported with built in support for CRC-16 IBM, CRC-16 CCITT, and CRC-32. The two polynomial register CRCPOLYA and CRCPOLYB allow the user to provide custom 16-bit or 32-bit CRC polynomials which may be shared across any number of channels. The Checksum calculations follows the IP Header Checksum formula.

26.7.8.2. CRC On/Off

The CHCTRLB***k***.CRCEN register control enables or disables CRC/Checksum functions for the channel. When CRCEN=0, all CRC controls and the CRC data register are ignored for the channel.

26.7.8.3. CRC/Checksum Modes

The CHCTRLCRC***k***.CRCMD[2:0] controls the type of CRC/Checksum operation to perform.

26.7.8.3.1. CRC-16 IBM/ANSI

Setting CRCMD to 0 configures the CRC engine in CRC-16 (IBM/ANSI) mode. This CRC is commonly used for USB and ANSI X3.28 protocol.

Polynomial (0x8005):

$x^{16} + x^{15} + x^2 +1$

26.7.8.3.2. CRC-16 CCITT

Setting CRCMD to 1 configures the CRC engine in CRC-16-CCITT mode. This CRC is commonly used for Bluetooth and Secure Digital cards.

Polynomial (0x1021): $x^{16} + x^{12} + x^5 +1$

26.7.8.3.3. Custom 16-bit CRC

When CRCMD=2 the CRC engine uses the 16-bit polynomial from register CRCPOLYA. If CRCMD=3, the CRC engine uses the 16-bit polynomial from register CRCPOLYB. The user is expected to program CRCPOLYA/B with the polynomial as described in Polynomial Registers.

26.7.8.3.4. CRC-32

Setting the CRCMD to 4, configures the CRC engine in CRC-32 mode for this channel. This CRC is commonly used for Ethernet and MPEG2.

Polynomial (0x04C11DB7):

$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x +1$

Note: Typical implementations of CRC-32 require an initialization value of 0xFFFFFFFF as well as setting CRCXOR=1 and CRCROUT=1 for the channel.

For More information, refer to:

- Reflected Output
- XOR Output
- CRC Data

26.7.8.3.5. Custom 32-bit CRC

When CRCMD=5 the CRC engine uses the 32-bit polynomial from register CRCPOLYA. If CRCMD=6, the CRC engine uses the 32-bit polynomial in register CRCPOLYB. The user is expected to program CRCPOLYA/B with the polynomial as described in Polynomial Registers.

26.7.8.3.6. IP Header Checksum

When CRCMD is set to 7 for the channel, the CRC engine implements an Checksum calculation. See Checksum Calculations for details on how the calculation is done.

26.7.8.4. Polynomial Registers

The DMA provides two polynomial registers, CRCPO-LYA and CRCPOLYB, which are shared between the channels. These registers allow users to program 2 unique CRC polynomials which any number of channels can use.

For example, the user can program POLYA with a 32-bit polynomial such as CRC-32 Castagnoli (0x1EDC6F41) and configure channel 1 and 3 to use CRCPOLYA in 32-bit CRC mode (CHCTRLCRC1.CRCMD=5 and CHCTRLCRC3.CRCMD=5). At the same time, the user can program POLYB with a unique 16-bit polynomial to be used by channel 2 (CHCTRLCRC2.CRCMD=3).

26.7.8.4.1. CRC Polynomial Configuration Examples

Example of 32-Bit Polynomial CRC Setup shows and example CRC configuration with a polynomial size of 32. For each term (xn) of the polynomial, a corresponding bit in CRCPOLYA/POLYB is set to ‘1’. *Example of 16-Bit Polynomial CRC Setup* shows an example of a 16-bit polynomial, CRC-16-DNP.

26.7.8.5. Append Mode

The DMA provides a CRC append mode, CHCTRLCRC***k***.CRCAPP, which appends the CRC or Checksum value to the end of a block transfer.

When CRCAPP=0, the DMA reads the data from the source, passes through the CRC module and writes the data to the destination. The calculated CRC or checksum is left in the CRCDAT register at the end of the block transfer.

Setting CRCAPP=1 enables the append mode. The DMA performs the same function as described for CRCAPP=0. However, at the end of the block transfer the DMA appends the CHCRCDAT***k***.CRCDATA content to the destination by writing up to 4 bytes of data depending on the CRCMD setting. The BYTORDER and WBOEN settings do not affect the CRC value being written to the destination. Only the CHCTRLCRC***k***.CRCXOR and CHCTRLCRC***k***.CRCROUT register affect how the CHCRCDAT***k***.CRCDATA value is written out.

26.7.8.6. Reflected Input

CHCTRLCRC***k***.CRCRIN defines the order in which bits within a byte are feed into the CRC module. When CRCRIN=0, bits remain in source order. When CRCRIN=1, for each byte, the bit order is reflected before being processed by the CRC engine. A reflected byte is defined as follows:

Let ***j*** represent a number from 0 to 7, then Reflected Byte[***j***] = Original Byte[7-***j***]

26.7.8.7. Reflected Output

CHCTRLCRC***k***.CRCROUT defines the order in which the CRC bits are read back or written out in append mode. When CRCROUT = 0 bits remain in native order with the MSB of the CRC located at the bit 31. When CRCROUT = 1 the bit order is reflected. This function can be combined with CHCTRL-CRC***k***.CRCXOR to reverse and invert the CRC value. Reflected bit order is defined as follows:

Let ***j*** represent a number from 0 to 31 for CRC-32 or 0 to 15 for CRC-16, then

Reflected CRC32[***j***] = Original CRC32[31-***j***] Reflected CRC16[***j***] = Original CRC16[15-***j***]

26.7.8.8. XOR Output

CHCTRLCRC***k***.CRCXOR determines if the CRC value is XORed with all 1's when read from CHCRCDAT***k***. When CRCXOR = 0 the CRC value is XORed with all 0's.

When CRCXOR = 1 the CRC value read out of CHCRCDAT***k*** is defined as:

CRC16 Read Value = CRC16 Result ^ 0xFFFF CRC32 Read Value = CRC32 Result ^ 0xFFFF_FFFF

26.7.8.9. CRC Data

The CHCRCDAT***k***.CRCDAT provides the initialization value for both CRC and checksum calculations before the block transfer starts. At the end of the block transfer the result can be read from CRCDAT. If the calculations are to take place over multiple block transfers, provide a initial value before the first block transfer. Do not update the CRCDAT register between block transfers. Once all the block transfers are completed, the CRCDAT will contain the resulting value.

26.7.8.10. Checksum Calculations

An IP header checksum is a simple checksum used to provide basic protection against bit corruption during transmission. The header is calculated by dividing the data stream into half-words (16-bits) and adding them together. Any overflow is also added back into the sum. The result is the one's complement of the calculated sum. The following code shows the details on how a check-sum is calculated.

The CRCDAT value is loaded with the expected check sum value prior to the start of the block transfer, at the end of the block transfer the CRCDAT will read back 0xFFFF if successful. Note, that checksum in CRC-DAT is the non-inverted value if CRCXOR=0. To read back the 1s-complement value of the checksum, set CRCXOR=1 prior to performing the calculation.

To perform IP Header checksums, configure the channel as follows:

- Configure the block transfer options to read the IP Header in memory.
- Set the CHCTRLB***k***.CASTEN=1.
- Configure CHCTRLB***k***.CRCEN to enable CRC/Checksum for the channel.
- Set CHCTRLCRC***k***.CRCMD to 7 to select IP Header Checksum mode.
- Set CHCRCDAT***k***.CRCDAT to 0x0000.
- Set CHCTRLCRC***k***.CRCXOR=1.
- Set CHCTRLA***k***.SWFRC to 1 to initiate the transfer.
- When the transfer completes CRCDAT register reports the IP Header checksum.

```
Consider an IP header frame byte stream
(MSBit first, little endian ordering):
0F 2B 3C 4D 5E 6F 78 9A
0 1 2 3 4 5 6 7 (address)
The checksum is calculated by splitting
the stream into 16-bit words and summing
them as follows:
2B0F
+4D3C
----
784B
+6F5E
----
E7A9
+9A78
----
18221 <- Note the overflow!
0001 <- Overflow added
----
8222 <- MSB added back into sum
8222 is the sum of the IP Header. The
checksum is the ones-complement of the
sum: 7DDD = ~8222
```

26.7.8.11. CRC Calculations

The CRC engine is a parallel CRC circuit.

The engine implements a programmable internal feed-back LSFR in parallel fashion. Since internal feedback is provided, zero padding at the end of a data block is not required to obtain the final CRC. The engine operates either on 8-bits, 16-bits, or 32-bits of data per clock cycle and supports either 16-bit or 32-bit polynomials. Any combination of 8/16/32 bit word length and 16/32 bit polynomial are supported.

Data entering the block will be reordered such that the CRC checksum calculated for a block is the same regardless of read address mode (byte mode, half-word mode, or word mode). This reordering is completely transparent to software and is independent from the BYTORD setting chosen by the user. Because the system is little endian, in order to get the same CRC result, read halfwords will be byte swapped, and read words will have bytes in reverse order.

Table 26-5. Example of 32-Bit Polynomial CRC Setup

Bit Name	Bit Value				Description
CHCTRL-CRC***k***.CRCMD	3'b110 or 3'b101				32-bit polynomial length using CRCPOLYA or CRCPOLYB registers
CRCPOLYA or POLYB	0001 0110	1110 1111	1101 0100	1100 000-	CRC polynomial: $x^{32}+x^{28}+x^{27}+x^{26}+x^{25}+x^{23}+x^{22}+x^{20}+x^{19}+x^{18}+x^{14}+x^{13}+x^{11}+x^{10}+x^{9}+x^{8}+x^{6}+1$

Table 26-6. Example of 16-Bit Polynomial CRC Setup

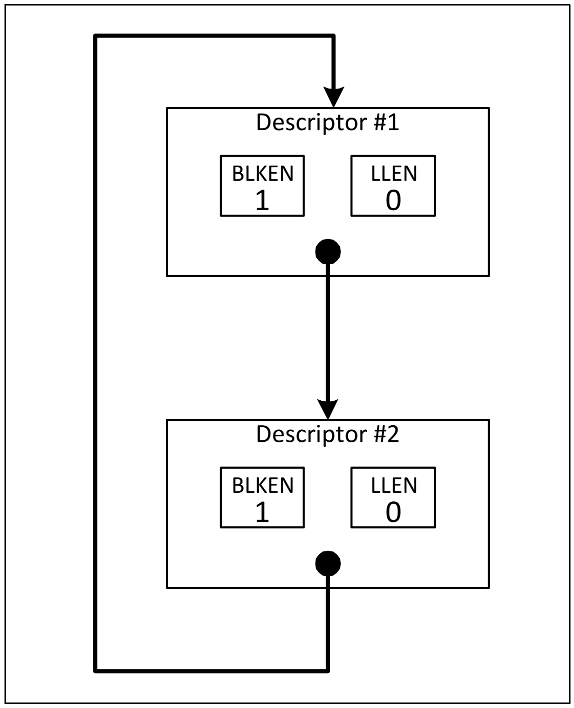
Bit Name	Bit Value				Description
CHCTRL-CRC***k***.CRCMD	3'b011 or 3'b010				16-bit polynomial length using CRCPOLYA or CRCPOLYB registers
CRCPOLYA or POLYB	0000 0011	0000 1101	0000 0110	0000 010-	CRC polynomial: $x^{16}+x^{13}+x^{12}+x^{11}+x^{10}+x^{8}+x^{6}+x^{5}+x^{2}+1$

26.7.9. Examples

26.7.9.1. Ping Pong Linked List

Consider a scenario where two data buffers are created in system memory. There is a linked-list descriptors associated with each of the data buffer. Each descriptor defines block transfer of the associated buffer to the same communication peripheral. The following figure shows the linked-list structure for such a scenario. The descriptors are linked in circular fashion. The BDCFG.LLEN bit in both descriptors are set to zero. In this configuration, software can update one of the buffers while the DMA is transferring the other buffer to the peripheral.

Figure 26-9. Ping-Pong Linked-List Example



Initially, software programs the address of Descriptor #1 into the CHNXT*k*.NXT register, assembles the data buffer, and then sets the CHCTRLA*k*.LLEN bit to 1. The DMA loads the descriptor, configures the channel SFRs to perform the block transfer, and finally sets the CHCTRLA*k*.ENABLE=1 and CHCTRLA*k*.LLEN=0 as configured in Descriptor #1. The channel is now read to perform block transfer #1. Note, after the Descriptor #1 loads, the CHNXT*k*.NXT register contains the address location of Descriptor #2.

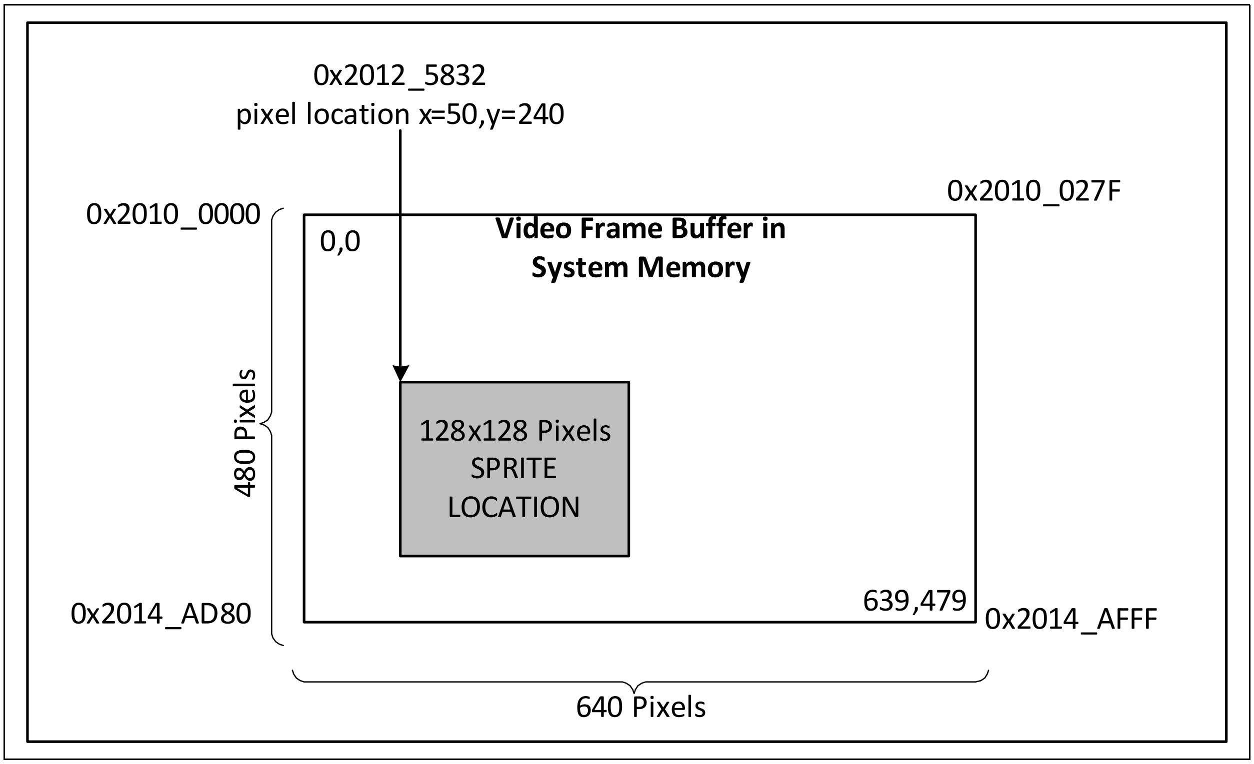
While the DMA is processing the block transfer for Descriptor #1, software works on assembling the data for Descriptor #2. On completion of data assembly, software writes the CHCTRLA*k*.LLEN bit to 1. If the DMA has already completed block transfer #1, it loads Descriptor #2 immediately and prepares for block transfer #2. If block transfer #1 has not completed, the DMA waits until the block transfer completes before loading Descriptor #2. After the Descriptor #2 loads, the CHNXT*k*.NXT register contains the address location of Descriptor #1. Since LLEN=0 after loading Descriptor #2, the DMA will not load Descriptor #1. Software will write LLEN=1 when data for Descriptor #1 is ready.

This linked-list configuration allows for a “ping-pong” usage of buffers with a simple program model. Software only needs to write to CHCTRLA*k*.LLEN to notify the DMA there is new data to send and track when a block transfer has completed to know if a buffer is free to update.

26.7.9.2. Linked-List with Cell Striding Example

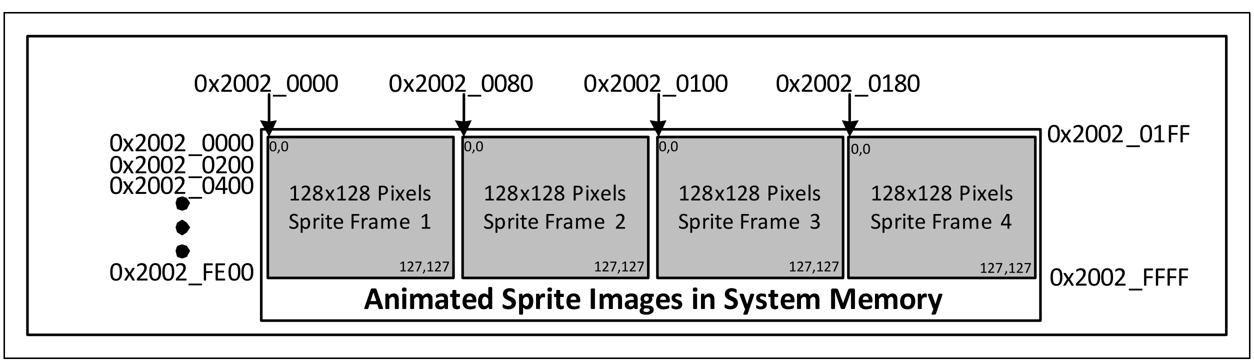
Consider a simple video display scenario where a video frame buffer exists in memory at address location 0x2010_0000 as shown in the following figure.

Figure 26-10. Video Frame Buffer



The frame buffer is 640x480 pixels and a pixel is represented by a byte of data. The user would like to update the frame buffer with an animated sprite located at (50,240) or address 0x2012_5832 in the video frame buffer. The animated sprite frames are located in memory at 0x2002_0000. There are four frames in the animation and each sprite frame is 128x128 pixels (1 pixel/byte). The following figure shows each animation frame is stored in memory. The system display controller is setup to display the video frame buffer. The display controller is capable of generating a DMA request at the start of a vertical sync.

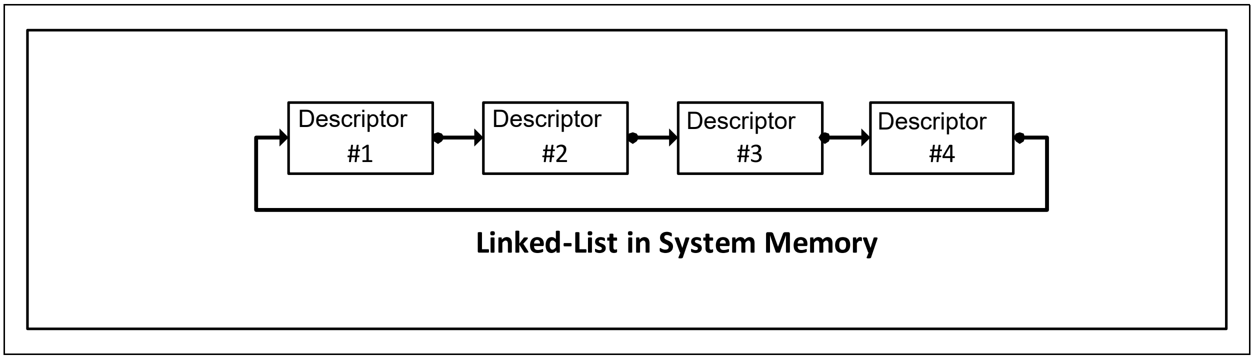
Figure 26-11. Sprite Frame



To animate the sprite, a channel on the DMA will be configured to update the video frame buffer at every vertical sync. On each vertical sync, the next frame of the animation is copied into the video frame buffer at location (50,240).

Software sets up a linked-list with four descriptors, one for each sprite frame. The linked-list is configured as a circular chain, so the fourth descriptor points back to the first descriptor as shown in the following figure. This will create a looped animation of the sprite.

Figure 26-12. Linked List with Cell Stride



For the Source Start Address, each descriptor points to a different frame of the animation.

Descriptor #1 (Sprite Frame 1):

```
BDSSA = 0x2002_0000,
        BDCFG.SSA=1
```

Descriptor #2 (Sprite Frame 2):

```
BDSSA = 0x2002_0080,
        BDCFG.SSA=1
```

Descriptor #3 (Sprite Frame 3):

```
BDSSA = 0x2002_0100,
        BDCFG.SSA=1
```

Descriptor #4 (Sprite Frame 4):

```
BDSSA = 0x2002_0180,
        BDCFG.SSA=1
```

The BCFG.ENABLE and BCFG.LLEN are both set to one for all four descriptors. Note, the only updates required for each successive image transfer is the Source Start Address, therefore this is the only optional information provided in each descriptor.

The DMA channel is configured as follows:

- The address location for Descriptor #1 is loaded into the next descriptor register

  ```
  CHNXTk.NXT =
          &Descriptor #1
  ```

- Destination Start Address is set to 0x2012_5832

  ```
  CHDSAk.DSA =
          0x2012_5832
  ```

- The Block Transfer Size is set to the size of a single sprite frame, in this case 128x128 pixels = 16384 bytes

  ```
  CHXSIZk.BLKSZ =
          16384
  ```

- The Cell Transfer Size is set to a 128 pixels = 128 bytes. The Source Cell Stride Size is set to 128x4 pixels = 512. This tells the DMA that each cell (128 pixels) is located 512 bytes apart. The Destination Cell Stride Size is set to 640 pixels = 640 bytes since that is the size of the video frame X dimension.

```
CHXSIZk.CSZ =
        128
```

```
CHSSTRDk.SSTRD =
        512-128
```

```
CHDSTRDk.DSTRD =
        640-128
```

- Since this is a memory to memory data copy, the address sequence mode is set to Increment Address / Auto

`CHCTRLBk.RAS='b010`

`CHCTRLBk.WAS='b010`

- The start trigger event is configured to use the vertical sync

```
CHCTRLBk.TRIG =
        Display Controller Vertical Sync
```

- Since the entire sprite frame needs to be copied into the video frame buffer during the vertical sync, the user sets Cell Auto Start Enable of Ensuing Transfers

```
CHCTRLBk.CASTEN =
        1
```

- All other channel features are disabled

Once the channel is configured, the user sets the CHCTRLA*k*.LLEN bit to 1.

At this point the DMA loads Descriptor #1 and updates the CHSSA*k*.SSA and CHNXT*k*.NXT registers before setting CHCTRLA*k*.ENABLE and CHCTRLA*k*.LLEN. On the next vertical sync trigger, the DMA transfers the sprite frame #1 into the video frame buffer. The DMA clears CHCTRLA*k*.ENABLE at the completion of the block transfer. Since LLEN=1, the next descriptor loads and SSA updates to the start of sprite frame #2, NXT is loaded with a pointer to Descriptor 3. Note, if a vertical sync trigger occurs while the DMA is loading a new descriptor, the trigger event is ignored since CHCTRLA*k*.ENABLE=0 at this time. The DMA configures CHCTRLA*k*.ENABLE=1 and LLEN=1 once the SFRs are configured. On the next vertical sync trigger, the DMA transfers sprite frame #2 into the video frame buffer. Again, on completion of the block transfers, the DMA loads the next descriptor. Each consecutive descriptor and block transfer are completed in a similar manner. Since the linked-list is circular, the animation loop continues until software sets CHCTRLA*k*.LLEN=0.

26.8. Module Description

26.8.1. Enable

If the user disables the DMA module by clearing the CTRLA.ENABLE bit, care must be taken to ensure that the DMAR and DMAW bus hosts are not frozen in a state that would lock up the Bus Matrix. To this end the current DMAR and DMAW bus transactions shall be completed before honoring the disable request.

Any channel that has its BLKBUSY bit asserted will enter a resumable suspended state. If the user writes to a suspended channel's SFR register, the channel will reset. See Block Enable for details on a channel reset. The CHCTRLAk.ENABLE bit is also cleared on a channel reset. If the CHCTRLAk.ENABLE bit is set to 1 and a channel is suspended, activity on the channel resumes from the point of suspension.

26.8.2. Block Enable

If the user disables a channel by clearing the CHCTRLA *k*.ENABLE bit. No new bus requests are permitted for the channel as long as CHCTRL*k*.ENABLE=0.

If CHCTRL*k*.ENABLE is set to 0, while CHSTAT*k*.BLKBUSY=1, the channel enters a suspended state. If CHCTRL*k*.ENABLE is set to 1, a suspended channel resumes activity.

If any of the channel's SFR registers are updated while suspended, the channel will reset and flush any FIFO data. Reset of a channel does not affect other channels. Reset consists of resetting the channel FSM into an idle state and clearing the channel's status bits. All counters for the channel are reset to the start of a block transfer. Clear the CHSTAT*k*.BLKBUSY bit at the end of the reset sequence. This mechanism serves as an indication to software that the channel reset has completed.

26.8.3. Descriptor Loads

The DMA issues a 2-word burst request to read in BDNXT and BDCFG. All other words are optional and can be read as single words or by burst requests.

26.8.4. Interrupts

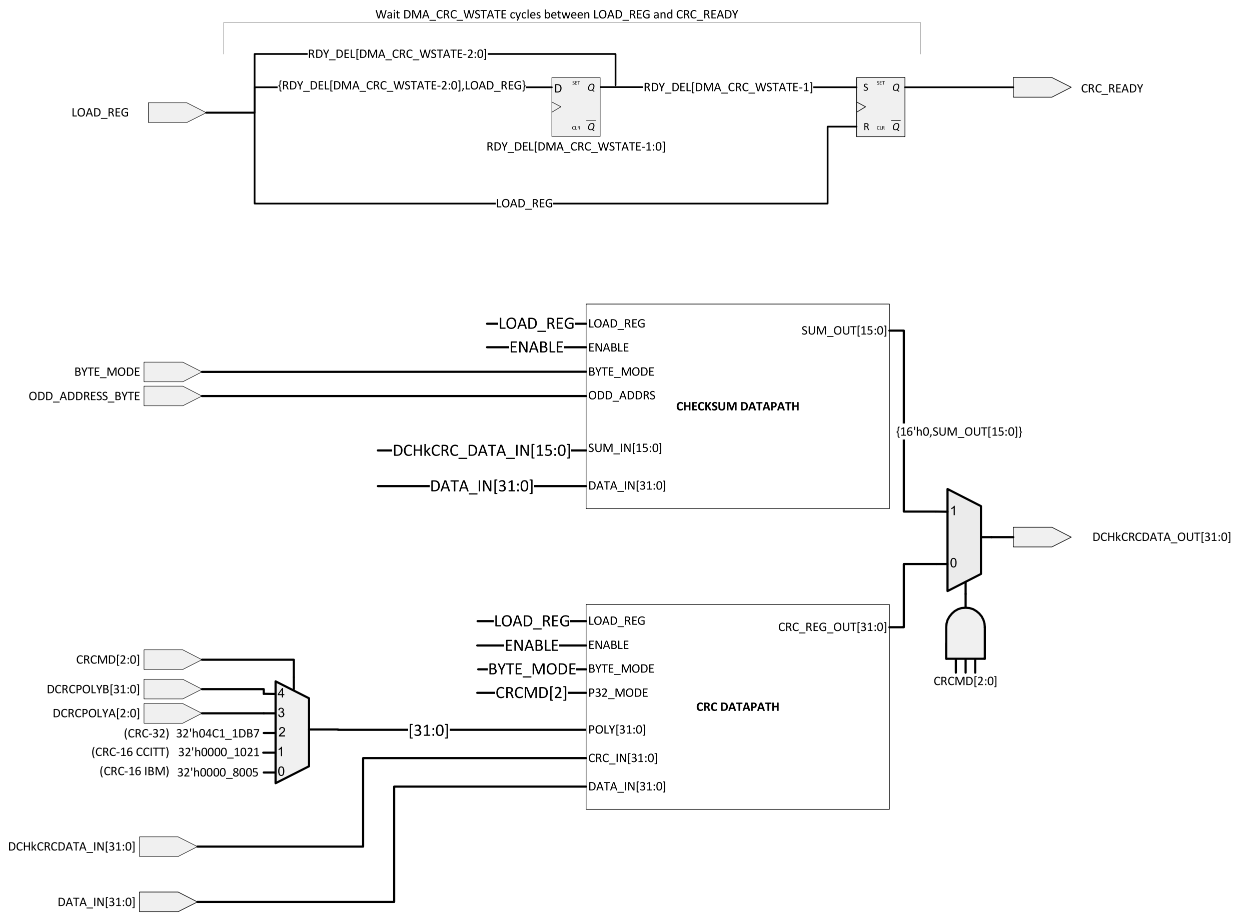
Interrupt generation by the DMA occurs at a channel level. Within each channel there is a range of bits that can set an interrupt flag. These are:

- **Start Detected (SD):** Stated Detect interrupt is set when a start trigger is detected for the channel
- **Block Transfer Abort (TA):** When the channel receives an abort trigger as described in Abort Trigger Source, the TA bit is set. The CHCTRLA*k*.ENABLE bit is cleared and the FSM advances to a block transfer complete state.
- **Cell Transfer Complete (CC):** The CC flag is set when a cell transfer is complete. A cell transfer completes each time CHXSIZ*k*.CSZ bytes have been written to the destination. Also, set the CC bit when BC is set.
- **Block Transfer Complete (BC):** Set the BC bit when a block transfer completes or is aborted. A block transfer completes when CHXSIZ*k*.BLKSZ bytes have been written to the destination.
- **Block Transfer Half Complete (BH):** Set the BH bit on the first clock cycle when CHSTBC*k*.BBTC ≥ CHXSIZ*k*.BLKSZ / 2. The user may clear BH any time after the DMA sets the bit. The DMA will not set BH again for the duration of the block transfer.
- **Linked List Complete (LL):** Set the LL bit when LLEN=1 and the CHNXT*k*.NXT register value is 0xFFFF_FFFF (NULL). The DMA will also clear LLEN under this condition.
- **Read Error (RDE)**
- **Write Error (WRE)**

26.8.5. CRC and Checksum Engine

The CRC and IP Header Checksum datapaths are designed to process up to 64-bits of data in a single clock. The following figure shows an overview of the CRC/Checksum engine and data paths for 32-bits.

Figure 26-13. CRC Engine

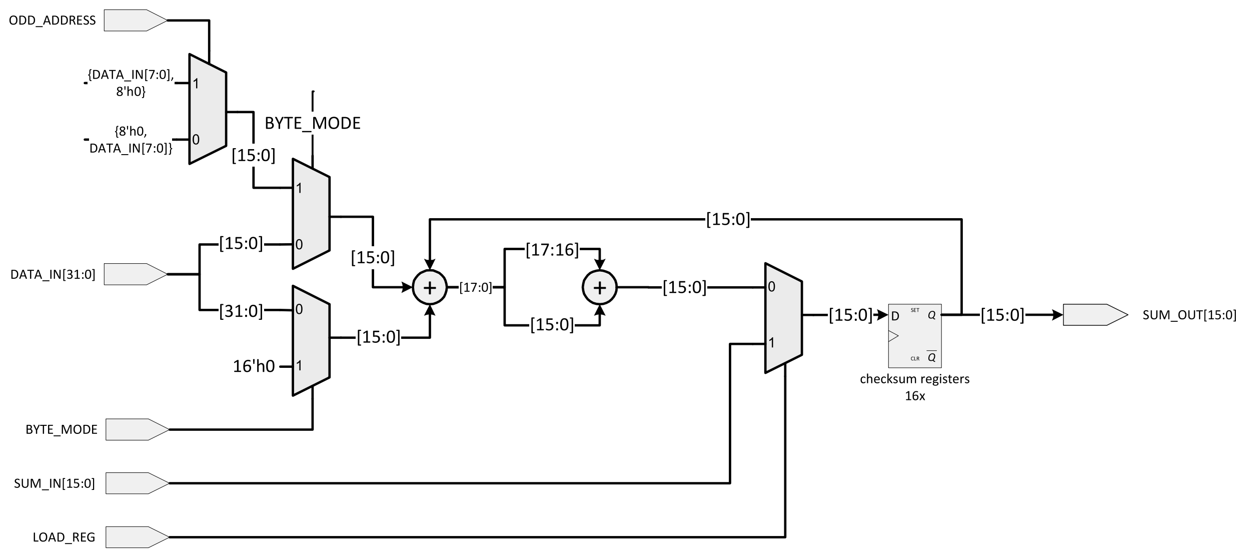


Data entering the block will be reordered such that the CRC checksum calculated for a block is the same regardless of read address mode (byte mode, half-word mode, or word mode). This reordering is completely transparent to software and is independent from the BYTORD setting chosen by the user. Because the system is little endian, in order to get the same CRC result, read halfwords will be byte swapped, and read words will have bytes in reverse order.

26.8.5.1. IP Header Checksum Datapath

The following figure shows a representation of the checksum datapath. The datapath processes a word each cycle. Each time a channel is granted bus access, the lower 16 bits of the CRCDAT register is loaded into the checksum register using signal LOAD_REG. On loss of arbitration or the end of cell transfer, save the checksum result SUM_OUT[15:0] into the lower 16 bits of CRCDAT. Repeat the process each time the channel gains arbitration and completes a cell transfer.

Figure 26-14. IP Header Checksum Datapath



In word mode, the datapath splits the 32-bit word into two 16-bit words, sums them with the existing sum result (either previously loaded or calculated from the last data word). If the result is greater than 16-bits, the upper 16-bit is summed with the lower 16-bits and captured in the sum result register.

In half-word mode, half-words are added to the previous sum result. If the result is larger than 16-bits, the upper 16-bits are summed with the lower 16-bits and stored into the sum register.

In byte mode, bytes read from even address locations extended to 16-bits with 0's and added to the previous sum result. Again, if the value is larger than 16-bits, the upper 16-bits are summed with the lower 16-bits and stored into the sum register. For bytes read from odd addresses, the byte is shifted left 8-bits (multiply by 2^8) and summed with the previous sum result. If the resulting sum is larger than 16-bits, the upper 16-bits are summed with the lower 16-bits and stored in the sum register.

At the end of a cell transfer the checksum is stored back into CRCDAT[15:0].

26.9. Register Summary

For descriptions and definitions of both Register and bitfield properties, refer to Register Properties.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRLA	31:24								
		23:16								
		15:8								
		7:0							ENABLE	
0x04	CTRLB	31:24								
		23:16							QOS2[1:0]	
		15:8							QOS1[1:0]	
		7:0							QOS0[1:0]	
0x08	DBGCTRL	31:24								
		23:16								
		15:8								
		7:0								DBGRUN
0x0C	CRCPOLYA	31:24	POLYA[31:24]							
		23:16	POLYA[23:16]							
		15:8	POLYA[15:8]							
		7:0	POLYA[7:0]							
0x10	CRCPOLYB	31:24	POLYB[31:24]							
		23:16	POLYB[23:16]							
		15:8	POLYB[15:8]							
		7:0	POLYB[7:0]							
0x14 ... 0x17	Reserved									
0x18	INTSTAT3	31:24								
		23:16								
		15:8								
		7:0	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
0x1C	INTSTAT2	31:24								
		23:16								
		15:8								
		7:0	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
0x20	INTSTAT1	31:24								
		23:16								
		15:8								
		7:0	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0

26.9.1. DMA Control A Register

Name: CTRLA
Offset: 0x00
Reset: 0x00000000
Property: PAC Write-Protection

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

Bit 1 – ENABLE DMA Enable

If the DMA is enabled and this bit is written to zero, the DMA may have outstanding bus transactions that need to complete before it can completely disable.

Value	Description
0	DMA module and channels are disabled
1	DMA module and channels are enabled

26.9.2. DMA Control B Register

Name: CTRLB
Offset: 0x04
Reset: 0x00000000
Property: PAC Write-Protection

Table 26-7. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
							QOS2[1:0]	
Access							R/W	R/W
Reset							0	0

Bit	15	14	13	12	11	10	9	8
							QOS1[1:0]	
Access							R/W	R/W
Reset							0	0

Bit	7	6	5	4	3	2	1	0
							QOS0[1:0]	
Access							R/W	R/W
Reset							0	0

Bits 0:1, 8:9, 16:17 – QOSn Priority Group n QOS control [n=1..3]
Sets the Quality of service level for Channel Priority Group n. Setting this value affects arbitration within the device bus fabric. This value does not affect arbitration within the DMA.
Note: DMA0 has 3 priority levels, DMA1 has 2 priority levels.

Table 26-8. Generator Clock Source Selection

Value	Name	Description
0x0	LEVEL_0	QoS level is 0 (lowest)
0x1	LEVEL_1	QoS level is 1 (lower)
0x2	LEVEL_2	QoS level is 2 (medium)
0x3	LEVEL_3	QoS level is 3 (high)

26.9.3. Debug Control Register

Name: DBGCTRL
Offset: 0x08
Reset: 0x00000000
Property: PAC Write-Protection

Table 26-9. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

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

Bit 0 – DBGRUN Debug Run

This bit controls the DMA functionality when the CPU is halted by an external debugger.

Note: The user should be certain to set this field if he wishes the DMA to operate normally during debug.

Value	Description
0	DMA halts the operation during debug. All outstanding bus requests complete before halting.
1	DMA continues the normal operation during debug.

26.9.4. DMA CRC Polynomial A Register

Name: CRCPOLYA
Offset: 0x0C
Reset: 0x00000000
Property: PAC Write-Protection. Enable Write Protected

Table 26-10. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	POLYA[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
	POLYA[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
	POLYA[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
	POLYA[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 – POLYA[31:0] CRC Polynomial Coefficients Bits

The most significant polynomial term is always included in the CRC polynomial and is not required to be programmed in this register. This includes x32 for a 32-bit polynomial and x16 for a 16-bit polynomial.

For example, set this register to 0x741B8CD7 for the following 32-bit polynomial:

$p(x) = x^{32} + x^{30} + x^{29} + x^{28} + x^{26} + x^{20} + x^{19} + x^{17} + x^{16} + x^{15} + x^{11} + x^{10} + x^{7} + x^{6} + x^{4} + x^{2} + x + 1$

Set this register to 0x00000589 for the following 16-bit polynomial $p(x) = x^{16} + x^{10} + x^{8} + x^{7} + x^{3} + 1$

Value	Description
0	Exclude $x^n$from the CRC polynomial.
1	Include the term x^n, where n is the bit location, in the CRC polynomial.

26.9.5. DMA CRC Polynomial B Register

Name: CRCPOLYB
Offset: 0x10
Reset: 0x00000000
Property: PAC Write-Protection, Enable Write Protected

Table 26-11. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	POLYB[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
	POLYB[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
	POLYB[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
	POLYB[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 – POLYB[31:0] CRC Polynomial Coefficients B Bits

The most significant polynomial term is always included in the CRC polynomial and is not required to be programmed in this register. This includes x32 for a 32-bit polynomial and x16 for a 16-bit polynomial.

For example, set this register to 0x741B8CD7 for the following 32-bit polynomial:

$p(x) = x^{32} + x^{30} + x^{29} + x^{28} + x^{26} + x^{20} + x^{19} + x^{17} + x^{16} + x^{15} + x^{11} + x^{10} + x^{7} + x^{6} + x^{4} + x^{2} + x + 1$

Set this register to 0x00000589 for the following 16-bit polynomial $p(x) = x^{16} + x^{10} + x^{8} + x^{7} + x^{3} + 1$

Value	Description
0	Exclude $x^n$from the CRC polynomial.
1	Include the term x^n, where n is the bit location, in the CRC polynomial.

26.9.6. DMA Interrupt Priority 3 Status Register

Name: INTSTAT3
Offset: 0x18
Reset: 0x00000000
Property: PAC Write-Protection

Table 26-12. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
Access	R/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 – CHx DMA Channel Active Interrupt at Priority 3
Note: This register is only available for DMA Controller module 0.

26.9.7. DMA Interrupt Priority 2 Status Register

Name: INTSTAT2
Offset: 0x1C
Reset: 0x00000000
Property: PAC Write-Protection

Table 26-13. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
Access	R/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 – CHx DMA Channel Active Interrupt at Priority 2
Note: DMA0 has 8 channels, DMA1 has 4 channels.

26.9.8. DMA Interrupt Priority 1 Status Register

Name: INTSTAT1
Offset: 0x20
Reset: 0x00000000
Property: PAC Write-Protection

Table 26-14. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
Access	R/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 – CHx DMA Channel Active Interrupt at Priority 1
Note: DMA0 has 8 channels, DMA1 has 4 channels.

26.10. Channelk Register Summary, k = 0,1,...,11

For descriptions and definitions of both Register and bitfield properties, refer to Register Properties.

Note: Each of the 12 channels has 19 channel-specific registers: CHCTRLAk through CHSTATk, k=0,1,...,11. Each set of channel-specific registers is offset by 0x50 from the previous set:
CHCTRLA*k* offset = 0x50+k*0x50

CHCTRLB*k* offset = 0x54+k*0x50

CHEVCTRL*k* offset = 0x58+k*0x50

.

.

.

CHSTAT*k* offset = 0x98+k*0x50

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00 ... 0x4F	Reserved									
0x50	CHCTRLA0	31:24								RUNSTDBY
		23:16								SWFRC
		15:8								LLEN
		7:0								ENABLE
0x54	CHCTRLB0	31:24	CRCEN		CASTEN			PATEN	PATLEN	PIGNEN
		23:16	TRIG[7:0]							
		15:8	BYTORD[1:0]		WBOEN				PR[1:0]	
		7:0		RAS[2:0]				WAS[2:0]		
0x58	CHEVCTRL0	31:24								
		23:16								
		15:8								
		7:0	EVOE	EVSTRIE	EVAUXIE		EVOMODE[1:0]		EVAUXACT[1:0]	
0x5C	CHINTENCLR0	31:24								
		23:16								
		15:8								
		7:0			LL	BH	BC	CC	TA	SD
0x60	CHINTENSET0	31:24								
		23:16								
		15:8								
		7:0			LL	BH	BC	CC	TA	SD
0x60	CHINTF0	31:24								
		23:16						RDE	WRE	
		15:8								
		7:0			WRELL	BH	BC	CC	TA	SD
0x64 ... 0x67	Reserved									
0x68	CHSSA0	31:24	SSA[31:24]							
		23:16	SSA[23:16]							
		15:8	SSA[15:8]							
		7:0	SSA[7:0]							
0x6C	CHDSA0	31:24	DSA[31:24]							
		23:16	DSA[23:16]							
		15:8	DSA[15:8]							
		7:0	DSA[7:0]							
0x70	CHSSTRD0	31:24								
		23:16								
		15:8	SSTRD[15:8]							
		7:0	SSTRD[7:0]							

Channelk Register Summary, k = 0,1,...,11 (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x74	CHDSTRD0	31:24								
		23:16								
		15:8	DSTRD[15:8]							
		7:0	DSTRD[7:0]							
0x78	CHXSIZ0	31:24	BLKSZ[15:8]							
		23:16	BLKSZ[7:0]							
		15:8							CSZ[9:8]	
		7:0	CSZ[7:0]							
0x7C	CHPDAT0	31:24	PIGN[7:0]							
		23:16								
		15:8	PDAT[15:8]							
		7:0	PDAT[7:0]							
0x80	CHCTRLCRC0	31:24								
		23:16								
		15:8								
		7:0	CRCRIN	CRCROUT	CRCXOR		CRCAPP	CRCMD[2:0]		
0x84	CHCRCDAT0	31:24	CRCDAT[31:24]							
		23:16	CRCDAT[23:16]							
		15:8	CRCDAT[15:8]							
		7:0	CRCDAT[7:0]							
0x88	CHNXT0	31:24	NXT[31:24]							
		23:16	NXT[23:16]							
		15:8	NXT[15:8]							
		7:0	NXT[7:0]							
0x8C	CHLLCFGSTAT0	31:24								
		23:16								
		15:8							CRCDAT	CTRLCRC
		7:0	PDAT	XSIZ	DSTRD	SSTRD	DSA	SSA	EVCTRL	CTRLB
0x90	CHSTATBC0	31:24								
		23:16								BBTC[16]
		15:8	BBTC[15:8]							
		7:0	BBTC[7:0]							
0x94	CHSTATCC0	31:24								
		23:16								
		15:8						CBTC[10:8]		
		7:0	CBTC[7:0]							
0x98	CHSTAT0	31:24								
		23:16								
		15:8								
		7:0						DREAD	CELLBUSY	BLKBUSY
0x9C ... 0x9F	Reserved									
0xA0	CHCTRLA1	31:24								RUNSTDBY
		23:16								SWFRC
		15:8								LLEN
		7:0								ENABLE
0xA4	CHCTRLB1	31:24	CRCEN		CASTEN			PATEN	PATLEN	PIGNEN
		23:16	TRIG[7:0]							
		15:8	BYTORD[1:0]		WBOEN				PR[1:0]	
		7:0		RAS[2:0]				WAS[2:0]		
0xA8	CHEVCTRL1	31:24								
		23:16								
		15:8								
		7:0	EVOE	EVSTRIE	EVAUXIE		EVOMODE[1:0]		EVAUXACT[1:0]	
0xAC	CHINTENCLR1	31:24								
		23:16								
		15:8								
		7:0			LL	BH	BC	CC	TA	SD

Channelk Register Summary, k = 0,1,...,11 (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0xB0	CHINTENSET1	31:24								
		23:16								
		15:8								
		7:0			LL	BH	BC	CC	TA	SD
0xB0	CHINTF1	31:24								
		23:16						RDE	WRE	
		15:8								
		7:0			WRELL	BH	BC	CC	TA	SD
0xB4 ... 0xB7	Reserved									
0xB8	CHSSA1	31:24	SSA[31:24]							
		23:16	SSA[23:16]							
		15:8	SSA[15:8]							
		7:0	SSA[7:0]							
0xBC	CHDSA1	31:24	DSA[31:24]							
		23:16	DSA[23:16]							
		15:8	DSA[15:8]							
		7:0	DSA[7:0]							
0xC0	CHSSTRD1	31:24								
		23:16								
		15:8	SSTRD[15:8]							
		7:0	SSTRD[7:0]							
0xC4	CHDSTRD1	31:24								
		23:16								
		15:8	DSTRD[15:8]							
		7:0	DSTRD[7:0]							
0xC8	CHXSIZ1	31:24	BLKSZ[15:8]							
		23:16	BLKSZ[7:0]							
		15:8							CSZ[9:8]	
		7:0	CSZ[7:0]							
0xCC	CHPDAT1	31:24	PIGN[7:0]							
		23:16								
		15:8	PDAT[15:8]							
		7:0	PDAT[7:0]							
0xD0	CHCTRLCRC1	31:24								
		23:16								
		15:8								
		7:0	CRCRIN	CRCROUT	CRCXOR		CRCAPP	CRCMD[2:0]		
0xD4	CHCRCDAT1	31:24	CRCDAT[31:24]							
		23:16	CRCDAT[23:16]							
		15:8	CRCDAT[15:8]							
		7:0	CRCDAT[7:0]							
0xD8	CHNXT1	31:24	NXT[31:24]							
		23:16	NXT[23:16]							
		15:8	NXT[15:8]							
		7:0	NXT[7:0]							
0xDC	CHLLCFGSTAT1	31:24								
		23:16								
		15:8							CRCDAT	CTRLCRC
		7:0	PDAT	XSIZ	DSTRD	SSTRD	DSA	SSA	EVCTRL	CTRLB
0xE0	CHSTATBC1	31:24								
		23:16								BBTC[16]
		15:8	BBTC[15:8]							
		7:0	BBTC[7:0]							
0xE4	CHSTATCC1	31:24								
		23:16								
		15:8						CBTC[10:8]		
		7:0	CBTC[7:0]							

Channelk Register Summary, k = 0,1,...,11 (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0xE8	CHSTAT1	31:24								
		23:16								
		15:8								
		7:0						DREAD	CELLBUSY	BLKBUSY
0xEC ... 0xEF	Reserved									
0xF0	CHCTRLA2	31:24								RUNSTDBY
		23:16								SWFRC
		15:8								LLEN
		7:0								ENABLE
0xF4	CHCTRLB2	31:24	CRCEN		CASTEN			PATEN	PATLEN	PIGNEN
		23:16	TRIG[7:0]							
		15:8	BYTORD[1:0]		WBOEN				PR[1:0]	
		7:0		RAS[2:0]				WAS[2:0]		
0xF8	CHEVCTRL2	31:24								
		23:16								
		15:8								
		7:0	EVOE	EVSTRIE	EVAUXIE		EVOMODE[1:0]		EVAUXACT[1:0]	
0xFC	CHINTENCLR2	31:24								
		23:16								
		15:8								
		7:0			LL	BH	BC	CC	TA	SD
0x0100	CHINTENSET2	31:24								
		23:16								
		15:8								
		7:0			LL	BH	BC	CC	TA	SD
0x0100	CHINTF2	31:24								
		23:16						RDE	WRE	
		15:8								
		7:0			WRELL	BH	BC	CC	TA	SD
0x0104 ... 0x0107	Reserved									
0x0108	CHSSA2	31:24	SSA[31:24]							
		23:16	SSA[23:16]							
		15:8	SSA[15:8]							
		7:0	SSA[7:0]							
0x010C	CHDSA2	31:24	DSA[31:24]							
		23:16	DSA[23:16]							
		15:8	DSA[15:8]							
		7:0	DSA[7:0]							
0x0110	CHSSTRD2	31:24								
		23:16								
		15:8	SSTRD[15:8]							
		7:0	SSTRD[7:0]							
0x0114	CHDSTRD2	31:24								
		23:16								
		15:8	DSTRD[15:8]							
		7:0	DSTRD[7:0]							
0x0118	CHXSIZ2	31:24	BLKSZ[15:8]							
		23:16	BLKSZ[7:0]							
		15:8							CSZ[9:8]	
		7:0	CSZ[7:0]							
0x011C	CHPDAT2	31:24	PIGN[7:0]							
		23:16								
		15:8	PDAT[15:8]							
		7:0	PDAT[7:0]							

Channelk Register Summary, k = 0,1,...,11 (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0120	CHCTRLCRC2	31:24								
		23:16								
		15:8								
		7:0	CRCRIN	CRCROUT	CRCXOR		CRCAPP	CRCMD[2:0]		
0x0124	CHCRCDAT2	31:24	CRCDAT[31:24]							
		23:16	CRCDAT[23:16]							
		15:8	CRCDAT[15:8]							
		7:0	CRCDAT[7:0]							
0x0128	CHNXT2	31:24	NXT[31:24]							
		23:16	NXT[23:16]							
		15:8	NXT[15:8]							
		7:0	NXT[7:0]							
0x012C	CHLLCFGSTAT2	31:24								
		23:16								
		15:8							CRCDAT	CTRLCRC
		7:0	PDAT	XSIZ	DSTRD	SSTRD	DSA	SSA	EVCTRL	CTRLB
0x0130	CHSTATBC2	31:24								
		23:16								BBTC[16]
		15:8	BBTC[15:8]							
		7:0	BBTC[7:0]							
0x0134	CHSTATCC2	31:24								
		23:16								
		15:8						CBTC[10:8]		
		7:0	CBTC[7:0]							
0x0138	CHSTAT2	31:24								
		23:16								
		15:8								
		7:0						DREAD	CELLBUSY	BLKBUSY
0x013C ... 0x013F	Reserved									
0x0140	CHCTRLA3	31:24								RUNSTDBY
		23:16								SWFRC
		15:8								LLEN
		7:0								ENABLE
0x0144	CHCTRLB3	31:24	CRCEN		CASTEN			PATEN	PATLEN	PIGNEN
		23:16	TRIG[7:0]							
		15:8	BYTORD[1:0]		WBOEN				PR[1:0]	
		7:0		RAS[2:0]				WAS[2:0]		
0x0148	CHEVCTRL3	31:24								
		23:16								
		15:8								
		7:0	EVOE	EVSTRIE	EVAUXIE		EVOMODE[1:0]		EVAUXACT[1:0]	
0x014C	CHINTENCLR3	31:24								
		23:16								
		15:8								
		7:0			LL	BH	BC	CC	TA	SD
0x0150	CHINTENSET3	31:24								
		23:16								
		15:8								
		7:0			LL	BH	BC	CC	TA	SD
0x0150	CHINTF3	31:24								
		23:16						RDE	WRE	
		15:8								
		7:0			WRELL	BH	BC	CC	TA	SD
0x0154 ... 0x0157	Reserved									

Channelk Register Summary, k = 0,1,...,11 (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0158	CHSSA3	31:24	SSA[31:24]							
		23:16	SSA[23:16]							
		15:8	SSA[15:8]							
		7:0	SSA[7:0]							
0x015C	CHDSA3	31:24	DSA[31:24]							
		23:16	DSA[23:16]							
		15:8	DSA[15:8]							
		7:0	DSA[7:0]							
0x0160	CHSSTRD3	31:24								
		23:16								
		15:8	SSTRD[15:8]							
		7:0	SSTRD[7:0]							
0x0164	CHDSTRD3	31:24								
		23:16								
		15:8	DSTRD[15:8]							
		7:0	DSTRD[7:0]							
0x0168	CHXSIZ3	31:24	BLKSZ[15:8]							
		23:16	BLKSZ[7:0]							
		15:8							CSZ[9:8]	
		7:0	CSZ[7:0]							
0x016C	CHPDAT3	31:24	PIGN[7:0]							
		23:16								
		15:8	PDAT[15:8]							
		7:0	PDAT[7:0]							
0x0170	CHCTRLCRC3	31:24								
		23:16								
		15:8								
		7:0	CRCRIN	CRCROUT	CRCXOR		CRCAPP	CRCMD[2:0]		
0x0174	CHCRCDAT3	31:24	CRCDAT[31:24]							
		23:16	CRCDAT[23:16]							
		15:8	CRCDAT[15:8]							
		7:0	CRCDAT[7:0]							
0x0178	CHNXT3	31:24	NXT[31:24]							
		23:16	NXT[23:16]							
		15:8	NXT[15:8]							
		7:0	NXT[7:0]							
0x017C	CHLLCFGSTAT3	31:24								
		23:16								
		15:8							CRCDAT	CTRLCRC
		7:0	PDAT	XSIZ	DSTRD	SSTRD	DSA	SSA	EVCTRL	CTRLB
0x0180	CHSTATBC3	31:24								
		23:16								BBTC[16]
		15:8	BBTC[15:8]							
		7:0	BBTC[7:0]							
0x0184	CHSTATCC3	31:24								
		23:16								
		15:8						CBTC[10:8]		
		7:0	CBTC[7:0]							
0x0188	CHSTAT3	31:24								
		23:16								
		15:8								
		7:0						DREAD	CELLBUSY	BLKBUSY
0x018C ... 0x018F	Reserved									
0x0190	CHCTRLA4	31:24								RUNSTDBY
		23:16								SWFRC
		15:8								LLEN
		7:0								ENABLE

Channelk Register Summary, k = 0,1,...,11 (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0194	CHCTRLB4	31:24	CRCEN		CASTEN			PATEN	PATLEN	PIGNEN
		23:16	TRIG[7:0]							
		15:8	BYTORD[1:0]		WBOEN				PR[1:0]	
		7:0		RAS[2:0]				WAS[2:0]		
0x0198	CHEVCTRL4	31:24								
		23:16								
		15:8								
		7:0	EVOE	EVSTRIE	EVAUXIE		EVOMODE[1:0]		EVAUXACT[1:0]	
0x019C	CHINTENCLR4	31:24								
		23:16								
		15:8								
		7:0			LL	BH	BC	CC	TA	SD
0x01A0	CHINTENSET4	31:24								
		23:16								
		15:8								
		7:0			LL	BH	BC	CC	TA	SD
0x01A0	CHINTF4	31:24								
		23:16						RDE	WRE	
		15:8								
		7:0			WRELL	BH	BC	CC	TA	SD
0x01A4 ... 0x01A7	Reserved									
0x01A8	CHSSA4	31:24	SSA[31:24]							
		23:16	SSA[23:16]							
		15:8	SSA[15:8]							
		7:0	SSA[7:0]							
0x01AC	CHDSA4	31:24	DSA[31:24]							
		23:16	DSA[23:16]							
		15:8	DSA[15:8]							
		7:0	DSA[7:0]							
0x01B0	CHSSTRD4	31:24								
		23:16								
		15:8	SSTRD[15:8]							
		7:0	SSTRD[7:0]							
0x01B4	CHDSTRD4	31:24								
		23:16								
		15:8	DSTRD[15:8]							
		7:0	DSTRD[7:0]							
0x01B8	CHXSIZ4	31:24	BLKSZ[15:8]							
		23:16	BLKSZ[7:0]							
		15:8							CSZ[9:8]	
		7:0	CSZ[7:0]							
0x01BC	CHPDAT4	31:24	PIGN[7:0]							
		23:16								
		15:8	PDAT[15:8]							
		7:0	PDAT[7:0]							
0x01C0	CHCTRLCRC4	31:24								
		23:16								
		15:8								
		7:0	CRCRIN	CRCROUT	CRCXOR		CRCAPP	CRCMD[2:0]		
0x01C4	CHCRCDAT4	31:24	CRCDAT[31:24]							
		23:16	CRCDAT[23:16]							
		15:8	CRCDAT[15:8]							
		7:0	CRCDAT[7:0]							
0x01C8	CHNXT4	31:24	NXT[31:24]							
		23:16	NXT[23:16]							
		15:8	NXT[15:8]							
		7:0	NXT[7:0]							

Channelk Register Summary, k = 0,1,...,11 (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x01CC	CHLLCFGSTAT4	31:24								
		23:16								
		15:8							CRCDAT	CTRLCRC
		7:0	PDAT	XSIZ	DSTRD	SSTRD	DSA	SSA	EVCTRL	CTRLB
0x01D0	CHSTATBC4	31:24								
		23:16								BBTC[16]
		15:8	BBTC[15:8]							
		7:0	BBTC[7:0]							
0x01D4	CHSTATCC4	31:24								
		23:16								
		15:8						CBTC[10:8]		
		7:0	CBTC[7:0]							
0x01D8	CHSTAT4	31:24								
		23:16								
		15:8								
		7:0						DREAD	CELLBUSY	BLKBUSY
0x01DC ... 0x01DF	Reserved									
0x01E0	CHCTRLA5	31:24								RUNSTDBY
		23:16								SWFRC
		15:8								LLEN
		7:0								ENABLE
0x01E4	CHCTRLB5	31:24	CRCEN		CASTEN			PATEN	PATLEN	PIGNEN
		23:16	TRIG[7:0]							
		15:8	BYTORD[1:0]		WBOEN				PR[1:0]	
		7:0		RAS[2:0]				WAS[2:0]		
0x01E8	CHEVCTRL5	31:24								
		23:16								
		15:8								
		7:0	EVOE	EVSTRIE	EVAUXIE		EVOMODE[1:0]		EVAUXACT[1:0]	
0x01EC	CHINTENCLR5	31:24								
		23:16								
		15:8								
		7:0			LL	BH	BC	CC	TA	SD
0x01F0	CHINTENSET5	31:24								
		23:16								
		15:8								
		7:0			LL	BH	BC	CC	TA	SD
0x01F0	CHINTF5	31:24								
		23:16						RDE	WRE	
		15:8								
		7:0			WRELL	BH	BC	CC	TA	SD
0x01F4 ... 0x01F7	Reserved									
0x01F8	CHSSA5	31:24	SSA[31:24]							
		23:16	SSA[23:16]							
		15:8	SSA[15:8]							
		7:0	SSA[7:0]							
0x01FC	CHDSA5	31:24	DSA[31:24]							
		23:16	DSA[23:16]							
		15:8	DSA[15:8]							
		7:0	DSA[7:0]							
0x0200	CHSSTRD5	31:24								
		23:16								
		15:8	SSTRD[15:8]							
		7:0	SSTRD[7:0]							

Channelk Register Summary, k = 0,1,...,11 (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0204	CHDSTRD5	31:24								
		23:16								
		15:8	DSTRD[15:8]							
		7:0	DSTRD[7:0]							
0x0208	CHXSIZ5	31:24	BLKSZ[15:8]							
		23:16	BLKSZ[7:0]							
		15:8							CSZ[9:8]	
		7:0	CSZ[7:0]							
0x020C	CHPDAT5	31:24	PIGN[7:0]							
		23:16								
		15:8	PDAT[15:8]							
		7:0	PDAT[7:0]							
0x0210	CHCTRLCRC5	31:24								
		23:16								
		15:8								
		7:0	CRCRIN	CRCROUT	CRCXOR		CRCAPP	CRCMD[2:0]		
0x0214	CHCRCDAT5	31:24	CRCDAT[31:24]							
		23:16	CRCDAT[23:16]							
		15:8	CRCDAT[15:8]							
		7:0	CRCDAT[7:0]							
0x0218	CHNXT5	31:24	NXT[31:24]							
		23:16	NXT[23:16]							
		15:8	NXT[15:8]							
		7:0	NXT[7:0]							
0x021C	CHLLCFGSTAT5	31:24								
		23:16								
		15:8							CRCDAT	CTRLCRC
		7:0	PDAT	XSIZ	DSTRD	SSTRD	DSA	SSA	EVCTRL	CTRLB
0x0220	CHSTATBC5	31:24								
		23:16								BBTC[16]
		15:8	BBTC[15:8]							
		7:0	BBTC[7:0]							
0x0224	CHSTATCC5	31:24								
		23:16								
		15:8						CBTC[10:8]		
		7:0	CBTC[7:0]							
0x0228	CHSTAT5	31:24								
		23:16								
		15:8								
		7:0						DREAD	CELLBUSY	BLKBUSY
0x022C ... 0x022F	Reserved									
0x0230	CHCTRLA6	31:24								RUNSTDBY
		23:16								SWFRC
		15:8								LLEN
		7:0								ENABLE
0x0234	CHCTRLB6	31:24	CRCEN		CASTEN			PATEN	PATLEN	PIGNEN
		23:16	TRIG[7:0]							
		15:8	BYTORD[1:0]		WBOEN				PR[1:0]	
		7:0		RAS[2:0]				WAS[2:0]		
0x0238	CHEVCTRL6	31:24								
		23:16								
		15:8								
		7:0	EVOE	EVSTRIE	EVAUXIE		EVOMODE[1:0]		EVAUXACT[1:0]	
0x023C	CHINTENCLR6	31:24								
		23:16								
		15:8								
		7:0			LL	BH	BC	CC	TA	SD

Channelk Register Summary, k = 0,1,...,11 (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0240	CHINTENSET6	31:24								
		23:16								
		15:8								
		7:0			LL	BH	BC	CC	TA	SD
0x0240	CHINTF6	31:24								
		23:16						RDE	WRE	
		15:8								
		7:0			WRELL	BH	BC	CC	TA	SD
0x0244 ... 0x0247	Reserved									
0x0248	CHSSA6	31:24	SSA[31:24]							
		23:16	SSA[23:16]							
		15:8	SSA[15:8]							
		7:0	SSA[7:0]							
0x024C	CHDSA6	31:24	DSA[31:24]							
		23:16	DSA[23:16]							
		15:8	DSA[15:8]							
		7:0	DSA[7:0]							
0x0250	CHSSTRD6	31:24								
		23:16								
		15:8	SSTRD[15:8]							
		7:0	SSTRD[7:0]							
0x0254	CHDSTRD6	31:24								
		23:16								
		15:8	DSTRD[15:8]							
		7:0	DSTRD[7:0]							
0x0258	CHXSIZ6	31:24	BLKSZ[15:8]							
		23:16	BLKSZ[7:0]							
		15:8							CSZ[9:8]	
		7:0	CSZ[7:0]							
0x025C	CHPDAT6	31:24	PIGN[7:0]							
		23:16								
		15:8	PDAT[15:8]							
		7:0	PDAT[7:0]							
0x0260	CHCTRLCRC6	31:24								
		23:16								
		15:8								
		7:0	CRCRIN	CRCROUT	CRCXOR		CRCAPP	CRCMD[2:0]		
0x0264	CHCRCDAT6	31:24	CRCDAT[31:24]							
		23:16	CRCDAT[23:16]							
		15:8	CRCDAT[15:8]							
		7:0	CRCDAT[7:0]							
0x0268	CHNXT6	31:24	NXT[31:24]							
		23:16	NXT[23:16]							
		15:8	NXT[15:8]							
		7:0	NXT[7:0]							
0x026C	CHLLCFGSTAT6	31:24								
		23:16								
		15:8							CRCDAT	CTRLCRC
		7:0	PDAT	XSIZ	DSTRD	SSTRD	DSA	SSA	EVCTRL	CTRLB
0x0270	CHSTATBC6	31:24								
		23:16								BBTC[16]
		15:8	BBTC[15:8]							
		7:0	BBTC[7:0]							
0x0274	CHSTATCC6	31:24								
		23:16								
		15:8						CBTC[10:8]		
		7:0	CBTC[7:0]							

Channelk Register Summary, k = 0,1,...,11 (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0278	CHSTAT6	31:24								
		23:16								
		15:8								
		7:0						DREAD	CELLBUSY	BLKBUSY
0x027C ... 0x027F	Reserved									
0x0280	CHCTRLA7	31:24								RUNSTDBY
		23:16								SWFRC
		15:8								LLEN
		7:0								ENABLE
0x0284	CHCTRLB7	31:24	CRCEN		CASTEN			PATEN	PATLEN	PIGNEN
		23:16	TRIG[7:0]							
		15:8	BYTORD[1:0]		WBOEN				PR[1:0]	
		7:0		RAS[2:0]				WAS[2:0]		
0x0288	CHEVCTRL7	31:24								
		23:16								
		15:8								
		7:0	EVOE	EVSTRIE	EVAUXIE		EVOMODE[1:0]		EVAUXACT[1:0]	
0x028C	CHINTENCLR7	31:24								
		23:16								
		15:8								
		7:0			LL	BH	BC	CC	TA	SD
0x0290	CHINTENSET7	31:24								
		23:16								
		15:8								
		7:0			LL	BH	BC	CC	TA	SD
0x0290	CHINTF7	31:24								
		23:16						RDE	WRE	
		15:8								
		7:0			WRELL	BH	BC	CC	TA	SD
0x0294 ... 0x0297	Reserved									
0x0298	CHSSA7	31:24	SSA[31:24]							
		23:16	SSA[23:16]							
		15:8	SSA[15:8]							
		7:0	SSA[7:0]							
0x029C	CHDSA7	31:24	DSA[31:24]							
		23:16	DSA[23:16]							
		15:8	DSA[15:8]							
		7:0	DSA[7:0]							
0x02A0	CHSSTRD7	31:24								
		23:16								
		15:8	SSTRD[15:8]							
		7:0	SSTRD[7:0]							
0x02A4	CHDSTRD7	31:24								
		23:16								
		15:8	DSTRD[15:8]							
		7:0	DSTRD[7:0]							
0x02A8	CHXSIZ7	31:24	BLKSZ[15:8]							
		23:16	BLKSZ[7:0]							
		15:8							CSZ[9:8]	
		7:0	CSZ[7:0]							
0x02AC	CHPDAT7	31:24	PIGN[7:0]							
		23:16								
		15:8	PDAT[15:8]							
		7:0	PDAT[7:0]							

Channelk Register Summary, k = 0,1,...,11 (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x02B0	CHCTRLCRC7	31:24								
		23:16								
		15:8								
		7:0	CRCRIN	CRCROUT	CRCXOR		CRCAPP	CRCMD[2:0]		
0x02B4	CHCRCDAT7	31:24	CRCDAT[31:24]							
		23:16	CRCDAT[23:16]							
		15:8	CRCDAT[15:8]							
		7:0	CRCDAT[7:0]							
0x02B8	CHNXT7	31:24	NXT[31:24]							
		23:16	NXT[23:16]							
		15:8	NXT[15:8]							
		7:0	NXT[7:0]							
0x02BC	CHLLCFGSTAT7	31:24								
		23:16								
		15:8							CRCDAT	CTRLCRC
		7:0	PDAT	XSIZ	DSTRD	SSTRD	DSA	SSA	EVCTRL	CTRLB
0x02C0	CHSTATBC7	31:24								
		23:16								BBTC[16]
		15:8	BBTC[15:8]							
		7:0	BBTC[7:0]							
0x02C4	CHSTATCC7	31:24								
		23:16								
		15:8						CBTC[10:8]		
		7:0	CBTC[7:0]							
0x02C8	CHSTAT7	31:24								
		23:16								
		15:8								
		7:0						DREAD	CELLBUSY	BLKBUSY
0x02CC ... 0x02CF	Reserved									
0x02D0	CHCTRLA8	31:24								RUNSTDBY
		23:16								SWFRC
		15:8								LLEN
		7:0								ENABLE
0x02D4	CHCTRLB8	31:24	CRCEN		CASTEN			PATEN	PATLEN	PIGNEN
		23:16	TRIG[7:0]							
		15:8	BYTORD[1:0]		WBOEN				PR[1:0]	
		7:0		RAS[2:0]				WAS[2:0]		
0x02D8	CHEVCTRL8	31:24								
		23:16								
		15:8								
		7:0	EVOE	EVSTRIE	EVAUXIE		EVOMODE[1:0]		EVAUXACT[1:0]	
0x02DC	CHINTENCLR8	31:24								
		23:16								
		15:8								
		7:0			LL	BH	BC	CC	TA	SD
0x02E0	CHINTENSET8	31:24								
		23:16								
		15:8								
		7:0			LL	BH	BC	CC	TA	SD
0x02E0	CHINTF8	31:24								
		23:16						RDE	WRE	
		15:8								
		7:0			WRELL	BH	BC	CC	TA	SD
0x02E4 ... 0x02E7	Reserved									

Channelk Register Summary, k = 0,1,...,11 (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x02E8	CHSSA8	31:24	SSA[31:24]							
		23:16	SSA[23:16]							
		15:8	SSA[15:8]							
		7:0	SSA[7:0]							
0x02EC	CHDSA8	31:24	DSA[31:24]							
		23:16	DSA[23:16]							
		15:8	DSA[15:8]							
		7:0	DSA[7:0]							
0x02F0	CHSSTRD8	31:24								
		23:16								
		15:8	SSTRD[15:8]							
		7:0	SSTRD[7:0]							
0x02F4	CHDSTRD8	31:24								
		23:16								
		15:8	DSTRD[15:8]							
		7:0	DSTRD[7:0]							
0x02F8	CHXSIZ8	31:24	BLKSZ[15:8]							
		23:16	BLKSZ[7:0]							
		15:8							CSZ[9:8]	
		7:0	CSZ[7:0]							
0x02FC	CHPDAT8	31:24	PIGN[7:0]							
		23:16								
		15:8	PDAT[15:8]							
		7:0	PDAT[7:0]							
0x0300	CHCTRLCRC8	31:24								
		23:16								
		15:8								
		7:0	CRCRIN	CRCROUT	CRCXOR		CRCAPP	CRCMD[2:0]		
0x0304	CHCRCDAT8	31:24	CRCDAT[31:24]							
		23:16	CRCDAT[23:16]							
		15:8	CRCDAT[15:8]							
		7:0	CRCDAT[7:0]							
0x0308	CHNXT8	31:24	NXT[31:24]							
		23:16	NXT[23:16]							
		15:8	NXT[15:8]							
		7:0	NXT[7:0]							
0x030C	CHLLCFGSTAT8	31:24								
		23:16								
		15:8							CRCDAT	CTRLCRC
		7:0	PDAT	XSIZ	DSTRD	SSTRD	DSA	SSA	EVCTRL	CTRLB
0x0310	CHSTATBC8	31:24								
		23:16								BBTC[16]
		15:8	BBTC[15:8]							
		7:0	BBTC[7:0]							
0x0314	CHSTATCC8	31:24								
		23:16								
		15:8						CBTC[10:8]		
		7:0	CBTC[7:0]							
0x0318	CHSTAT8	31:24								
		23:16								
		15:8								
		7:0						DREAD	CELLBUSY	BLKBUSY
0x031C ... 0x031F	Reserved									
0x0320	CHCTRLA9	31:24								RUNSTDBY
		23:16								SWFRC
		15:8								LLEN
		7:0								ENABLE

Channelk Register Summary, k = 0,1,...,11 (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0324	CHCTRLB9	31:24	CRCEN		CASTEN			PATEN	PATLEN	PIGNEN
		23:16	TRIG[7:0]							
		15:8	BYTORD[1:0]		WBOEN				PR[1:0]	
		7:0		RAS[2:0]				WAS[2:0]		
0x0328	CHEVCTRL9	31:24								
		23:16								
		15:8								
		7:0	EVOE	EVSTRIE	EVAUXIE		EVOMODE[1:0]		EVAUXACT[1:0]	
0x032C	CHINTENCLR9	31:24								
		23:16								
		15:8								
		7:0			LL	BH	BC	CC	TA	SD
0x0330	CHINTENSET9	31:24								
		23:16								
		15:8								
		7:0			LL	BH	BC	CC	TA	SD
0x0330	CHINTF9	31:24								
		23:16						RDE	WRE	
		15:8								
		7:0			WRELL	BH	BC	CC	TA	SD
0x0334 ... 0x0337	Reserved									
0x0338	CHSSA9	31:24	SSA[31:24]							
		23:16	SSA[23:16]							
		15:8	SSA[15:8]							
		7:0	SSA[7:0]							
0x033C	CHDSA9	31:24	DSA[31:24]							
		23:16	DSA[23:16]							
		15:8	DSA[15:8]							
		7:0	DSA[7:0]							
0x0340	CHSSTRD9	31:24								
		23:16								
		15:8	SSTRD[15:8]							
		7:0	SSTRD[7:0]							
0x0344	CHDSTRD9	31:24								
		23:16								
		15:8	DSTRD[15:8]							
		7:0	DSTRD[7:0]							
0x0348	CHXSIZ9	31:24	BLKSZ[15:8]							
		23:16	BLKSZ[7:0]							
		15:8							CSZ[9:8]	
		7:0	CSZ[7:0]							
0x034C	CHPDAT9	31:24	PIGN[7:0]							
		23:16								
		15:8	PDAT[15:8]							
		7:0	PDAT[7:0]							
0x0350	CHCTRLCRC9	31:24								
		23:16								
		15:8								
		7:0	CRCRIN	CRCROUT	CRCXOR		CRCAPP	CRCMD[2:0]		
0x0354	CHCRCDAT9	31:24	CRCDAT[31:24]							
		23:16	CRCDAT[23:16]							
		15:8	CRCDAT[15:8]							
		7:0	CRCDAT[7:0]							
0x0358	CHNXT9	31:24	NXT[31:24]							
		23:16	NXT[23:16]							
		15:8	NXT[15:8]							
		7:0	NXT[7:0]							

Channelk Register Summary, k = 0,1,...,11 (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x035C	CHLLCFGSTAT9	31:24								
		23:16								
		15:8							CRCDAT	CTRLCRC
		7:0	PDAT	XSIZ	DSTRD	SSTRD	DSA	SSA	EVCTRL	CTRLB
0x0360	CHSTATBC9	31:24								
		23:16								BBTC[16]
		15:8	BBTC[15:8]							
		7:0	BBTC[7:0]							
0x0364	CHSTATCC9	31:24								
		23:16								
		15:8						CBTC[10:8]		
		7:0	CBTC[7:0]							
0x0368	CHSTAT9	31:24								
		23:16								
		15:8								
		7:0						DREAD	CELLBUSY	BLKBUSY
0x036C ... 0x036F	Reserved									
0x0370	CHCTRLA10	31:24								RUNSTDBY
		23:16								SWFRC
		15:8								LLEN
		7:0								ENABLE
0x0374	CHCTRLB10	31:24	CRCEN		CASTEN			PATEN	PATLEN	PIGNEN
		23:16	TRIG[7:0]							
		15:8	BYTORD[1:0]		WBOEN				PR[1:0]	
		7:0		RAS[2:0]				WAS[2:0]		
0x0378	CHEVCTRL10	31:24								
		23:16								
		15:8								
		7:0	EVOE	EVSTRIE	EVAUXIE		EVOMODE[1:0]		EVAUXACT[1:0]	
0x037C	CHINTENCLR10	31:24								
		23:16								
		15:8								
		7:0			LL	BH	BC	CC	TA	SD
0x0380	CHINTENSET10	31:24								
		23:16								
		15:8								
		7:0			LL	BH	BC	CC	TA	SD
0x0380	CHINTF10	31:24								
		23:16						RDE	WRE	
		15:8								
		7:0			WRELL	BH	BC	CC	TA	SD
0x0384 ... 0x0387	Reserved									
0x0388	CHSSA10	31:24	SSA[31:24]							
		23:16	SSA[23:16]							
		15:8	SSA[15:8]							
		7:0	SSA[7:0]							
0x038C	CHDSA10	31:24	DSA[31:24]							
		23:16	DSA[23:16]							
		15:8	DSA[15:8]							
		7:0	DSA[7:0]							
0x0390	CHSSTRD10	31:24								
		23:16								
		15:8	SSTRD[15:8]							
		7:0	SSTRD[7:0]							

Channelk Register Summary, k = 0,1,...,11 (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0394	CHDSTRD10	31:24								
		23:16								
		15:8	DSTRD[15:8]							
		7:0	DSTRD[7:0]							
0x0398	CHXSIZ10	31:24	BLKSZ[15:8]							
		23:16	BLKSZ[7:0]							
		15:8							CSZ[9:8]	
		7:0	CSZ[7:0]							
0x039C	CHPDAT10	31:24	PIGN[7:0]							
		23:16								
		15:8	PDAT[15:8]							
		7:0	PDAT[7:0]							
0x03A0	CHCTRLCRC10	31:24								
		23:16								
		15:8								
		7:0	CRCRIN	CRCROUT	CRCXOR		CRCAPP	CRCMD[2:0]		
0x03A4	CHCRCDAT10	31:24	CRCDAT[31:24]							
		23:16	CRCDAT[23:16]							
		15:8	CRCDAT[15:8]							
		7:0	CRCDAT[7:0]							
0x03A8	CHNXT10	31:24	NXT[31:24]							
		23:16	NXT[23:16]							
		15:8	NXT[15:8]							
		7:0	NXT[7:0]							
0x03AC	CHLLCFGSTAT10	31:24								
		23:16								
		15:8							CRCDAT	CTRLCRC
		7:0	PDAT	XSIZ	DSTRD	SSTRD	DSA	SSA	EVCTRL	CTRLB
0x03B0	CHSTATBC10	31:24								
		23:16								BBTC[16]
		15:8	BBTC[15:8]							
		7:0	BBTC[7:0]							
0x03B4	CHSTATCC10	31:24								
		23:16								
		15:8						CBTC[10:8]		
		7:0	CBTC[7:0]							
0x03B8	CHSTAT10	31:24								
		23:16								
		15:8								
		7:0						DREAD	CELLBUSY	BLKBUSY
0x03BC ... 0x03BF	Reserved									
0x03C0	CHCTRLA11	31:24								RUNSTDBY
		23:16								SWFRC
		15:8								LLEN
		7:0								ENABLE
0x03C4	CHCTRLB11	31:24	CRCEN		CASTEN			PATEN	PATLEN	PIGNEN
		23:16	TRIG[7:0]							
		15:8	BYTORD[1:0]		WBOEN				PR[1:0]	
		7:0		RAS[2:0]				WAS[2:0]		
0x03C8	CHEVCTRL11	31:24								
		23:16								
		15:8								
		7:0	EVOE	EVSTRIE	EVAUXIE		EVOMODE[1:0]		EVAUXACT[1:0]	
0x03CC	CHINTENCLR11	31:24								
		23:16								
		15:8								
		7:0			LL	BH	BC	CC	TA	SD

Channelk Register Summary, k = 0,1,...,11 (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x03D0	CHINTENSET11	31:24								
		23:16								
		15:8								
		7:0			LL	BH	BC	CC	TA	SD
0x03D0	CHINTF11	31:24								
		23:16						RDE	WRE	
		15:8								
		7:0			WRELL	BH	BC	CC	TA	SD
0x03D4 ... 0x03D7	Reserved									
0x03D8	CHSSA11	31:24	SSA[31:24]							
		23:16	SSA[23:16]							
		15:8	SSA[15:8]							
		7:0	SSA[7:0]							
0x03DC	CHDSA11	31:24	DSA[31:24]							
		23:16	DSA[23:16]							
		15:8	DSA[15:8]							
		7:0	DSA[7:0]							
0x03E0	CHSSTRD11	31:24								
		23:16								
		15:8	SSTRD[15:8]							
		7:0	SSTRD[7:0]							
0x03E4	CHDSTRD11	31:24								
		23:16								
		15:8	DSTRD[15:8]							
		7:0	DSTRD[7:0]							
0x03E8	CHXSIZ11	31:24	BLKSZ[15:8]							
		23:16	BLKSZ[7:0]							
		15:8							CSZ[9:8]	
		7:0	CSZ[7:0]							
0x03EC	CHPDAT11	31:24	PIGN[7:0]							
		23:16								
		15:8	PDAT[15:8]							
		7:0	PDAT[7:0]							
0x03F0	CHCTRLCRC11	31:24								
		23:16								
		15:8								
		7:0	CRCRIN	CRCROUT	CRCXOR		CRCAPP	CRCMD[2:0]		
0x03F4	CHCRCDAT11	31:24	CRCDAT[31:24]							
		23:16	CRCDAT[23:16]							
		15:8	CRCDAT[15:8]							
		7:0	CRCDAT[7:0]							
0x03F8	CHNXT11	31:24	NXT[31:24]							
		23:16	NXT[23:16]							
		15:8	NXT[15:8]							
		7:0	NXT[7:0]							
0x03FC	CHLLCFGSTAT11	31:24								
		23:16								
		15:8							CRCDAT	CTRLCRC
		7:0	PDAT	XSIZ	DSTRD	SSTRD	DSA	SSA	EVCTRL	CTRLB
0x0400	CHSTATBC11	31:24								
		23:16								BBTC[16]
		15:8	BBTC[15:8]							
		7:0	BBTC[7:0]							
0x0404	CHSTATCC11	31:24								
		23:16								
		15:8						CBTC[10:8]		
		7:0	CBTC[7:0]							

Channelk Register Summary, k = 0,1,...,11 (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0408	CHSTAT11	31:24								
		23:16								
		15:8								
		7:0						DREAD	CELLBUSY	BLKBUSY

26.10.1. Channel Control A Register

Name: CHCTRLAk
Offset: 0x50 + k*0x50 [k=0..11]
Reset: 0x00000000
Property: PAC Write-Protection

Table 26-15. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
								RUNSTDBY
Access								R/W
Reset								0

Bit	23	22	21	20	19	18	17	16
								SWFRC
Access								W
Reset								0

Bit	15	14	13	12	11	10	9	8
								LLEN
Access								R/W/HC
Reset								0

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

Bit 24 – RUNSTDBY Run in Standby

This bit is used to keep the DMA channel running in standby mode:

Value	Description
0	The channel is halted in standby.
1	The channel continues to run in standby. Continue module operation in idle/sleep mode.

Bit 16 – SWFRC Software Force Trigger

Write to 1 to issue a start trigger to the channel. Reading this bit always returns 0.

Bit 8 – LLEN Linked List Enable

Value	Description
0	Next descriptor loading is disabled for the channel.
1	DMA will load the next descriptor at address location to by CHNXTk.NXT[31:0] on completion of the current block transfer or if the channel is idle (i.e. CHCTRLAk.ENABLE=0 and CHSTATk.BLK-BUSY=0). If CHNXTk.NXT[31:0] = 0xFFFF_FFFF (NULL) the DMA will set the CHINTFk.LL status bit and clear LLEN. No further action takes place.

Bit 0 – ENABLE Channel Enable

Writing a 1 to ENABLE enables a block transfer. Upon completion or abort of the block transfer the DMA clears ENABLE.

Value	Description
0	Disable channel block transfers or suspend block transfer if CHSTATk.BLKBUSY=1.
1	Block transfer enabled. The DMA will initiate a block transfer on the start trigger selected by CHCTRLBk.TRIG or a software trigger, CHCTRLAk.SWFRC=1.

26.10.2. Channel Control B Register

Name: CHCTRLBk
Offset: 0x54 + k*0x50 [k=0..11]
Reset: 0x00000000
Property: PAC Write-Protection, CHCTRLAk.ENABLE =1 write protect

Note: An attempt to enter a priority higher than the maximum value implemented will write that maximum value instead.

Table 26-16. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	CRCEN		CASTEN			PATEN	PATLEN	PIGNEN
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
	TRIG[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
	BYTORD[1:0]		WBOEN				PR[1:0]	
Access	R/W	R/W	R/W				R/W/HS/HC	R/W/HS/HC
Reset	0	0	0				0	0

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

Bit 31 – CRCEN CRC Enable Bit

Value	Description
0	CRC module is disabled for this channel and transfers bypass the CRC module
1	CRC module is enabled for this channel and transfers are routed through the CRC module

Bit 29 – CASTEN Cell Auto Start Enable of Ensuing Transfers for this channel.

Value	Description
0	The start of each cell transfer in a block transfer will be delayed until requested by a start trigger.
1	Transfer a complete block of data on a single start trigger event. The first cell transfer will be delayed until requested by an start trigger. Once started it will continue processing additional cells continuously without any additional starting event until a block is finished.

Bit 26 – PATEN Channel Pattern Match Abort Enable

Value	Description
0	Pattern match is disabled
1	Abort transfer and clear CHCTRLA*k*.ENABLE on pattern match

Bit 25 – PATLEN Pattern Match Length

Value	Description
0	1 byte length
1	2 byte length

Bit 24 – PIGNEN Enable Pattern Ignore Byte

Value	Description
0	Disable this feature.
1	Treat any byte that matches PIGN bits as a don't care when pattern matching is enabled.

Bits 23:16 – TRIG[7:0] Trigger that can Start a Channel Transfer

If programed to a value greater than the maximum trigger index listed in Table 25-3, all external triggers are disabled. Only software triggering is available.
`k`= Peripheral request connected, where `k` is a value from 2 to 90

Value	Description
0	External events disabled, only software trigger can start a transfer.
1	Event system trigger (evsys_dma_chstrt_evt[*k*]) if*k*< 24 otherwise reserved

Bits 15:14 – BYTORD[1:0] Byte Order

Byte swapping takes place prior to sending data into the destination.

Value	Description
11	Bytes Swapped as: 3→2 / 2→3 / 1→0 / 0→1
10	Bytes Swapped as: 3→1 / 2→0 / 1→3 / 0→2
01	Bytes Swapped as: 3→0 / 2→1 / 1→2 / 0→3
00	Unchanged

Bit 13 – WBOEN Write Byte Order Enable

Value	Description
0	Write source data unchanged.
1	Write out data according to BYTORD[1:0].

Bits 9:8 – PR[1:0] Channel Priority Level

Sets the priority level of the channel. Reading back this value returns the current priority level of the channel. If EVAUXACT=1, hardware may adjust the priority level.

Value	Description
11	Channel has priority 4 (highest)
10	Channel has priority 3
01	Channel has priority 2
00	Channel has priority 1

Bits 6:4 – RAS[2:0] Channel Read Address Sequence

Sets the read address and transfer size.

Value	Description
111	Reserved
110	Reserved
101	Fixed address word (32-bit) burst transfer
100	Fixed address of halfword (16-bit) operand (single half-word aligned address)
011	Fixed byte address (single byte address with enable based upon 2 LSBs)
010	Auto increment address and transfer size.
001	Incrementing address +2 with transfers of halfword operands
000	Incrementing address +1 with transfers of byte operands

Bits 2:0 – WAS[2:0] Channel Write Address Sequence

Value	Description
111	Reserved
110	Reserved
101	Fixed address word (32-bit) burst transfer
100	Fixed address of halfword (16-bit) operand (single half-word aligned address)
011	Fixed byte address (single byte address with enable based upon 2 LSBs)
010	Auto increment address and transfer size.
001	Incrementing address +2 with transfers of halfword operands
000	Incrementing address +1 with transfers of byte operands

26.10.3. Channel Event Control Register

Name: CHEVCTRLk
Offset: 0x58 + k*0x50 [k=0..11]
Reset: 0x00000000
Property: PAC Write-Protection, CHCTRLAk.ENABLE =1 write protect

Table 26-17. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	EVOE	EVSTRIE	EVAUXIE		EVOMODE[1:0]		EVAUXACT[1:0]	
Access	R/W	R/W	R/W		R/W	R/W	R/W	R/W
Reset	0	0	0		0	0	0	0

Bit 7 – EVOE Channel Event Output Enable

This bit indicates if the Channel event generation is enabled. The event will be generated for every condition defined in the Channel Event Output Mode bits (CHEVCTRL*k*.EVOMODE).

Value	Description
0	Channel event generation is enabled
1	Channel event generation is enabled

Bit 6 – EVSTRIE Channel Start Event Input Enable

Value	Description
0	Channel event start action is disabled.
1	Channel event start action is enabled. See CHCTRLBk.TRIG for details on configuring the channel to use start event as a trigger.

Bit 5 – EVAUXIE Channel Auxiliary Event Enable

Value	Description
0	Channel event auxiliary action is disabled
1	Channel event auxiliary action is enabled

Bits 3:2 – EVOMODE[1:0] Channel Event Output Mode

This field sets when a output trigger event occurs. Output is always active high.

Value	Description
11	Ongoing trigger action from the start event trigger to the completion of a cell transfer reads
10	Ongoing trigger action from the start event trigger to the completion of a cell transfer writes
01	Generate 1 clock cycle strobe at the end of a cell transfer
00	Generate 1 clock cycle strobe at the end of a block transfer

Bits 1:0 – EVAUXACT[1:0] Channel Auxiliary Event Input Action

Value	Description
11	Abort linked list operation and block transfer on the rising edge of the input event
10	Conditional Trigger, active high level event
01	Increment channel priority on the rising edge of the input event
00	Abort block transfer on the rising edge of the input event

26.10.4. Channel Interrupt Enable Clear Register

Name: CHINTENCLRk
Offset: 0x5C + k*0x50 [k=0..11]
Reset: 0x00000000
Property: -

Table 26-18. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			LL	BH	BC	CC	TA	SD
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit 5 – LL Clear Linked List Done Interrupt Enable
Write a 1 to this bit to clear the interrupt enable.

Value	Description
0	No interrupt is enabled
1	Linked-List competed, is set after the block transfer completes and a NULL pointer is encountered

Bit 4 – BH Clear Block Transfer Half Complete Interrupt Enable
Write a 1 to this bit to clear the interrupt enable.

Value	Description
0	No interrupt is enabled
1	Half of the block transfer has completed.

Bit 3 – BC Clear Block Transfer Complete Interrupt Enable
Write a 1 to this bit to clear the interrupt enable.

Value	Description
0	No interrupt is enabled
1	A block transfer has been completed.

Bit 2 – CC Clear Cell Transfer Complete Interrupt Enable
Write a 1 to this bit to clear the interrupt enable.

Value	Description
0	No interrupt is enabled
1	A cell transfer has been completed (CSZ bytes has been transferred).

Bit 1 – TA Clear Transfer Abort Interrupt Enable
Write a 1 to this bit to clear the interrupt enable.

Value	Description
0	No interrupt is enabled
1	An abort trigger event has been detected and the DMA transfer has been aborted. The DMA will also clear CHCTRLA*k*.ENABLE on a TA event.

Bit 0 – SD Clear Start Detected Interrupt Enable
Write a 1 to this bit to clear the interrupt enable.

Value	Description
0	No interrupt is enabled
1	A start trigger event has been detected and the block transfer has started

26.10.5. Channel Interrupt Enable Clear Register

Name: CHINTENSETk
Offset: 0x60 + k*0x50 [k=0..11]
Reset: 0x00000000
Property: -

Table 26-19. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			LL	BH	BC	CC	TA	SD
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit 5 – LL Set Linked List Done Interrupt Enable
Write a 1 to this bit to set the interrupt enable.

Value	Description
0	No interrupt is enabled
1	Linked-List competed, NULL pointer encountered in CHNXT*k*.NXT when attempting to load the next descriptor.

Bit 4 – BH Set Block Transfer Half Complete Interrupt Enable
Write a 1 to this bit to set the interrupt enable.

Value	Description
0	No interrupt is enabled
1	Half of the block transfer has completed.

Bit 3 – BC Set Block Transfer Complete Interrupt Enable
Write a 1 to this bit to set the interrupt enable.

Value	Description
0	No interrupt is enabled
1	A block transfer has been completed.

Bit 2 – CC Set Cell Transfer Complete Interrupt Enable
Write a 1 to this bit to set the interrupt enable.

Value	Description
0	No interrupt is enabled
1	A cell transfer has been completed (CSZ bytes has been transferred).

Bit 1 – TA Set Transfer Abort Interrupt Enable
Write a 1 to this bit to set the interrupt enable.

Value	Description
0	No interrupt is enabled
1	An abort trigger event has been detected and the DMA transfer has been aborted. The DMA will also clear CHCTRLA*k*.ENABLE on a TA event.

Bit 0 – SD Set Start Detected Interrupt Enable
Write a 1 to this bit to set the interrupt enable.

Value	Description
0	No interrupt is enabled
1	A start trigger event has been detected and the block transfer has started

26.10.6. Channel Interrupt Flag Register

Name: CHINTFk
Offset: 0x60 + k*0x50 [k=0..11]
Reset: 0x00000000
Property: -

Note: Interrupt flags must be cleared and then read back to confirm the clear before exiting the ISR to avoid double interrupts.

Reset: This channel resets on a channel reset.

Table 26-20. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
						RDE	WRE	
Access						R/W	R/W	
Reset						0	0	

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			WRELL	BH	BC	CC	TA	SD
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit 18 – RDE Read Error Flag
Write a 1 to this bit to clear the interrupt flag.

Value	Description
0	No read bus errors detected
1	The last read request returned a bus error. The DMA will also clear CHCTRLA*k*.ENABLE and LLEN on a read error.

Bit 17 – WRE Write Error Flag
Write a 1 to this bit to clear the interrupt flag.

Value	Description
0	No read bus errors detected
1	The last write request returned a bus error. The DMA will also clear CHCTRLA*k*.ENABLE and LLEN on a write error.

Bit 5 – WRELL Linked List Done Interrupt Flag
Write a 1 to this bit to clear the interrupt flag.

Value	Description
0	No interrupt is enabled
1	Linked-List competed, NULL pointer encountered in CHNXT*k*.NXT when attempting to load the next descriptor and all data from the current block transfer has completed.

Bit 4 – BH Block Transfer Half Complete Interrupt Flag
Write a 1 to this bit to clear the interrupt flag.

Value	Description
0	No interrupt is enabled
1	Half of the block transfer has completed.

Bit 3 – BC Block Transfer Complete Interrupt Flag
Write a 1 to this bit to clear the interrupt flag.

Value	Description
0	No interrupt is enabled
1	A block transfer has been completed.

Bit 2 – CC Cell Transfer Complete Interrupt Flag
Write a 1 to this bit to clear the interrupt flag.

Value	Description
0	No interrupt is enabled
1	A cell transfer has been completed (CSZ bytes has been transferred).

Bit 1 – TA Transfer Abort Interrupt Flag
Write a 1 to this bit to clear the interrupt flag.

Value	Description
0	No interrupt is enabled
1	An abort trigger event has been detected and the DMA transfer has been aborted. The DMA will also clear CHCTRLA*k*.ENABLE on a TA event.

Bit 0 – SD Start Detected Interrupt Flag
Write a 1 to this bit to set the interrupt enable.

Value	Description
0	No interrupt is enabled
1	A start trigger event has been detected and the block transfer has started

26.10.7. Channel Source Start Address Register

Name: CHSSAk
Offset: 0x68 + k*0x50 [k=0..11]
Reset: 0x00000000
Property: PAC Write-Protection

Table 26-21. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	SSA[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
	SSA[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
	SSA[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
	SSA[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 – SSA[31:0] Channel Source Start Address

This address should be a physical byte address. (The value should never lie outside the implemented memory).
If CHCTRLB*k*.RAS[2:0] = 001 or 100, the address must be halfword aligned where SSA[0] = 0.
If CHCTRLB*k*.RAS[2:0] = 101, the address must be word aligned where SSA[1:0] = 00.
If CHCTRLB*k*.RAS[2:0] = 010 and only word transfers are desired, the address must be word aligned where SSA[1:0] = 00.

26.10.8. Channel Destination Start Address Register

Name: CHDSAk
Offset: 0x6C + k*0x50 [k=0..11]
Reset: 0x00000000
Property: PAC Write-Protection

Offset is k=0..DMA_CH_N-1)

Table 26-22. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	DSA[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
	DSA[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
	DSA[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
	DSA[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 – DSA[31:0] Channel Destination Start Address
This address should be a physical byte address. (The value should never lie outside the implemented memory).
If CHCTRLB*k*.WAS[2:0] = 001 or 100, the address must be halfword aligned where DSA[0] = 0.
If CHCTRLB*k*.WAS[2:0] = 101, the address must be word aligned where DSA[1:0] = 00.
If CHCTRLB*k*.WAS[2:0] = 010 and only word transfers are desired, the address must be word aligned where DSA[1:0] = 00.

26.10.9. Channel Source Cell Stride Size Register

Name: CHSSTRDk
Offset: 0x70 + k*0x50 [k=0..11]
Reset: 0x00000000
Property: PAC Write-Protection

CHCTRLA*k*.ENABLE=1 write protected.

Table 26-23. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
	SSTRD[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
	SSTRD[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 – SSTRD[15:0] Source Cell Stride Size

This value provide in this register is added to the last address of the cell transfer to determine the address of the next cell to read.
Next Cell Start Address = Current Cell Start Address + CHXSIZk.CSZ + SSTRD
`0xFFFF` =65,535 byte source cell stride size
`0x0001` =1 byte source cell stride size
`0x0000` = source cell stride defaults internally to the same value as CHXSIZk.CSZ.
If CHCTRLB*k*.RAS[2:0] = 001 or 100, the stride size must be halfword aligned where SSTRD[0] = 0.
If CHCTRLB*k*.RAS[2:0] = 101, the stride size must be word aligned where SSTRD[1:0] = 00.
If CHCTRLB*k*.RAS[2:0] = 010 and only word transfers are desired, the stride size must be word aligned where SSTRD[1:0] = 00.

26.10.10. Channel Destination Cell Stride Size Register

Name: CHDSTRDk
Offset: 0x74 + k*0x50 [k=0..11]
Reset: 0x00000000
Property: PAC Write-Protection

CHCTRLA*k*.ENABLE=1 write protected.

Table 26-24. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
	DSTRD[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
	DSTRD[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 – DSTRD[15:0] Destination Cell Stride Size
This value provide in this register is added to the last address of the cell transfer to determine the address of the next cell to read.
Next Cell Start Address = Current Cell Start Address + CHXSIZk.CSZ + DSTRD `0xFFFF` =65,535 byte destination cell stride size
`0x0000` = destination cell stride defaults to the same value as CHXSIZk.CSZ.
If CHCTRLB*k*.WAS[2:0] = 001 or 100, the stride size must be halfword aligned where DSTRD[0] = 0.
If CHCTRLB*k*.WAS[2:0] = 101, the stride size must be word aligned where DSTRD[1:0] = 00.
If CHCTRLB*k*.WAS[2:0] = 010 and only word transfers are desired, the stride size must be word aligned where DSTRD[1:0] = 00.

26.10.11. Channel Transfer Size Register

Name: CHXSIZk
Offset: 0x78 + k*0x50 [k=0..11]
Reset: 0x00000000
Property: PAC Write-Protection

CHCTRLA*k*.ENABLE=1 write protected.

Table 26-25. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	BLKSZ[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
	BLKSZ[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
							CSZ[9:8]	
Access							R/W	R/W
Reset							0	0

Bit	7	6	5	4	3	2	1	0
	CSZ[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 – BLKSZ[15:0] Block Transfer Size in Bytes
Sets the total number of bytes to transfer in the block.
0xFFFF = 65,535 bytes
0xFFFE = 65,534 bytes
***
0x0001 = 1 byte
0x0000 = 65,536 byte
The block size must align to the most restrictive address mode selected by RAS[2:0] or WAS[2:0].
CHCTRLBk .WAS[2:0] or CHCTRLBk.RAS[2:0] = 001 or 100, BLKSZ must be halfword aligned where BLKSZ[0] = 0.
CHCTRLBk .WAS[2:0] or CHCTRLBk.RAS[2:0] = 101, BLKSZ must be word aligned where BLKSZ[1:0] = 00.
If CHCTRLBk.WAS[2:0] or CHCTRLBk.RAS[2:0] = 010 and only word transfers are desired, BLKSZ must be word aligned where BLKSZ[1:0] = 00.

Bits 9:0 – CSZ[9:0] Cell Transfer Size in Bytes
Set the size of a cell transfer, which is the smallest amount of data transferred from source to destination on a start trigger.
0x3FF = 1023 byte cell transfer size

0x3FE = 1022 byte cell transfer size
0x001 = 1 byte cell transfer size
0x000 = 1024 byte cell transfer size
The cell size must align to the most restrictive address mode selected by RAS[2:0] or WAS[2:0].
CHCTRLBk.WAS[2:0] or CHCTRLBk.RAS[2:0] = 001 or 100, CSZ must be halfword aligned where CSZ[0] = 0.
CHCTRLBk.WAS[2:0] or CHCTRLBk.RAS[2:0] = 101, CSZ must be word aligned where CSZ[1:0] = 00.
If CHCTRLBk.WAS[2:0] or CHCTRLBk.RAS[2:0] = 010 and only word transfers are desired, CSZ must be word aligned where CSZ[1:0] = 00.

26.10.12. Channel Pattern Match Data Register

Name: CHPDATk
Offset: 0x7C + k*0x50 [k=0..11]
Reset: 0x00000000
Property: PAC Write-Protection

CHCTRLA*k*.ENABLE=1 write protected.

Table 26-26. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	PIGN[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
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	PDAT[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
	PDAT[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 – PIGN[7:0] Channel Pattern Ignore Value
When in Pattern Terminate Mode, any byte matching these bits during a pattern match may be ignored during the pattern match determination when PIGNEN is set. If a byte is read that is identical to this data byte the pattern match logic will treat it as a don't care when the pattern matching logic is enabled and PIGNEN bit is set.

Bits 15:0 – PDAT[15:0] Channel Pattern Match Data
Channel pattern match data to terminate the ongoing block transfer or linked-list.
PDAT[15:8] (the second byte of data, if enabled by PATLEN == 1) and PDAT[7:0] (the first byte of data) are to be matched with transferred data in order to allow terminate block transfer or linked-list.

26.10.13. Channel Control CRC Register

Name: CHCTRLCRCk
Offset: 0x80 + k*0x50 [k=0..11]
Reset: 0x00000000
Property: PAC Write-Protection

CHCTRLA*k*.ENABLE=1 write protected.

Table 26-27. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	CRCRIN	CRCROUT	CRCXOR		CRCAPP	CRCMD[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 – CRCRIN CRC Reflect Input Selection

This option is sometimes referred to as Reflected Byte or Reflected Input (RefIn).
This register is ignored if CHCTRLBk.CRCEN=0, otherwise CRCRIN provides the following functions.

Value	Description
0	Bytes are not reflected and are processed as read from the Source location.
1	Each byte is reflected bit-wise before being processed by the CRC engine.

Bit 6 – CRCROUT CRC Reflected Output Mode

This register is ignored if CHCTRLBk.CRCEN=0, otherwise CRCROUT provides the following functions.

Value	Description
0	CRC results are read back in the native bit order. If CRCAPP=1, the value appended to the end of the block is in the native bit order.
1	The CRC result are read back in reverse bit order. If CRCAPP=1, the value appended to the end of the block is in reverse bit order.

Bit 5 – CRCXOR CRC XOR Mode

This register is ignored if CHCTRLBk.CRCEN=0, otherwise CRCXOR provides the following functions.

Value	Description
0	CRC results are read back without XOR'ing. If CRCAPP=1, the appended value not XOR'ed.
1	CRC results are read back after being XOR'ed with 1's. This is the equivalent of XOR'ing the 16-bit CRC value with 0xFFFF or the 32-bit CRC value with 0xFFFF_FFFF. If CRCAPP=1, the value appended to the end of the block is result of the XOR.

Bit 3 – CRCAPP CRC Append Mode

This register is ignored if CHCTRLBk.CRCEN=0, otherwise CRCAPP provides the following functions.

Value	Description
0	The DMA transfers data from the source, re-orders it according to CHCTRLBk.BYTORD[1:0], drives it through the CRC and AFTER that writes the data to destination obeying WBOEN (Write Byte Order Enable) either re-ordered or unchanged. The resulting CRC is not appended but is available in the CHCRCDAT register.
1	The DMA transfers data from the source, re-orders it according to CHCTRLBk.BYTORD[1:0], drives it through the CRC and AFTER that writes the data to destination obeying WBOEN (Write Byte Order Enable) either re-ordered or unchanged. The DMA then writes the final calculated CRC at the end of the block.

Bits 2:0 – CRCMD[2:0] CRC/Checksum Mode

This register is ignored if CHCTRLBk.CRCEN=0, otherwise CRCMD provides the following functions.

Value	Description
111	Calculate an IP Header Checksum
110	Calculate CRC based on the 32-bit polynomial provided in register CRCPOLYB 101 =Calculate CRC based on the 32-bit polynomial provided in register CRCPOLYA 100 =CRC-32 (0x04C11DB7)
011	Calculate CRC based on the 16-bit polynomial provided in register CRCPOLYB[15:0] 010 =Calculate CRC based on the 16-bit polynomial provided in register CRCPOLYA[15:0] 001 =CRC-16 CCITT (0x1021)
000	CRC-16, also known as CRC-16-IBM and CRC-16-ANSI (0x8005)

26.10.14. Channel CRC/Checksum Data Register

Name: CHCRCDATk
Offset: 0x84 + k*0x50 [k=0..11]
Reset: 0x00000000
Property: PAC Write-Protection

CHCTRLA*k*.ENABLE=1 write protected.

Table 26-28. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	CRCDAT[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
	CRCDAT[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
	CRCDAT[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
	CRCDAT[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 – CRCDAT[31:0] CRC Data

Writing to this register will seed the CRC/Checksum generator.
Reading from this register will return the current value of the CRC/checksum.
If CHCTRLCRC.CRCMD is set to the IP header checksum mode, only the lower 16-bits contain information; the upper 16-bits are always read back zero. Data written to this register is converted and read back in one's complement form. (i.e. current checksum value).
If CHCTRLCRC.CRCMD is set to a 16-bit CRC mode, the lower 16-bits contain CRC value; the upper 16-bits are always zero. If CHCTRLCRC.CRCXOR is set, read back provides the 1's complement of the CRC value. If CHCTRLCRC.CRCROUT is set, the lower 16-bits are read back in reverse order.
If CHCTRLCRC.CRCMD is set to a 32-bit CRC mode, the register contains the 32-bit CRC value.
If CHCTRLCRC.CRCXOR is set, read back provides the 1's complement of the CRC value. If CHCTRLCRC.CRCROUT is set, the 32-bit value is read back in reverse bit order.

26.10.15. Channel Next Descriptor Address Pointer

Name: CHNXTk
Offset: 0x88 + k*0x50 [k=0..11]
Reset: 0x00000000
Property: PAC Write-Protection

CHCTRLA*k*.ENABLE=1 write protected.

Table 26-29. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	NXT[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
	NXT[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
	NXT[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
	NXT[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 – NXT[31:0] Channel Address Pointer to Next Descriptor
This register contains the physical address of the next descriptor to load. Unless set to a NULL pointer, the lower two bits, [1:0], of this register should always be written to 0's. If the value is set to 0xFFFF_FFFF, the DMA interprets the address as a NULL pointer.

26.10.16. Channel Linked List Configuration Status Register

Name: CHLLCFGSTATk
Offset: 0x8C + k*0x50 [k=0..11]
Reset: 0x00000000
Property: PAC Write-Protection

This channel resets on a channel reset.

Table 26-30. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
							CRCDAT	CTRLCRC
Access							R	R
Reset							0	0

Bit	7	6	5	4	3	2	1	0
	PDAT	XSIZ	DSTRD	SSTRD	DSA	SSA	EVCTRL	CTRLB
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 9 – CRCDAT CRC Data Descriptor Load

Value	Description
0	Do not change the field CRCDAT.
1	CHCRCDAT*k*.CRCDAT[31:0] is loaded from memory location BDCRCDAT

Bit 8 – CTRLCRC Control CRC Descriptor Load

Value	Description
0	Do not change the field CTRLCRC.
1	CHCTRLCRC*k*is loaded from memory location BDCTRLCRC.

Bit 7 – PDAT Match Pattern Descriptor Load

Value	Description
0	Do not change Registers fields PDAT or PIGN.
1	CHPDAT*k*.PDAT[15:0] is loaded from memory location BDPDATA[15:0] and PIGN[7:0] is loaded from memory location BDPDAT[31:24].

Bit 6 – XSIZ Transfer Size Descriptor Load

Value	Description
0	Do not change BLKSZ and CSZ.
1	CHXSIZk.BLKSZ and CHXSIZk.CSZ are loaded from BDXSIZ

Bit 5 – DSTRD Destination Cell Stride Size Descriptor Load

Value	Description
0	Do not change DSTRD
1	CHDSTRDk.DSTRD is loaded from memory location BDDSTRD

Bit 4 – SSTRD Source Cell Stride Size Descriptor Load

Value	Description
0	Do not change SSTRD
1	CHSSTRDk.SSTRD is loaded from memory location BDSSTRD

Bit 3 – DSA Destination Start Address Descriptor Load

Value	Description
0	Do not change DSA
1	CHDSA*k*.DSA is loaded from memory location BDDSA

Bit 2 – SSA Source Start Address Descriptor Load

Value	Description
0	Do not change SSA
1	CHDSA*k*.SSA is loaded from memory location BDSSA

Bit 1 – EVCTRL EVCTRL Register Descriptor Load

Value	Description
0	Do not change CHEVCTRLk
1	Control Register CHEVCTRLk is loaded from memory location BDEVCTRL

Bit 0 – CTRLB CTRLB Register Descriptor Load

Value	Description
0	Do not change CHCTRLBk
1	Control Register CHCTRLBk is loaded from memory location BDCTRLB

26.10.17. Channel Status Block Count Register

Name: CHSTATBCk
Offset: 0x90 + k*0x50 [k=0..11]
Reset: 0x00000000
Property: PAC Write-Protection

Offset is k=0..DMA_CH_N-1)

Table 26-31. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
								BBTC[16]
Access								R/W
Reset								0

Bit	15	14	13	12	11	10	9	8
	BBTC[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
	BBTC[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 16:0 – BBTC[16:0] Bytes Transferred in the Block Counter
Reports the number of bytes transferred in the block.
`0x10000` = 65,536 bytes transferred
`0x00001` = 1 bytes transferred
`0x00000` = 0 bytes transferred

26.10.18. Channel Status Cell Count Register

Name: CHSTATCCk
Offset: 0x94 + k*0x50 [k=0..11]
Reset: 0x00000000
Property: PAC Write-Protection

Offset is k=0..DMA_CH_N-1)

Table 26-32. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
						CBTC[10:8]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	7	6	5	4	3	2	1	0
	CBTC[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 10:0 – CBTC[10:0] Bytes Transferred in the Cell Counter
Reports the number of bytes transferred in the cell.
`0x400` = 1024 bytes transferred
`0x001` = 1 byte transferred
`0x000` = 0 bytes transferred

26.10.19. Channel Status Cell Count Register

Name: CHSTATk
Offset: 0x98 + k*0x50 [k=0..11]
Reset: 0x00000000
Property: PAC Write-Protection

Offset is k=0..DMA_CH_N-1)

Table 26-33. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						DREAD	CELLBUSY	BLKBUSY
Access						R	R	R
Reset						0	0	0

Bit 2 – DREAD Descriptor Read Status Bit
Note: After an EV abort occurs, a software reset should be issued to clear this bit.

Value	Description
0	Descriptor has not been read or is not available to read.
1	Descriptor read and loaded into channel registers.

Bit 1 – CELLBUSY Channel Cell Transfer Busy Status Bit

Value	Description
0	Channel is idle
1	Channel is performing a cell transfer

Bit 0 – BLKBUSY Channel Block Transfer Busy Status Bit

Value	Description
0	Channel is idle.
1	Channel is performing a block transfer. Setting CHCTRLAk.ENABLE=0 will suspend the block transfer. On a channel reset, this bit will clear at the completion of the reset sequence. Software can poll this bit to determine when the channel has finished resetting.

27. Supply Controller (SUPC)

27.1. Overview

The Supply Controller (SUPC) manages the voltage reference, power supply, and supply monitoring of the device. It is also able to control multiple output pins named “SUPC_OUT[1:0]”.

The SUPC controls all the voltage regulators supplying the multiple 1.2v domain. It sets the voltage regulators according to the sleep modes, or the user configuration.

The SUPC controls several Brown-out Reset (BOR): BOR_VDDREG monitors the voltage applied to the regulators (VDDREG), BOR_VDDIO monitors the voltage applied to I/Os, BOR_VDDA monitors the voltage applied to VDDA. In Backup mode, a dedicated Duty Cycle BOR (DCBOR) is able to monitor VDDREG and VDDIO/VDDA. BOR_VDDUSB monitors the voltage applied to the VUSB3V3. PORCORE tightly coupled with the capless regulator guarantees the internal voltage of the core (VDDCORE).

The SUPC controls a configurable Low-Voltage Detector used to monitor VDDIO in both directions.

The SUPC controls an internal reference voltage and a current dependent on the temperature which can be used by analog modules like the ADC. The SUPC also controls the device’s charge pumps.

Note: VDDA and AVDD are used interchangeably.

27.2. Features

The SUPC controls the following analog supply elements:

- Voltage Regulator System
  - Main voltage regulators: LDO in Active, Standby, or Hibernate mode. Used for VDDCORE_SW domain
  - Voltage regulator called VREGRAM used for VDDCORE_RAM domain and PLL
  - Low-Power voltage regulator in Backup mode (LPVREGC)
  - Additional capless regulator for USB transceivers (VREGUSB)
- Voltage Reference System (Bandgap)
  - Reference voltage for ADC
  - Temperature sensor
  - Charge Pump for I/O pad and analog cells as PTC/AC/ADC in case of low VDD voltage
- 3.3V Brown-out Reset (BOR) Detector
  - Three instances of BOR are used when calibrated to monitor VDDIO/VDDA and VDDREG power supply voltages, during power-up, Active mode and Standby mode
  - Programmable threshold value loaded from USER CFG page at startup
  - Triggers resets
- 3.3V Low-Power Brown-out Reset (DCBOR)
  - Used in Backup mode to monitor VDDIO/VDDA and VDDREG
  - Threshold values loaded from USER CFG page of CFM Flash Memory
  - Triggers resets
  - Operating modes: Continuous mode and Sampled mode (with programmable sampling frequency)
- 1.2V PORCORE Detector
  - Monitors VDDCORE power supply voltage
  - Tightly coupled with the capless regulator

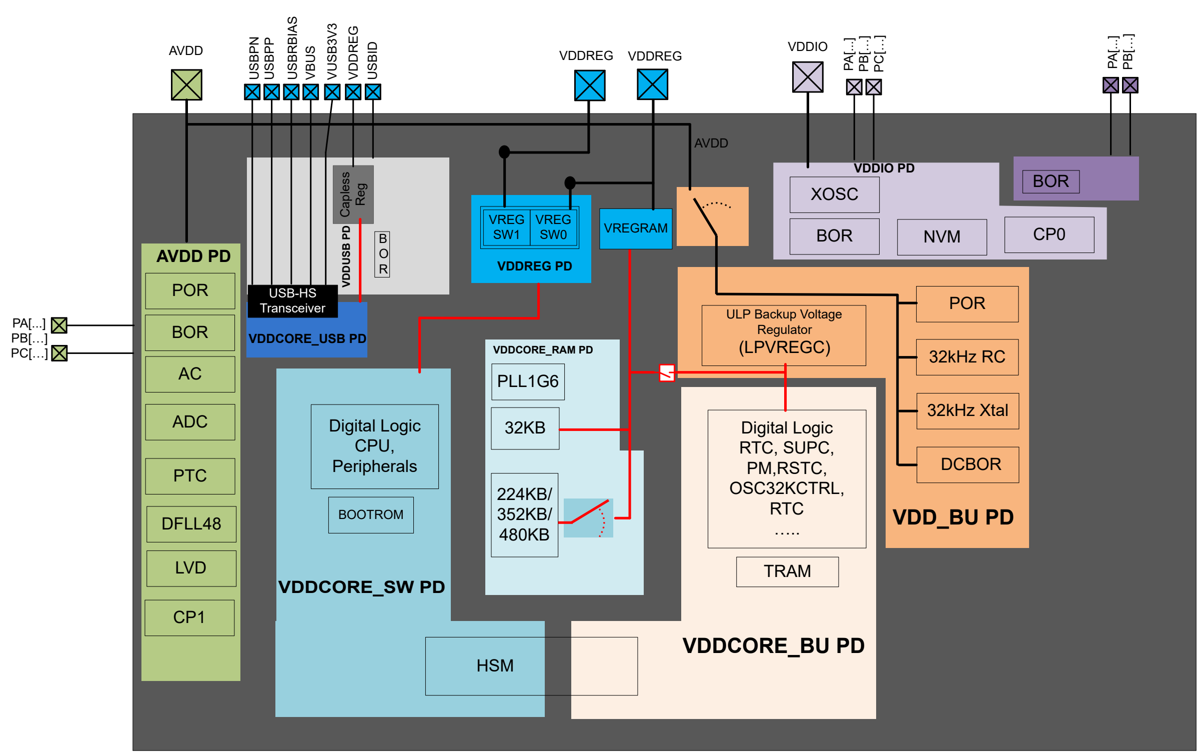
- Triggers resets
- 3.3V Programmable Low-voltage Detector
  - Monitors VDDIO
  - Configurable threshold and direction
  - Can trigger Interrupt
- Output pins
  - Pin toggling on RTC event or by SUPC in Backup mode

27.3. Block Diagram

As shown in the following figure, there are several power supply pins:

- AVDD – Powers the Backup power domain as well as the ADC, Analog Comparator (AC), and the Peripheral Touch Controller (PTC).
- VDDREG – Powers the internal regulators.
- VDDIO – Powers I/O lines, an External Crystal Oscillator (XOSC), two charge pumps which support the Analog Comparator (AC) and the ADC on the device (ADC and PTC ADC). Most VDDIO pins also power the device's Flash panel through double bonding.
- VDDUSB3V – Powers the USB ports on the device.

Figure 27-1. Supply Controller Block Diagram



Power domains (PDs) shown above are not independent. VDDCORE_*, VDDIO, and VDDREG share the same ground, VSS. However, AVDD has its own ground, AVSS. AVDD and VDDIO inputs must share the same supply, VDD.

The block diagram above also shows a variety of internally regulated power domains (VDDCORE PDs), nominally at 1.2V:

- **VDDCORE_BU:** Powers the backup domain. It contains peripherals that remain powered in the Backup Sleep mode.

- **VDDCORE_SW:** The main voltage domain for the CPU, bus, and most peripherals, which can be switch off (the _SW suffix).
- **VDDCORE_USB:** Domain for the USB port.
- **VDDCORE_RAM:** This domain is used to retain the devices SRAM.

The device's internal main voltage regulators have three different modes, controlled by the Supply Controller (SUPC):

- **Active Run Mode:** The default mode when the CPU and peripherals are running.
- **Idle/Standby:** When the CPU and peripherals are in Standby mode.
- **Hibernate/Backup/Off Mode:** When the chip is in Backup mode, the internal regulator is OFF, the VDDCORE_SW core power domain is OFF. The VDDCORE_BU backup domain is powered by the backup regulator (Low-Power Voltage Regulator for Core - LPVREGC).

27.4. Signals Description

Table 27-1. Signals Description

Signal Name	Type	Description
SUPC_OUT0	Digital Output	SUPC Output 0
SUPC_OUT1	Digital Output	SUPC Output 1

27.5. Peripheral Dependencies

Peripheral Name	Base Address	NVIC IRQ Index: Source	PAC Peripheral Identifier (PAC.WRCTRL.PERIDx)	EVSYS Generator (EVSYS.CHANNELn.EVGENx)	Power Domain
SUPC	0x4400_8000	4 : LVDET, LVDRDY, BORVDDUSB_0, ADDVREGRDY_0	4	1:LVDET	VDDCORE_BU

27.6. Functional Description

27.6.1. Voltage Regulators

27.6.1.1. Enabling, Disabling, and Resetting

The main regulators output supply level is automatically defined by the Sleep mode selected in the Power Manager module.

Additional regulators are disabled by default and can be enabled and disabled by writing the corresponding AVREGEN bit in the VREGCTRL register. There is one additional regulator used to power the USB.

27.6.1.2. VDDCORE Control

The VDDCORE supplies (VDDCORE_SW, VDDCORE_RAM, VDDCORE_BU, VDDCORE_USB if enabled in Standby mode).

27.6.1.3. Additional Regulator Voltage Control

Additional regulator voltage level is off while in Sleep mode equal or deeper than Standby mode to reduce consumption, meaning that domain driven by this regulator (USB) cannot be used in Standby mode. This default behavior can be changed by configuring the ”Additional Voltage Regulator Configuration” bits field in the VREGCTRL CONTROL register (VREGCTRL.AVREGCFG).

27.6.1.4. Charge Pump for Low VDDIO Voltage

This is highly recommended to activate charge pumps when VDDA/VDDIO is too low. The control of Charge pump *n* (*n* = 0,1) is managed by setting the corresponding bit in VREGCTRL.CPEN[1:0]. When CPEN[*n*] bit is set, the enable and auto-enable bits of the charge pump[N] are set. In standby mode, the charge pumps are automatically turned OFF except if a consumer (PTC, ADC or AC) of the charge pump is requesting the charge pump.

27.6.2. Voltage References

The reference voltages are generated by a functional analog block controlled by the SUPC. It is providing a fixed-voltage source VREF for modules such as ADC and AC as well as a temperature dependent voltage in support of a temperature sensor.

27.6.3. Output and Input Pins

The SUPC can drive two outputs. By writing a '1' to the corresponding Output Enable bit in the Backup Output Control register (BKOUT.EN), the SUPC_OUT*n* pin is driven by the SUPC.

The OUT pin can be set by writing a '1' to the corresponding Set Output bit in the Backup Output Control register (BKOUT.SETx). The OUT pin can be cleared by writing a '1' to the corresponding CLR bit (BKOUT.CLRx).

If the “Toggle Output Mode X is set to “RTCTGL”, the corresponding SUPC_OUT*n* pin will toggle when an RTC event occurs. Note that before configuring this mode, this is recommended to configure the RTC module properly. If the “Toggle Output Mode X is set to “BKUPTGL”, the corresponding SUPC_OUT*n* pin will toggle to one when the device goes to Backup Mode. It will toggle to zero when exiting Backup Mode.

The BKIN register reflects the current state of the of the corresponding BKOUT pin if enabled (SUPC.BKOUT.EN = 1).

27.6.4. Brown-out Reset

- **BOR_VDDREG:** After power-up or user reset, the BOR monitoring VDDREG is active. Its hysteresis values are loaded from NVM calibration User Config (UCFG).
- **BOR_VDDIO:** After power-up or user reset, the BOR monitoring VDDIO is active. Its threshold and hysteresis values are loaded from NVM Calibration User Config (UCFG).
- **BOR_VDDA:** After power-up or user reset, the BOR monitoring VDDA is active. Its threshold and hysteresis values are loaded from NVM Calibration User Config (UCFG).
- **BOR_VDDUSB:** After power-up or user reset, the BOR monitoring VDDUSB is active. Its threshold and hysteresis values are not configurable.
- **DCBOR:** After power-up or user reset, the DCBOR used in backup mode is inactive. Its threshold and hysteresis values are loaded from NVM Calibration User Config (UCFG).

27.6.4.1. 3.3V Regular Brown-out Reset (VDDREG_BOR or VDDIO/VDDA_BOR)

In all modes except the Backup mode, the BOR_VDDREG and BOR_VDDIO/VDDA compare respectively the VDDREG and VDDIO/VDDA voltage with the VDDREG and VDDIO/VDDA brown-out threshold level. This level is set during power-up when NVM calibration values are loaded. When VDDREG or VDDIO/VDDA crosses below the brown-out threshold level, BOR generates a Reset.

27.6.4.2. 3.3V Regular Brown-out Reset (VDDUSB_BOR)

In all modes except the Backup mode, the BOR_VDDUSB compares the VDDUSB voltage with the brown-out threshold level. When VDDUSB crosses below the brown-out threshold level, it can generate an interrupt if enabled.

27.6.4.3. 3.3V Duty Cycle Brown-out Reset (DCBOR)

In Backup mode, the regular VDDREG_BOR, VDDIO_BOR and VDDA_BOR are off to save consumption. The DCBOR takes over to monitor the 3.3V power supplies.

To reduce power consumption, DCBOR is used in sampling mode. The Sampling Mode is a low-power mode where the DCBOR is being repeatedly enabled on a sampling clock. The DCBOR 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. The frequency of the clock is controlled by the Prescaler Select bit groups in the BOR register (BOR.DCBORPSEL). Refer to “BOR SFR" register for more details.

In Backup mode, the DCBOR alternates monitoring VDDREG and VDDIO/VDDA. When VDDREG or VDDIO/VDDA crosses below the brown-out threshold level, the BOR generates a Reset.

27.6.4.4. BOR Filtering

For BOR reset generation we have an input filter to filter-out any glitches which are less than 100nS. Supplementary BOR filtering is also possible for BOR_VDDA, BOR_VDDIO, and BOR_VDDREG by using the BOR.BORFILT bitfield. This filtering does not apply for the DCBOR used in the Backup mode.

27.6.5. Programmable Low Voltage Detector

The programmable LVD is able to monitor the VDDIO input power supply.

Before enabling the LVD by setting LVD.ENABLE bit, the threshold level bits field (LVD.LEVEL) and the direction (LVD.DIR) have to be configured. The LVD register is Enable-Protected, meaning that it can only be written when the LVD is disabled (LVD.ENABLE). As long as the Enable bits is ‘1’, any writes to LVD register will be discarded and a bus error will be generated. The Enable bits are not Enable-Protected.

27.6.5.1. Settling Time

After being enabled, a start-up delay is required before the result of the comparison is ready. During the startup-time, the output value is zero (STATUS.LVDET).

After the start-up time has passed, the LVDET ready bit is set (STATUS.LVDRDY) indicating that STATUS.LVDET bit is available, and the appropriate LVD interrupt flag or output event are also generated. New comparisons are performed continuously until LVD.ENABLE bit is written to zero. The start-up time applies only to the first comparison. If LVD is disabled (LVD.ENABLE is written to 0), the STATUS.LVDRDY bit will take time to fall because of resynchronization. As a consequence, this is forbidden to re-enable immediately the LVD if the STATUS.LVDRDY field is still one.

27.6.5.2. LVD Interrupt and Event

• **LVD Output Interrupt source:** LVD Output interrupt source status (STATUS.LVDET) is set to one when the VDDIO voltage value crosses the programmed threshold level (LVD.LEVEL) in the right direction (LVD.DIR). It is reset to zero when the VDDIO voltage value crosses the threshold level in the wrong direction (LVD.DIR). Note that if the LVD is just enabled and the voltage is already below the threshold (if DIR=1) or the voltage is already above the threshold (if DIR=0), then the LVD output interrupt is also generated.

• **LVD Output event:** The LVD can generate output event when the VDDIO voltage value crosses the programmed threshold level (LVD.LEVEL) in the right direction (LVD.DIR). To enable this feature, the LVD.OEVEN bit has to be set to one. Refer to the EVSYS Event System for details on configuring the event system.

27.6.5.3. LVD Behavior in Sleep Mode

LVD is off in hibernate and backup sleep modes.

LVD is automatically turned OFF in standby mode except if the LVD.RUNSTDBY bit is set.

27.6.6. Interrupts

The SUPC has the following interrupt sources, which are either synchronous or asynchronous wake-up sources:

- LVD Ready (LVDRDY), synchronous
- BOR VDDUSB (BORVDDUSB), asynchronous
- Additional Regulator Ready (AVDDREGRDY), synchronous
- LVD Detection (LVDDET), 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 via INTENSET. 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.

27.6.7. Power Management

The SUPC can operate in all sleep modes except backup sleep mode.

27.6.8. 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.

27.6.9. Register Access Protection

Registers with write-access can be write-protected optionally by the Peripheral Access Controller (PAC). Here is the list of the write-protected registers:

- INTENCLR
- INTENSET
- BOR
- LVD
- VREGCTRL
- VREFCTRL
- BKOUT

27.6.10. Temperature Calibration

27.6.10.1. Summary

With calibration, it is possible to achieve +/-5% performance from the onboard bandgap temperature sensor. This measurement can provide important information about the die temperature that may be critical in an operating environment with elevated ambient temperatures. Refer to the Electrical Characteristics section of the data sheet for maximum die temperature. Refer to the *Power and Temperature Consideration*s section.

The following table of calibration fuses and mathematical representations are a summary of information provided in the MPLAB Harmony v3 application example located in *<root>\csp_apps_pic32cz_ca\apps\temp_sense*. The values provided in these fuses result from a calibration procedure that is included in the final test program for each individual part. Some of these values are unique for each individual part which necessitates the implementation of the mathematical routines described below and more comprehensively in the document mentioned above.

Table 27-2.

OTP Cal Fuse Name	Address	Bitfield Name	Bitmask	Default	Description
FUSES_FCCFG69_REG	0x0A007194	ROOM_ADC_VAL_PTAT	0x00000fff	992	12-bit ADC conversion at room temperature of PTAT in LSB
-	-	ROOM_ADC_VAL_BUFF1V2	0x00fff000	1489	12-bit ADC conversion at room temperature of Buffered 1.2V in LSB
-	-	ROOM_TEMP_VAL_INT	0xff000000	25	Integer part of room temperature in °C
FUSES_FCCFG70_REG	0x0A007198	ROOM_TEMP_VAL_DEC	0x0000000f	0	Decimal part of room temperature in 0.1°C
-	-	ROOM_VREF_VAL	0x0000fff0	3300	voltage in mV of VREF = VDDANA
-	-	HOT_TEMP_VAL_DEC	0x000f0000	0	Decimal part of hot temperature in 0.1°C
-	-	Reserved	0xfff00000	-	-
FUSES_FCCFG71_REG	0x0A00719C	HOT_ADC_VAL_PTAT	0x00000fff	1325	12-bit ADC conversion at hot temperature in LSB of PTAT in LSB
-	-	HOT_ADC_VAL_BUFF1V2	0x00fff000	1489	12-bit ADC conversion at hot temperature of Buffered 1.2V in LSB
-	-	HOT_TEMP_VAL_INT	0xff000000	125	Integer part of hot temperature in °C

27.6.10.2. Slope Calculation

T_{room} (°C): Floating point combination of calibrations values ROOM_TEMP_INT and ROOM_TEMP_DEC representing the ambient temperature calibration point.

T_{hot} (°C): Floating point combination of calibrations values HOT_TEMP_INT and HOT_TEMP_DEC representing the hot temperature calibration point.

$\Delta_{ptat\text{-}cal}$ (bit): Raw bit value difference of the calibration value HOT_ADC_VAL_PTAT - ROOM_ADC_VAL_PTAT.

$V_{ref\text{-}cal}$ (mV): Calibration reference voltage ROOM_VREF_VAL in mV.

$\frac{dV}{dT}\left(\frac{V}{^oC}\right) - slope$: Change of voltage with respect to temperature.

$$\frac{dV}{dT}\left(\frac{V}{^oC}\right) = \frac{V_{ref-cal}}{(1000)(4095)}\left(\frac{V}{bit}\right) \bullet \frac{\Delta_{ptat-cal}}{T_{hot} - T_{room}}\left(\frac{bit}{^oC}\right)$$

27.6.10.3. Die Temperature Calculation

N_{adc} (bit): Raw bit value of current ADC read.

$\Delta_{ptat\text{-}app}$ (bit): Raw bit value difference between HOT_ADC_VAL_PTAT (Vref ratio compensated) and N_{ADC}.

$\Delta_{ptat\text{-}app}$ (bit) = HOT_ADC_VAL_PTAT – N_{adc} * ($V_{ref\text{-}app}$ / $V_{ref\text{-}cal}$)

$V_{ht\text{-}cal}$ (V): Voltage representation of hot temperature calibration value.

$$\left(V\right) = \frac{dV}{dT}\left(\frac{V}{^oC}\right) \bullet T_{hot}\left(^oC\right)$$

$V_{ref\text{-}app}$ (mV): Application reference voltage in mV

$V_{diff\text{-}app}$ (V): Voltage representation of the raw bit value difference between the hot temperature calibration reference and the current ADC read value (compensated for the V_{ref} ratio already happened, hence $V_{ref\text{-}cal}$ is used).

$$V_{diff-app}\left(V\right) = \frac{V_{ref-cal}}{(1000)(4095)}\left(\frac{V}{bit}\right) \bullet \Delta_{ptat-app}\left(bit\right)$$

T_j: Junction (die) temperature.

$$T_j\left({}^{o}C\right)\frac{\left(V_{ht-cal} - V_{diff-app}\right)}{\left(\frac{dV}{dT}\right)}$$

27.7. Register Summary

For descriptions and definitions of both Register and bitfield properties, refer to Register Properties.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	INTENCLR	31:24								
		23:16								
		15:8								ADDVREGRDY
		7:0			BORVDDUSB				LVDRDY	LVDET
0x04	INTENSET	31:24								
		23:16								
		15:8								ADDVREGRDY
		7:0			BORVDDUSB				LVDRDY	LVDET
0x08	INTFLAG	31:24								
		23:16								
		15:8								ADDVREGRDY
		7:0			BORVDDUSB				LVDRDY	LVDET
0x0C	STATUS	31:24								
		23:16								
		15:8								ADDVREGRDY
		7:0			BORVDDUSB				LVDRDY	LVDET
0x10	SYNCBUSY	31:24								
		23:16								
		15:8								
		7:0								BOR
0x14	BOR	31:24								
		23:16								
		15:8							BORFILT[1:0]	
		7:0		DCBORPSEL[2:0]						ACTION
0x18	LVD	31:24								
		23:16					LEVEL[3:0]			
		15:8								
		7:0				RUNSTDBY	OEVEN	DIR	ENABLE	
0x1C	VREGCTRL	31:24								AVREGSTDBY
		23:16								AVREGEN
		15:8							CPEN[1:0]	
		7:0			LVHIB	LVSTDBY		OFFSTDBY		
0x20	VREFCTRL	31:24								
		23:16								
		15:8								
		7:0				TSEN			LPHIB	LPSTDBY
0x24 ... 0x27	Reserved									
0x28	BKOUT	31:24					TGLOM1[1:0]		TGLOM0[1:0]	
		23:16						SETn		SETn
		15:8						CLRn		CLRn
		7:0						ENn		ENn
0x2C	BKIN	31:24								
		23:16								
		15:8								
		7:0							BKIN[1:0]	

27.7.1. Interrupt Enable Clear

Name: INTENCLR
Offset: 0x0000
Reset: 0x00000000
Property: PAC Write-Protection

Table 27-3. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
								ADDVREGRDY
Access								R/W
Reset								0

Bit	7	6	5	4	3	2	1	0
			BORVDDUSB				LVDRDY	LVDET
Access			R/W				R/W	R/W
Reset			0				0	0

Bit 8 – ADDVREGRDY Additional Regulator Ready Interrupt Enable Clear
Writing a zero to this bit has no effect.
Writing a one to the bit disables the ADDVREGRDY interrupt.
Each bit will read as the current value of the ADDVREGRDY interrupt enable.

Bit 5 – BORVDDUSB BORVDDUSBn Interrupt Enable Clear
Writing a zero to this bit has no effect.
Writing a one to the bit disables the BORVDDUSB interrupt.
Each bit will read as the current value of the BORVDDUSB interrupt enable.

Bit 1 – LVDRDY Low Voltage Detector Ready Interrupt Enable Clear
Writing a zero to this bit has no effect.
Writing a one to this bit disables the LVDRDY interrupt.
This bit will read as the current value of the LVDRDY interrupt enable.

Bit 0 – LVDET Low Voltage Detector Interrupt Enable Clear
Writing a zero to this bit has no effect.
Writing a one to this bit disables the LVDET interrupt.
This bit will read as the current value of the LVDET interrupt enable.

27.7.2. Interrupt Enable Set

Name: INTENSET
Offset: 0x0004
Reset: 0x00000000
Property: PAC Write-Protection

Table 27-4. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
								ADDVREGRDY
Access								R/W
Reset								0

Bit	7	6	5	4	3	2	1	0
			BORVDDUSB				LVDRDY	LVDET
Access			R/W				R/W	R/W
Reset			0				0	0

Bit 8 – ADDVREGRDY Additional Regulator Ready Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to the bit enables the ADDVREGRDY interrupt.
Each bit will read as the current value of the ADDVREGRDY interrupt enable.

Bit 5 – BORVDDUSB BORVDDUSB Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to the bit enables the corresponding BORVDDUSB interrupt.
Each bit will read as the current value of the BORVDDUSB interrupt enable.

Bit 1 – LVDRDY Low Voltage Detector Ready Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit enables the LVDRDY interrupt.
This bit will read as the current value of the LVDRDY interrupt enable.

Bit 0 – LVDET Low Voltage Detector Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit enables the LVDET interrupt.
This bit will read as the current value of the LVDET interrupt enable.

27.7.3. Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x0008
Reset: 0x00000000
Property: -

Note: Subsequent to an interrupt flag being cleared, the flag must be read back to verify the clear before exiting the ISR. Failure to do this can result in duplicate interrupts.

Table 27-5. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
								ADDVREGRDY
Access								R/W
Reset								0

Bit	7	6	5	4	3	2	1	0
			BORVDDUSB				LVDRDY	LVDET
Access			R/W				R/W	R/W
Reset			0				0	0

Bit 8 – ADDVREGRDY Additional Regulator Ready
Set by hardware when the Additional Regulator is ready and the output voltage is correct.
Write one to clear the bit.

Bit 5 – BORVDDUSB Brown-Out detected for a VDD_USB
Set by hardware when a Brown-Out has been detected on the corresponding VDD_USB.
Write one to clear the corresponding bit.

Bit 1 – LVDRDY Low Voltage Detector Ready
Set by hardware the Low Voltage Detector is ready to operate.
Write one to clear the corresponding bit.

Bit 0 – LVDET Low Voltage Detected
Set by hardware if VDDIO crosses the threshold voltage in a “good” direction according to LVD.DIR.
Write one to clear the corresponding bit.

27.7.4. Flag Status

Name: STATUS
Offset: 0x000C
Reset: 0x00000000
Property: -

Table 27-6. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
								ADDVREGRDY
Access								R
Reset								0

Bit	7	6	5	4	3	2	1	0
			BORVDDUSB				LVDRDY	LVDET
Access			R				R	R
Reset			0				0	0

Bit 8 – ADDVREGRDY Additional Regulator Status
Set by hardware when the voltage corresponding Additional Regulator is ok.

Bit 5 – BORVDDUSB BORVDDUSB Status
Set by hardware when VDD_USB is OK.

Bit 1 – LVDRDY Low Voltage Detector Ready Status
Set by hardware when the Low Voltage Detector is ready.

Bit 0 – LVDET Low Voltage Detector Status
Set to one if VDDIO crosses the threshold voltage in the “good” direction according to LVD.DIR.

27.7.5. Synchronization Busy

Name: SYNCBUSY
Offset: 0x0010
Reset: 0x00000000
Property: -

Table 27-7. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

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

Bit 0 – BOR BOR Register Synchronization Flag

When hardware sets this bit, no writes are permitted to the BOR register. After writing to the BOR register wait for this flag to clear before making additional writes.

27.7.6. BOR Control

Name: BOR
Offset: 0x0014
Reset: 0x00000000
Property: PAC Write Protection

Table 27-8. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
							BORFILT[1:0]	
Access							R/W	R/W
Reset							0	0

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

Bits 9:8 – BORFILT[1:0] BOR Filtering

Value	Description
0x0	No digital filtering (NOFILT)
0x1	32µs filtering (FILT32US)
0x2	125µs filtering (FILT125US)
0x3	250µs filtering (FILT250US)

Bits 6:4 – DCBORPSEL[2:0] Duty Cycle BOR Prescaler Select

Value	Description
0x0	Not Divided (NODIV)
0x1	Divide clock by 2 (DIV2)
0x2	Divide clock by 4 (DIV4)
0x3	Divide clock by 8 (DIV8)
0x4	Divide clock by 16 (DIV16)
0x5	Divide clock by 32 (DIV32)
0x6	Divide clock by 64 (DIV64)
0x7	Divide clock by 128 (DIV128)

Bit 0 – ACTION Action when Threshold Crossed

Value	Description
0x0	The BOR generates a reset ($\overline{\text{RESET}}$)
0x1	Reserved

27.7.7. LVD Control

Name: LVD
Offset: 0x0018
Reset: 0x00000000
Property: PAC Write Protection

Note: The LVD comparator is testing for a threshold on the VDDIO supply.

Table 27-9. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					LEVEL[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
				RUNSTDBY	OEVEN	DIR	ENABLE	
Access				R/W	R/W	R/W	R/W	
Reset				0	0	0	0	

Bits 19:16 – LEVEL[3:0] Threshold Level.
Refer to REG 52 in Power Supply.

Bit 4 – RUNSTDBY Run During Standby

Value	Description
0	Run during standby disabled
1	Run during standby enabled

Bit 3 – OEVEN Output Event Enable

Value	Description
0	Output events disabled
1	Output events enabled

Bit 2 – DIR Detection Direction

Value	Description
0	Rising detection
1	Falling detection

Bit 1 – ENABLE Enable Low Voltage Detection

Value	Description
0	Low Voltage Detection disabled
1	Low Voltage Detection enabled

27.7.8. Voltage Regulator System (VREG) Control

Name: VREGCTRL
Offset: 0x001C
Reset: 0x00000004
Property: PAC Write-Protection

Note: During normal operation, all voltage regulators that are in use must be left in the On state to allow for the proper transition between different low-power or standby states.

Table 27-10. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
								AVREGSTDBY
Access								R/W
Reset								0

Bit	23	22	21	20	19	18	17	16
								AVREGEN
Access								R/W
Reset								0

Bit	15	14	13	12	11	10	9	8
							CPEN[1:0]	
Access							R/W	R/W
Reset							0	0

Bit	7	6	5	4	3	2	1	0
			LVHIB	LVSTDBY		OFFSTDBY		
Access			R/W/HC	R/W/HC		R/W		
Reset			0	0		0		

Bit 24 – AVREGSTDBY Additional Voltage Regulator Configuration

Value	Description
`0x0`	USB regulator is off in Sleep, Standby, Hibernate or Backup mode.
`0x1`	USB regulator is ON in Standby mode if the AVREGEN bit is set. It is OFF in Hibernate or Backup mode.

Bit 16 – AVREGEN Additional Voltage Regulator Enabled

Value	Description
`0x0`	USB regulator is disabled (Default)
`0x1`	USB regulator is Enabled

Bits 9:8 – CPEN[1:0] Analog Peripheral Charge Pump Enabled

Value	Description	Requirements
0x0	All charge pumps disabled.	AVDD ≥ 2.5v

CPEN (continued)

Value	Description	Requirements
0x1	Enable charge pump for I/O analog mux and Analog Comparator (AC)	AVDD < 2.5v
---	Reserved	
0x3	Enable charge pumps for I/O, AC, ADC, and PTC	

Notes:

1. When AVDD < 2.5v the corresponding appropriate CPEN must be enabled.
2. Users must have previously enabled the charge pump clocks defined in Configuration Register 5, FUCFG5.

Bit 5 – LVHIB Low Voltage Hibernate Enable

Note: LVHIB must always be forced to 1.

Value	Description
`0x0`	In Hibernate mode, VDDCORE_BU and VDDCORE_RAM are set to 1.2v.
`0x1`	In Hibernate mode, VDDCORE_BU and VDDCORE_RAM are set to 0.8v

Bit 4 – LVSTDBY Low Voltage Standby Enable

Value	Description
`0x0`	In Standby mode, VDDCORE_BU, VDDCORE_RAM, VDDCORE_SW and optionally VDDCOREUSB/PLL are set to 1.2v.
`0x1`	In Standby mode, VDDCORE_BU, VDDCORE_RAM, VDDCORE_SW and operationally VDDCOREUSB/PLL are set to 0.8v.

Bit 2 – OFFSTDBY Off in Standby Control VREGSW 0 and 1

Value	Description
`0x0`	In Standby mode, VREGSW 0,1 are OFF
`0x1`	In standby mode, VREGSW 0,1 are ON

27.7.9. Voltage References System (VREF) Control

Name: VREFCTRL
Offset: 0x0020
Reset: 0x00000003
Property: PAC Write-Protection

Table 27-11. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
				TSEN			LPHIB	LPSTDBY
Access				R/W			R/W	R/W
Reset				0			1	1

Bit 4 – TSEN Temperature Sensor Output Enable

Value	Description
0	Temperature sensor output to ADC disabled
1	Temperature sensor output to ADC Enabled

Bit 1 – LPHIB Bandgap and Regulators Low Power Hibernate Enable
Note: LPHIB must always be forced to 0.

Value	Description
0x0	In hibernate mode, bandgap is set to nominal power mode. As a consequence, enabled regulator(s) are set to nominal power mode. (FullPower)
0x1	In hibernate mode, bandgap is set to low power mode. As a consequence, enabled regulator(s) are set to low power mode. (LowPower)

Bit 0 – LPSTDBY Bandgap and Regulators Low Power Standby Enable
Note: LPSTDBY must always be forced to 0.

Value	Description
0x0	In standby mode, bandgap and enabled regulators are set to nominal power mode. (Full Power)
0x1	In standby mode, bandgap and enabled regulators are set to low power mode. (Low Power)

27.7.10. Backup Output Control

Name: BKOUT
Offset: 0x0028
Reset: 0x00000000
Property: PAC Write-Protection

Note: These register bits are useful for indicating to users external application logic MCU Backup mode entry/exit.

Table 27-12. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
					TGLOM1[1:0]		TGLOM0[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
						SETn		SETn
Access						W		W
Reset						0		0

Bit	15	14	13	12	11	10	9	8
						CLRn		CLRn
Access						W		W
Reset						0		0

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

Bits 24:25, 26:27 – TGLOMn SUPC_OUTn Pin Toggle Output Mode for Backup Output n, n = 0,1

Value	Description
`0x0`	SUPC_OUTn pin output does not toggle (DISABLE).
`0x1`	SUPC_OUTn pin output toggles on RTC events (RTCTGL).
`0x2`	SUPC_OUTn pin output is set when the device enters Backup mode (BKUPTGL). The output should then be cleared by software using CLR*n.*
`0x3`	Reserved.

Bits 16,18 – SETn SUPC_OUTn Pin Set Output for Backup Output n, n = 0,1
Writing a '0' to a bit has no effect.
Writing a '1' to a bit will set the corresponding output.
Reading this bit returns '0'.

Bits 8,10 – CLRn SUPC_OUTn Pin Clear Output for Backup Output n, n = 0,1
Writing a '0' to a bit has no effect.
Writing a '1' to a bit will clear the corresponding output.
Reading this bit returns '0'.

Bits 0,2 – ENn SUPC_OUTn Pin Enable Output for Backup Output n, n = 0,1

Value	Description
0	SUPC_OUTn Backup Output pin disabled.
1	SUPC_OUTn Backup Output pin enabled.

27.7.11. Backup Input Control

Name: BKIN
Offset: 0x2C
Reset: 0x00000000
Property: -

Table 27-13. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

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

Bits 1:0 – BKIN[1:0] Backup Input Value

28. Power Manager (PM)

28.1. 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 satisfy 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 manually controls which power domains will be turned on and off in standby, hibernate, and backup sleep mode.

The application code decides which sleep mode to enter and when to enter the mode. Interrupts from enabled peripherals and all enabled reset sources can restore the device from a sleep mode to active mode.

In backup and hibernate mode, the PM allows retaining the state of the I/O lines, preventing I/O lines from toggling during wake-up.

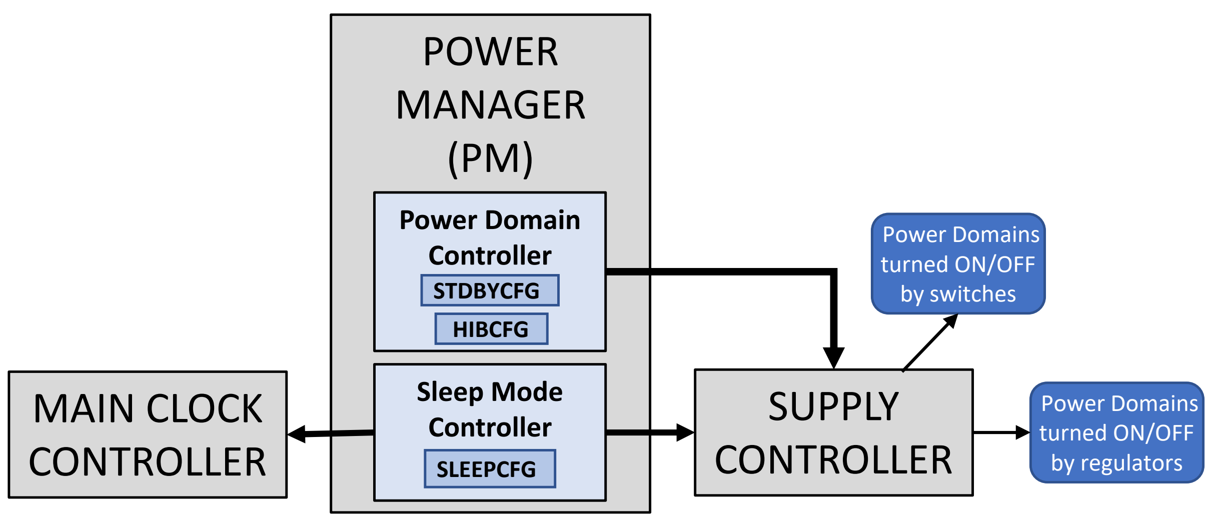
28.2. Features

The Power Management module provides:

- Idle, Standby, Hibernate, and Backup Sleep modes as well as Off mode
- Sleepwalking in Standby mode
- I/O line state retention in Backup mode

28.3. Block Diagram

Figure 28-1. Power Manager Block Diagram



28.4. Peripheral Dependencies

Table 28-1. PM Configuration Summary

Peripheral Name	Base Address	NVIC IRQ Index: Source	PAC Peripheral ID (PAC.WRCTRL.PERIDx)	Power Domain
PM	0x4400_6000	3 : SLEEPRDY	3	VDDREG

28.5. Functional Description

28.5.1. Power Domains

Leaving aside the domains supplied by external power pins, such as VDDREG, AVDD, VDDIO and VUSB3V3, the device is split into these internal power domains:

- VDDCORE_BU_PD
- VDDCORE_SW_PD
- VDDCORE_RAM_PD
- VDDCORE_USB_PD

VDDCORE_SW_PD: is the Switchable Power Domain. It contains the CPU and all the peripherals except those located in the backup power domain.

VDDCORE_BU_PD: Contains the backup peripherals: OSC32KCTRL, SUPC, RSTC, RTC, TRAM, and the PM itself.

VDDCORE_RAM_PD: Contains the system RAM. It can be partially or fully turned off in Standby or Hibernate mode according to the user configuration.

VDDCORE_USB_PD : Dedicated power domain containing USB-PHY.

28.5.2. 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:

- **Active mode:** CPU is running, all power domains are active.
- **Idle Sleep mode:** The CPU is stopped. Synchronous clocks are stopped except when requested. The logic state is retained.
- **Standby Sleep mode:** The CPU is stopped as well as the peripherals. The logic state is retained, and power domain gating can be used to off the VDDCORE_RAM_PD power domain.
- **Hibernate Sleep mode:** VDDCORE_SW_PD power domain is turned OFF. The VDDCORE_BU_PD backup power domain is kept powered to allow few features to run (RTC, 32KHz clock sources, and wake-up from external pins). The VDDCORE_RAM_PD power domain can be retained according to software configuration.
- **Backup Sleep mode:** Only the 3.3 V backup domain (VDD_BU_PD)is kept powered to allow few features to run (RTC, 32KHz clock sources, and wake-up from external pins).
- **Off Sleep mode:** The entire device is powered off.

28.5.3. 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 Sleep Mode Controller.

The PM Power Domain Controller allows to reduce the power consumption in Standby mode even further.

28.5.4. Basic Operation

28.5.4.1. Initialization

After a Power-on Reset, the PM is enabled, the device is in Active mode, and all the power domains are in active state.

28.5.4.2. Enabling, Disabling, and Resetting

The PM is always enabled and cannot be reset.

28.5.4.3. Sleep Mode Controller

A Sleep mode is entered by executing the Wait For Interrupt instruction (WFI). The Sleep Mode bits in the Sleep Configuration register (SLEEPCFG.SLEEPMODE[2:0]) select the level of the sleep mode.

Note: Due to clock domain synchronization a small latency occurs between the store instruction and actual writing of the SLEEPCFG.SLEEPMODE[2:0]. Software must ensure that the SLEEPCFG register reads the desired value of SLEEPMODE before issuing the WFI instruction.

Entry and exit from the various sleep modes is shown in the following table:

Table 28-2. Sleep Mode Entry and Exit

Mode	Mode Entry	Wake-Up Sources
IDLE	SLEEPCFG.SLEEPMODE = IDLE + WFI	Synchronous [2] (APB, AHB), asynchronous [1]
STANDBY	SLEEPCFG.SLEEPMODE = STANDBY + WFI	Synchronous[3], asynchronous [1]
HIBERNATE	SLEEPCFG.SLEEPMODE = HIBERNATE + WFI	Hibernate reset detected by the RSTC
BACKUP	SLEEPCFG.SLEEPMODE = BACKUP + WFI	Backup reset detected by the RSTC
OFF	SLEEPCFG.SLEEPMODE = OFF + WFI	External Reset

The sleep modes (Idle, Standby, Hibernate, Backup, and Off) and their effect on clock activity, regulators, and NVM state are described in the sections below. Refer to the Power Domain Controller section for the power domain gating effect.

Table 28-3. Modes of Operation

Mode	Main Clock	CPU	AHBx & APBx Clocks	GCLK Clocks	Oscillators ONDEMAND=		Regulators				NVM
					0	1	VREGSW	VREGRAM	Addl. Regs	LPVREG	
ACTIVE	Run	Run	Run	RiR[2]	Run	RiR[2]	On	On	On/Off[5]	Off	Active
IDLE	Run	Stop	Stop[1]	RiR[2]	Run	RiR[2]	On	On	On/Off[5]	Off	Active
STANDBY	Stop	Stop	Stop[1]	Stop[1]	RiRR[3]	RiR[2]	On	LPM[4]	On/Off[6]	Off	LPM[4]
HIBERNATE	Stop	Stop	Stop	Stop	Stop	Stop	Off	LPM[4]	Off	Off	Off
BACKUP	Stop	Stop	Stop	Stop	Stop	Stop	Off	Off	Off	On	Off
OFF	Stop	Stop	Off	Off	Off	Off	Off	Off	Off	Off	Off

Notes:

1. Stop except if running during Sleepwalking mode.
2. RiR = Run if Requested.
3. RiRR = Run if Requested or RUNSTDBY = 1.
4. LPM = Low-Power Mode.
5. On or Off, controlled by VREGCTRL.AVREGEN.
6. On or Off, controlled by (VREGCTRL.AVREGSTDBY = 1) && VREGCTRL.AVREGEN. If On then it will be in Low-Power mode.

28.5.4.3.1. Idle Mode

The 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. Before entering the Idle mode, the user must select the Idle Sleep mode in the Sleep Configuration register (SLEEPCFG.SLEEPMODE = IDLE).

Additionally, if the SLEEPONEXIT bit in the Arm Cortex System Control register (SCR) is set, the Idle mode will be entered when the CPU exits the lowest priority ISR (Interrupt Service Routine. See Arm Cortex documentation for details). This mechanism can be useful for applications that only require the processor to run when an interrupt occurs.

Exiting Idle mode: The processor wakes the system up when it detects any non-masked interrupt with sufficient priority to cause exception entry according to the PRIMASK register. The system goes back to the Active mode. The CPU and affected modules are restarted.

GCLK clocks, regulators, and RAM are not affected by the Idle Sleep mode and operate in normal mode.

28.5.4.3.2. Standby Mode

The Standby mode is the lowest power configuration while keeping the state of the logic and the content of the RAM.

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.CTRLA.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).

Additionally, if the SLEEPONEXIT bit in the Arm Cortex System Control register (SCR) is set, the Standby mode will be entered when the CPU exits the lowest priority ISR (Interrupt Service Routine. This mechanism can be useful for applications that only require the processor to run when an interrupt occurs. Refer to the "Arm Cortex documentation" for details.

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 awakened, 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. In Sleepwalking, a peripheral involved in the Sleepwalking task can wake up the system whatever the interrupt type (asynchronous or synchronous).

28.5.4.3.3. Hibernate and Backup Modes

Hibernate and Backup modes allow the lowest power consumption aside from off. The device is entirely powered off except for the VDDCORE_BU_PD domain. In Hibernate mode, the VDDCORE_RAM_PD power domain can be retained according to software configuration. All peripherals in backup domain are allowed to run, for example, the RTC can be clocked by a 32.768 kHz oscillator. All PM registers are reset except the CTRLA.IORET bit.

Entering Hibernate or Backup mode: This mode is entered by executing the WFI instruction after selecting the Hibernate or Backup mode by writing the Sleep Mode bits in the Sleep Configuration register (SLEEPMODE = HIBERNATE or = BACKUP).

Exiting Hibernate or Backup mode: This mode is triggered when a Hibernate or Backup Reset is detected by the Reset Controller (RSTC).

Notes:

1. In Hibernate mode, the VREGRAM (in Low-Power mode) regulator is used to allow powering the VDDCORE_RAM_PD power domain which can be fully retained according to software configuration.
2. In Backup mode, the Backup Regulator (LPVREGC) is used.
3. Before entering Backup Sleep mode, it is recommended to poll the INTFLAG.SLEEPRDY bit to make sure that the backup regulator is ready. Let us say that the WFI instruction is executed whereas this flag is not set, then the system will go in a “pseudo backup mode where the VDDCORE_SW_PD power domain is turned off, but the VREGRAM is still used. Then the system will really go to Backup Sleep mode once the INTFLAG.SLEEPRDY field is set.
4. When WFI instruction is executed, before entering Hibernate or Backup, the PM waits until all the synchronous clocks have been properly turned off by MCLK and MCLK honors all clock requests from peripherals before turning off the synchronous clocks. Once there are no clock requests pending, the VDDCORE_SW_PD power domain can be turned off. Consequently, software is responsible to disable peripherals before executing the sleep instruction to avoid pending clock requests.
5. Let us say an interrupt occurs at the moment when the CPU is sleeping and MCLK is honoring a pending clock request then the device will wake up and the CPU can service the corresponding interrupt handler. The RCAUSE.BACKUP bit is zero as no backup reset has occurred.

28.5.4.3.4. 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 after it is verified that the SLEEPMODE field has been set to Off mode.

Exiting Off mode: This mode is left by pulling the RESET pin low.

28.5.4.4. I/O Line State Retention in Hibernate and Backup Modes

When entering Hibernate or Backup mode, PORTs are powered off but the pin configuration is retained. When the device exits the Hibernate or Backup mode, the I/O line configuration can either be released or stretched, based on the I/O Retention bit in the CTRLA register (CTRLA.IORET).

If IORET = 0 when exiting Hibernate or Backup mode, the I/O lines configuration is released and driven by the reset values of each PORT.

If the IORET = 1 when exiting Hibernate or Backup mode, the configuration of the I/O lines is retained until the IORET bit is written to 0. It allows the I/O lines to be retained until the application has reprogrammed each PORT.

28.5.4.5. Power Domain Controller

The Power Domain Controller provides several ways of how power domains are handled while the device is in Standby, Hibernate or Backup mode:

Standby mode:

When entering a Standby mode, the VDDCORE_RAM_PD power domain can be either fully or partially retained according to STDBYCFG.RAMCFG bits. When running a Sleepwalking task, VDDCORE_RAM_PD power domain is active whatever the STDBYCFG.RAMCFG is set to retain all RAM memory.

Dedicated power domain VDDCORE_USB_PD powered by the associated regulator can be ON or OFF according to the SUPC.VREGCTRL.AVREGSTDBY[0] and AVREGCFGEN[0] values.

Hibernate mode:

When entering a Hibernate mode, the VDDCORE_SW_PD power domain is off. As in Standby mode, the VDDCORE_RAM_PD power domain can be selectively turned ON or OFF by using HIBCFG.RAMCFG field.

Dedicated power domain (VDDCORE_USB_PD) is OFF.

Backup mode:

When entering Backup mode, the VDDCORE_SW_PD and VDDCORE_RAM_PD power domains are off, as well as the dedicated power domain (VDDCORE_USB_PD). VDDCORE_BU_PD is still active (powered by LPVREGC).

OFF mode:

When entering Off mode, all the power domains are off. I/Os are in high-impedance mode except the RESET_N pad which is still in input mode able to detect a reset to wake up the chip.

28.5.4.6. Regulators, RAMs, and NVM State in Sleep Modes

Idle sleep mode:

GCLK clocks, regulators, NVM, and RAM are not affected in Idle sleep mode and operate normally.

Standby sleep mode:

By default, in Standby sleep mode, the RAMs, NVM, and regulators are automatically set in Low-Power mode in order to reduce power consumption.

- The RAM can be disabled when STDBYCFG.RAMCFG = 0 when the device is in Standby mode. RAM contents are retained.
- If not powered-down, the RAM can be set in Low-Power mode according to STDBYCFG.LPRAM.
- If neither powered-down nor Low-Power mode is enabled, the RAM data is retained and not set in retention mode. In this case, the wakeup-time is better as there is no need to turn OFF/ON the RAM power-switches.

Hibernate sleep mode:

RAM can be disabled in Hibernate mode and data retained.

If not powered-down, the RAM can be set in low-power mode according to the HIBCFG.LPRAM bit. If in low- power mode, RAM data is retained.

If neither powered-down nor low-power mode, RAM data is retained and not set in retention mode. In this case, the wakeup-time is better as there is no need to turn OFF/ON the RAM power-switches.

In Hibernate mode NVM is powered down.

Backup sleep mode:

The RAMs located in VDDCORE_RAM_PD are powered down.

NVM is powered down.

VDDCORE_RAM_PD is OFF as well as VDDCORE_SW_PD.

The backup domain VDDCORE_BU_PD is powered by LPVREG.

28.5.5. Advanced Features

28.5.5.1. Sleepwalking

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 awakened by an interrupt (from a peripheral involved in SleepWalking) or enter again into Standby sleep mode.

In Standby when Sleepwalking is ongoing:

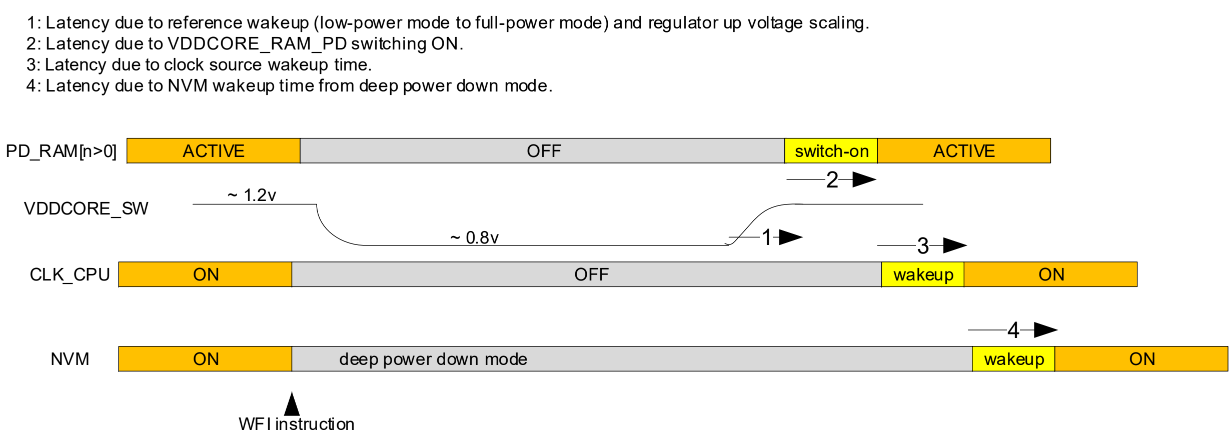
- All the power domains are turned ON including PDRAM power domain
- Low power mode of the regulators is not activated during sleepwalking

28.5.5.2. Wake-Up Time

As shown in the following figure, total wake-up time depends on:

1. Latency due to Reference and Regulator effect.
2. Latency due to Power Domain Gating:
   Usually, wake-up time is measured with the assumption that the power domains are already in active state. When using Power Domain Gating, changing a power domain from OFF to active state will take a certain time, refer to Electrical Characteristics. If all power domains were already in active state in standby sleep mode, this latency is zero.
3. Latency due to the CPU clock source wake-up time.
4. Latency due to the NVM wakeup time from deep power down mode.

Figure 28-2. Total Wake-Up Time from Standby Sleep Mode



28.5.6. Interrupts

Sleep Mode Entry Ready (SLEEPRDY): indicates that the device is ready to enter backup sleep mode, that is, it indicates that the low power bandgap of the LPVREGC regulator is ready, given the fact that it can take several 100's of microseconds at power-up for the low power bandgap to settle.

This interrupt is a synchronous wake-up source. For example, application can setup the device to enter backup sleep mode, go to standby sleep and get a wakeup interrupt from SLEEPRDY so it can then go to Backup Sleep.

This 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. This interrupt can be enabled by writing a '1' to the corresponding bit in the Interrupt Enable Set (INTENSET) register, and disabled by writing a '1' to the corresponding bit in the Interrupt Enable Clear (INTENCLR) register.

An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled in the NVIC module. The interrupt request remains active until the interrupt flag is cleared, the interrupt is disabled, or the peripheral is reset.

28.5.7. Sleep Mode Operation

The Power Manager is always active in sleep modes.

28.5.8. Debug Operation

When the CPU is halted in debug mode, the PM continues normal operation. If the 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.

When there is no debug session, in the Hibernate or Backup sleep mode the core power domains are OFF. If the Hibernate or Backup 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 Hibernate or Backup sleep mode upon a reset condition, the core domains are reset except the debug logic, allowing users to continue in the current debug session.

28.5.9. Register Access Protection

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

- Interrupt Flag Status and Clear register (INTFLAG)

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

28.6. Register Summary

For descriptions and definitions of both Register and bitfield properties, refer to Register Properties.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRLA	7:0						IORET		
0x01	SLEEPCFG	7:0						SLEEPMODE[2:0]		
0x02 ... 0x03	Reserved									
0x04	INTENCLR	7:0								SLEEPRDY
0x05	INTENSET	7:0								SLEEPRDY
0x06	INTFLAG	7:0								SLEEPRDY
0x07	Reserved									
0x08	STDBYCFG	7:0						LPRAM		RAMCFG
0x09	HIBCFG	7:0						LPRAM		RAMCFG

28.6.1. Control A

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

Table 28-4. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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

Bit 2 – IORET I/O Retention

When entering Hibernate or Backup mode, I/O ports are powered off but the pin configurations are retained. When the device exits either mode, pin configurations are either released or stretched, based on this bit:

Value	Description
`0x0`	When the device exits the Hibernate or Backup mode, pin configurations are released and driven to the current values defined pin's PORT registers.
`0x1`	When the device exits the Hibernate or Backup mode, pin configurations are retained until the IORET bit is cleared. This allows the pins to retain their states until the application has programmed new PORT values and then clears IORET.

28.6.2. Sleep Configuration

Name: SLEEPCFG
Offset: 0x0001
Reset: 0x02
Property: PAC Write-Protection

Table 28-5. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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

Bits 2:0 – SLEEPMODE[2:0] Sleep Mode

Value	Definition
0x0	Reserved
0x1	Reserved
0x2	CPU, AHB and APB clocks are OFF (IDLE)
0x3	Reserved
0x4	All Clocks are OFF (STANDBY)
0x5	Backup domain is ON as well as some PDRAMs (HIBERNATE)
0x6	Only Backup domain is powered ON (BACKUP)
0x7	All power domains are powered OFF (OFF)

28.6.3. Interrupt Enable Clear

Name: INTENCLR
Offset: 0x0004
Reset: 0x00
Property: PAC Write-Protection

Table 28-6. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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

Bit 0 – SLEEPRDY Backup Sleep Mode Entry Ready Disable
Writing a ‘0’ to this bit has no effect.
Writing a ‘1’ to this bit will disable the SLEEPRDY interrupt.
Note: Reading this bit returns whether this interrupt is enabled (1 = enabled).

28.6.4. Interrupt Enable Set

Name: INTENSET
Offset: 0x0005
Reset: 0x00
Property: PAC Write-Protection

Table 28-7. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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

Bit 0 – SLEEPRDY Backup Sleep Mode Entry Ready Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will enable the SLEEPRDY interrupt.
Note: Reading this bit returns whether this interrupt is enabled (1 = enabled).

28.6.5. Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x0006
Reset: 0x00
Property: –

Note: Subsequent to an interrupt flag being cleared, the flag must be read back to verify the clear before exiting the ISR. Failure to do this can result in duplicate interrupts.

Table 28-8. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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

Bit 0 – SLEEPRDY Sleep Mode Entry Ready

Note: Writing a ‘1’ to this bit will clear the flag and the interrupt.

Value	Description
0	Device is not ready to enter Backup Sleep Mode.
1	Device is ready to enter Backup Sleep Mode.

28.6.6. Standby Configuration

Name: STDBYCFG
Offset: 0x0008
Reset: 0x0004
Property: PAC Write-Protection

Table 28-9. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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

Bit 2 – LPRAM Low Power RAM Enable

Value	Description
0x0	RAM is not retained in standby when Low Power mode is active
0x1	RAM is retained in standby when Low Power mode is active

Bit 0 – RAMCFG Standby Mode RAM Configuration

Value	Description
0	All the RAMs are retained (RET)
1	Only the first 32kB are retained (OFF)

28.6.7. Hibernate Configuration

Name: HIBCFG
Offset: 0x0009
Reset: 0x0004
Property: PAC Write-Protection

Table 28-10. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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

Bit 2 – LPRAM Low Power RAM Enable

Value	Description
0x0	RAM is not retained in standby when Low Power mode is active
0x1	RAM is retained in standby when Low Power mode is active

Bit 0 – RAMCFG Hibernate Mode RAM Configuration

Value	Description
0x0	All the RAMs are retained when in Hibernate Mode.
0x1	Only the first 32kB are retained when in Hibernate Mode.

29. Reset Controller (RSTC)

29.1. Overview

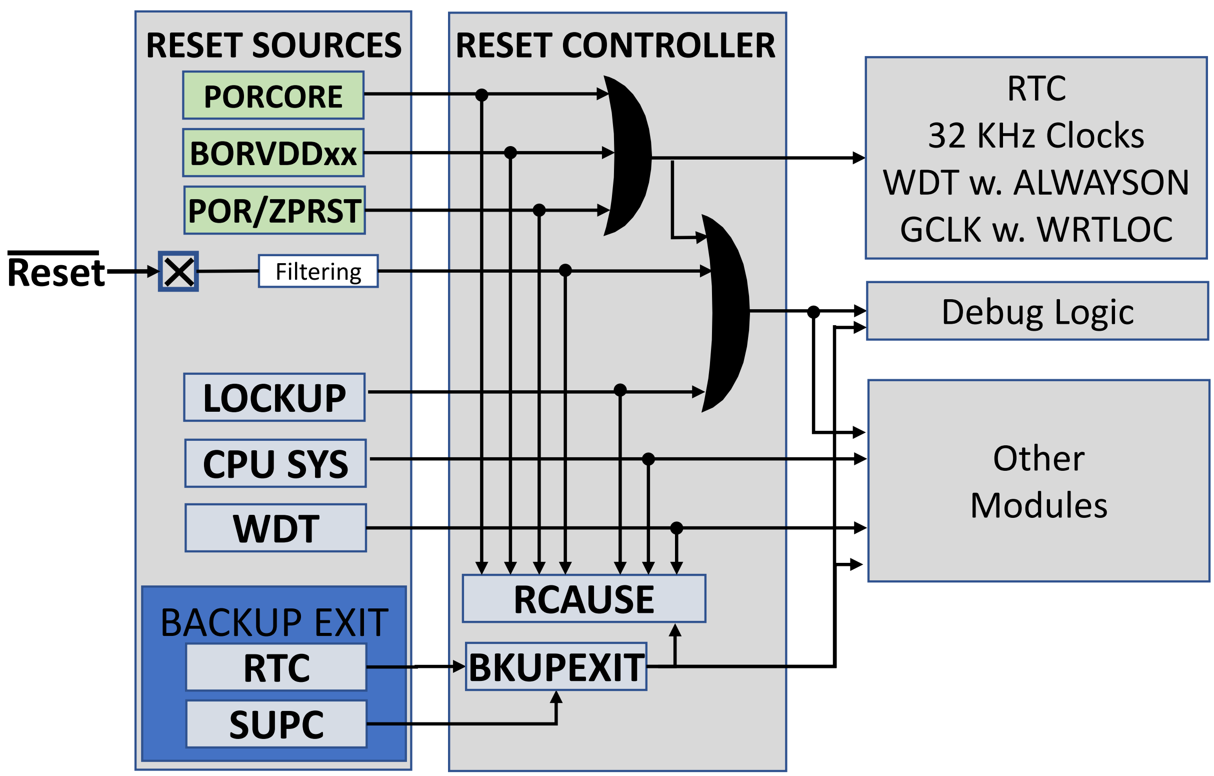
The Reset Controller (RSTC) manages the reset of the Microcontroller. It issues a Microcontroller reset, sets the device to its initial state and allows the reset source to be identified by software.

29.2. 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, PORCORE, BOR33
  - **User reset sources:** External reset (RESET), Watchdog reset, System Reset Request, CPU Lockup reset
  - **Backup exit sources:** Real-Time Counter (RTC)

29.3. Block Diagram

Figure 29-1. Reset Controller Block Diagram



29.4. Signals Description

Table 29-1. Signals Description

Signal Name	Type	Description
$\overline{\text{RESET}}$	Digital input	External reset

29.5. Peripheral Dependencies

Peripheral Name	Base Address	MCLK AHBx/APBx Clock Enable Mask Bit	PAC Peripheral ID (PAC.WRCTRL.PERIDx)	Power Domain
RSTC	0x4400_A000	MLCK.CLKMSK1[5]	5	VDDREG

29.6. Functional Description

The Reset Controller collects the various Reset sources and generates Resets for the device.

29.6.1. Basic Operation

29.6.1.1. Initialization

After a Power-on Reset, the RSTC is enabled and the Reset Cause (RCAUSE) register indicates the Reset source.

29.6.1.2. Enabling, Disabling, and Resetting

The RSTC module is always enabled.

29.6.1.3. Reset Causes and Effects

The latest Reset cause is available in the RCAUSE register and can be read during the application boot sequence to determine proper action.

These are the groups of Reset sources:

- **Power supply Reset:** Resets caused by an electrical issue. It covers POR and PORCORE resets.
- **Brown-out Reset:** Resets caused by BORVDDx detectors.
- **User Reset:** Resets caused by the application. It covers external Resets, system Reset requests, CPU Lockup Reset and Watchdog Timer Reset.
- **Backup Reset:** Reset cause when leaving Backup or Hibernate mode due to wake event RTC, or $\overline{\text{RESET}}$.

The following table lists the parts of the device that are reset, depending on the Reset type.

Table 29-2. Effects of Reset Causes

Effect: Reset:	CPU Logic, Bus Logic	RTC[1], OSC32KCTRL, BKOUT, BOR[2], SUPC, VREGCTRL & VREFCTRL[3] registers, CTRLA bits of PM	Flash Panel Controller	Debug Logic	Other Modules
POR, PORCORE	Reset	Reset	Reset	Reset	Reset
External Reset BORVDDx	Reset	No Reset	No Reset	Reset	Reset
WDT Reset Lockup Reset System Reset Request	Reset	No Reset	No Reset	No Reset	Reset
RTC	-	No Reset	Reset	Reset	-

Notes:
1. RTC.DBGCTRL is reset in Backup Sleep mode.
2. The BOR register is retained during Backup mode and is not reset when exiting.

The external Reset is generated when pulling the RESET pin low.

The POR, PORCORE, and BORVDDx (for example, BORVDDA, BORVDDREG) Reset sources are generated by the analog modules inside the PWR_SMOR or PWR_SMOR_DS controlled by the Supply Controller Interface (SUPC).

The WDT Reset is generated by the Watchdog Timer (WDT).

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 additional information, refer to the “Arm® Cortex® Technical Reference Manual” which is available for download at www.arm.com).

The Lockup Reset is a Reset generated by the CPU when it enters a lockup state. For additional information, refer to the “Arm® Cortex® Technical Reference Manual” which is available for download at www.arm.com).

29.6.2. Power Management

The Reset Controller module is always on.

29.6.3. Sleep Mode Operation

The RSTC module is active in all sleep modes.

29.6.4. Debug Operation

When the CPU is halted in Debug mode, the RSTC continues normal operation.

If the DBGCTRL.LCKUPDIS bit is set, the CPU LOCKUP reset source is disabled in Debug mode. Refer to the DBGCTRL register for additional information.

29.6.5. PAC Write-Protection

All registers with write-access can be optionally write protected by the Peripheral Access Controller (PAC).

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

When the CPU is halted in Debug mode, all write-protection is automatically disabled. Write-protection does not apply for accesses through an external debugger.

29.7. Register Summary

For descriptions and definitions of both Register and bitfield properties, refer to Register Properties.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	RCAUSE	15:8							LOCKUP	BACKUP
		7:0	SYST	WDT	EXT	BORVDDIO	BORVDDA	BORVDDREG	PORCORE	POR
0x02	BKUPEXIT	7:0	HIB						RTC	
0x03	Reserved									
0x04	DBGCTRL	7:0								LCKUPDIS

29.7.1. Reset Cause

Name: RCAUSE
Offset: 0x0000
Reset: 0x0000
Property: –

Note: Only one bit is set at any one time. This register only reflects the last event.

Table 29-3. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	15	14	13	12	11	10	9	8
							LOCKUP	BACKUP
Access							R	R
Reset							0	0

Bit	7	6	5	4	3	2	1	0
	SYST	WDT	EXT	BORVDDIO	BORVDDA	BORVDDREG	PORCORE	POR
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 9 – LOCKUP Lockup Reset
CPU has entered lockup state (for details, refer to the “Arm® Cortex™ Technical Reference Manual” available at www.arm.com).

Bit 8 – BACKUP Backup Reset
Reset cause when leaving Backup or Hibernate mode due to wake event (RTC).

Bit 7 – SYST System Reset Request
Reset generated by a System Reset Request by the CPU by asserting the SYSRESETREQ bit located in the Reset Control register of the CPU (for details, refer to the “Arm® Cortex™ Technical Reference Manual” available at www.arm.com).

Bit 6 – WDT Watchdog Reset
Reset generated by Watchdog Timer.

Bit 5 – EXT External Reset
Reset by the $\overline{\text{RESET}}$ pin driven low.

Bit 4 – BORVDDIO Brown-out VDDIO Detector Reset
Reset generated by VDDIO BOR Detector.

Bit 3 – BORVDDA Brown-out VDDA Detector Reset
Reset generated by VDDA BOR Detector.
Note: VDDA and AVDD are used interchangeably.

Bit 2 – BORVDDREG Brown-out VDDREG Detector Reset
Reset by VDDREG BOR Detector.

Bit 1 – PORCORE Core Power-on Reset
Reset by Core Power-on Reset.

Bit 0 – POR Power-on Reset
Reset generated by Power-on Reset.

29.7.2. Backup Exit Source

Name: BKUPEXIT
Offset: 0x0002
Reset: 0x00
Property: –

Note: Backup exit only: RTC=1
Hibernate exit: HIB and RTC = 1

Table 29-4. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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

Bit 7 – HIB Hibernate
Wake up from Hibernate Sleep Mode.

Bit 1 – RTC Real Timer Counter Interrupt
Wake up from Real Time Counter (RTC) interrupt.

29.7.3. Debug Control

Name: DBGCTRL
Offset: 0x0004
Reset: 0x00
Property: PAC Write-Protection

Table 29-5. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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

Bit 0 – LCKUPDIS Lockup Disable
If set the CPU LOCKUP Reset source is disabled in debug mode.
(The Lockup Reset is a Reset generated by the CPU when it enters a lockup state (for details refer to the ARM® Cortex™ Technical Reference Manual on www.arm.com)).

30. External Interrupt Controller (EIC)

30.1. 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, on falling, on 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.

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.

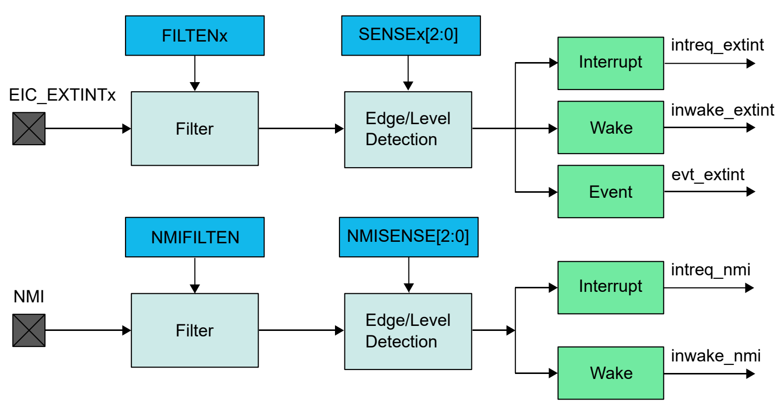
30.2. Features

The following are key features of the module:

- Up to 16 external interrupt pins (EIC_EXTINTx, x = 0-15)
- 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 interrupt, EIC_EXTINTx, pins
- Event generation trigger events from EIC_EXTINTx pins

30.3. Block Diagram

Figure 30-1. EIC Block Diagram



30.4. Signal Description

Signal Name	Type	Description
EIC_EXTINTx[15..0]	Digital Input	External interrupt pin

Note: One signal may be available on several pins.

30.5. Peripheral Dependencies

Peripheral Name	Base Address	NVIC IRQ Index: Source	MCLK AHBx/APBx Clock Enable Mask Bit	GCLK Peripheral Channel Clock Name:Register	PAC Peripheral Identifier (PAC.WRCTRL.PERIDx)	EVSYS Generator (EVSYS.CHANNELn.EVGENx)	Power Domain
EIC	0x4401_A000	N/A : NMI 13-28 : EXTINTx 29 : NSCHK	MCLK.CLKMSK1[13]	GCLK_EIC: GCLK.PCHCTRL[5]	13	21-36 : EXTINTx	VDDREG

30.5.1. I/O Lines

Using the EIC's I/O lines requires the I/O pins to be configured.

30.5.2. Power Management

All interrupts are available down to Standby Sleep mode, but the EIC can be configured to automatically mask some interrupts in order to prevent device wake-up.

The EIC will continue to operate in any sleep mode where the selected source clock is running. The EIC's interrupts can be used to wake up the device from sleep modes. Events connected to the Event System can trigger other operations in the system without exiting sleep modes.

30.5.3. Clocks

The EIC bus clock (CLK EIC APB) can be enabled and disabled by the Main Clock Controller, the default state of CLK EIC APB can be found in the Peripheral Clock Masking section.

Some optional functions need a peripheral clock, which can either be a generic clock (GCLK EIC, for wider frequency selection) or an Ultra Low-Power 32 kHz clock (CLK ULP32K, for highest power efficiency). One of the clock sources must be configured and enabled before using the peripheral:

GCLK EIC is configured and enabled in the Generic Clock Controller.

CLK ULP32K is provided by the internal Ultra Low-Power (OSCULP32K) Oscillator in the OSC32KCTRL module.

Both GCLK EIC and CLK ULP32K are asynchronous to the user interface clock (CLK EIC APB). Due to the clock being asynchronous, writes to certain registers will require synchronization between the clock domains. Refer to Synchronization for further details.

30.5.4. Interrupts

There are several interrupt request lines, at least one for the external interrupts (EXTINT) and one for non-maskable interrupt (NMI).

The EXTINT interrupt request line is connected to the interrupt controller. Using the EIC interrupt requires the interrupt controller to be configured first.

The NMI interrupt request line is also connected to the interrupt controller, but does not require the interrupt to be configured.

30.5.5. Events

The events are connected to the Event System. Using the events requires the Event System to be configured first.

30.5.6. 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.

30.5.7. Register Access Protection

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

- Interrupt Flag Status and Clear register(INTFLAG)
- Non-Maskable Interrupt Flag Status and Clear register(NMIFLAG)

Optional write-protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write-Protection" property in each individual register description.

PAC write-protection does not apply to accesses through an external debugger.

30.6. Functional Description

30.6.1. 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 (EIC_EXTINT) can be filtered using majority vote filtering, clocked by GCLK EIC or by CLK ULP32K.

30.6.2. Basic Operation

30.6.2.1. Initialization

The EIC must be initialized in the following order:

1. Enable CLK EICAPB.
2. If required, configure the NMI by writing the Non-Maskable Interrupt Control (NMICTRL) register.
3. Enable GCLK EIC or CLK ULP32K when one of the following external interrupt, EIC_EXTINTx pins, modes is selected:
   – EIC_EXTINT pin filtering
   – EIC_EXTINT pin synchronous edge detection
   – EIC_EXTINT pin de-bouncing
   – GCLK EIC is used when a frequency higher than 32 kHz is required for filtering
   – CLK ULP32K is recommended when power consumption is the priority
4. For CLK ULP32K write a '1' to the Clock Selection bit in the Control A register (CTRLA.CKSEL).
5. Configure the EIC input sense and filtering by writing the Configuration n register (CONFIG).
6. Optionally, enable the Asynchronous mode.
7. Optionally, enable the Debounce mode.
8. Enable the EIC by writing a '1' to CTRLA.ENABLE.

30.6.2.2. Enabling, Disabling, and Resetting

The EIC is enabled by writing a '1' the Enable bit in the Control A register (CTRLA.ENABLE). The EIC is disabled by writing CTRLA.ENABLE to '0'.

The EIC is reset by setting the Software Reset bit in the Control register (CTRLA.SWRST). All registers in the EIC will be reset to their initial state, and the EIC will be disabled.

Refer to the CTRLA register description for details.

30.6.3. 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 External Interrupt Sense Configuration n register (CONFIGn.SENSEx). The corresponding interrupt flag (INTFLAG.EXTINT[x]) in the Interrupt Flag Status and Clear register (INTFLAG) is set when the interrupt condition is met.

When the interrupt flag has been cleared in edge-sensitive mode, INTFLAG.EXTINT[x] will only be set if a new interrupt condition is met.

In level-sensitive mode, when interrupt has been cleared, INTFLAG.EXTINT[x] will be set immediately if the EIC_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 (CONFIG.FILTENx) is written to '1'. The majority vote filter samples the external pin three times with GCLK EIC or CLK ULP32K and outputs the value when two or more samples are equal.

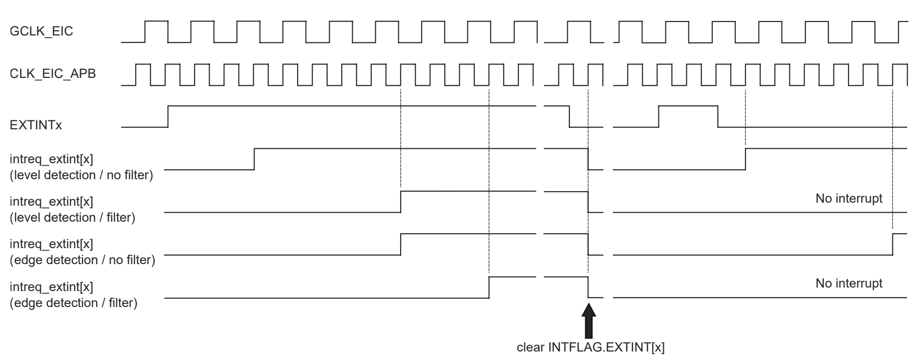
Table 30-1. Majority Vote Filter

Samples [0, 1, 2]	Filter Output
[0,0,0]	0
[0,0,1]	0
[0,1,0]	0
[0,1,1]	1
[1,0,0]	0
[1,0,1]	1
[1,1,0]	1
[1,1,1]	1

When an external interrupt is configured for level detection and when filtering is disabled, detection is done asynchronously. Level detection and asynchronous edge detection does not require GCLK EIC or 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 (CTRLA.CKSEL). GCLK EIC must be enabled in the GCLK module. In these modes the external pin is sampled at the EIC clock rate, therefore pulses with duration lower than two EIC clock periods may not be properly detected.

Figure 30-2. Interrupt Detection Latency by Modes (Rising Edge)



The detection latency depends on the detection mode.

Table 30-2. 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

30.6.4. Additional Features

30.6.4.1. 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.

30.6.4.2. Asynchronous Edge Detection Mode (No Debouncing)

The EIC_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 (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 (EIC_EXTINTx) pins are sampled using the EIC clock as defined by the Clock Selection bit in the Control A register (CTRLA.CKSEL). The External Interrupt flag (INTFLAG.EXTINT[x]) 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 (EIC_EXTINTx) pins set the External Interrupt flag (INTFLAG.EXTINT[x]) directly. In this mode, the EIC clock is not requested.

The asynchronous edge detection mode can be used in Idle and Standby sleep modes.

30.6.4.3. Interrupt Pin Debouncing

The external interrupt pin (EIC_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 the EIC_EXTINTx pin can be selected only if the Sense bits in the Configuration y register (CONFIGy.SENSEx) are set to RISE, FALL or BOTH. If the debouncing mode for pin EIC_EXTINTx is selected, the filter mode for that pin (CONFIGy.FILTENx) cannot be selected.

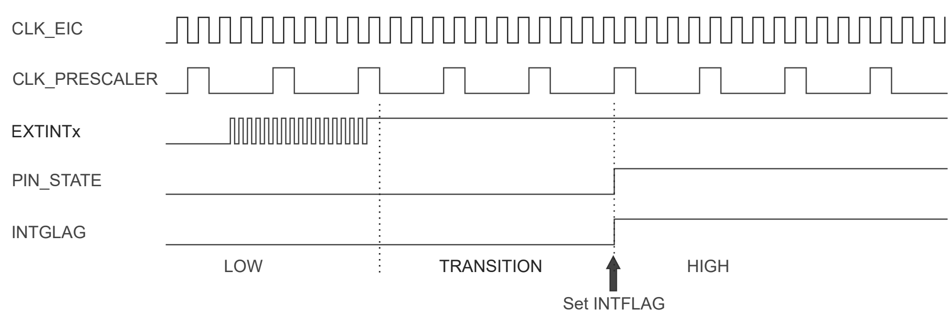
The debouncer manages an internal "valid pin state" that depends on the external interrupt (EIC_EXTINTx) pin transitions, the debouncing mode and the debouncer prescaler frequency. The valid pin state reflects the pin value after debouncing. The external interrupt pin(s) (EIC_EXTINTx) is sampled continuously on EIC clock. The sampled value is evaluated on each *low frequency clock* tick to detect a transitional edge when the sampled value is different of the current valid pin state. The sampled value is evaluated on each EIC clock when 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 EIC_EXTINTx pins, 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 PINSTATE register for both synchronous or asynchronous debouncing mode.

Synchronous edge detection – In this mode the external interrupt (EIC_EXTINTx) pin is sampled continuously on EIC clock.

1. A pin edge transition will be validated when the sampled value is consistently different of the current valid pin state for 4 (or 8 depending on bit 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 30-3. EIC_EXTINTx Pin Synchronous Debouncing (Rising Edge)

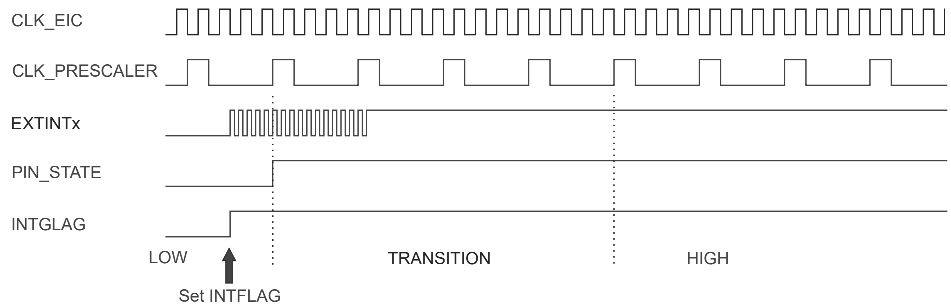


In the synchronous edge detection mode, the EIC clock is required. The synchronous edge detection mode can be used in Idle and Standby Sleep modes.

Asynchronous edge detection – In this mode, the external interrupt (EIC_EXTINTx) pins 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 30-4. EIC_EXTINTx Pin(s) 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.

30.6.5. Interrupts

The EIC has the following interrupt sources: External interrupt pins (EIC_EXTINTx; x = 0-15).

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. 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 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. Refer to the INTFLAG register for additional information on how to clear interrupt flags. The EIC has at least 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.

30.6.6. Events

The EIC can generate the following output events:

- External event from pin EIC_EXTINTx; x= 0-15

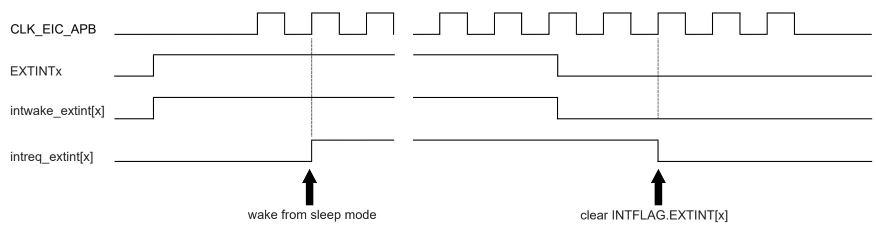
Setting an Event Output Control register (EVCTRL.EXTINTEO) enables the corresponding output event. Clearing this bit disables the corresponding output event. Refer to *Event System* for details on configuring the Event System.

When the condition on pin EIC_EXTINTx matches the configuration in the CONFIGn register, the corresponding event is generated, if enabled.

30.6.7. Sleep Mode Operation

In sleep modes, an EIC_EXTINTx pin can wake up the device if the corresponding condition matches the configuration in the CONFIG register, and the corresponding bit in the Interrupt Enable Set register (INTENSET) is written to '1'.

Figure 30-5. Wake-up Operation Example (High-Level Detection, No Filter, Interrupt Enable Set)



30.6.8. 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.

30.6.9. Synchronization

Due to the main clock domain and the peripheral clock domains being asynchronous, some registers need to be synchronized when written or read.

The following bits are synchronized when written:

- Software Reset bit in control register(CTRLA.SWRST)
- Enable bit in control register(CTRLA.ENABLE)

Required write-synchronization is denoted by the "Write-Synchronized" property in the register description.

30.7. Register Summary

For descriptions and definitions of both Register and bitfield properties, refer to Register Properties.

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	EXTINTEO[15:8]							
		7:0	EXTINTEO[7:0]							
0x0C	INTENCLR	31:24	NSCHK							
		23:16								
		15:8	EXTINT[15:8]							
		7:0	EXTINT[7:0]							
0x10	INTENSET	31:24	NSCHK							
		23:16								
		15:8	EXTINT[15:8]							
		7:0	EXTINT[7:0]							
0x14	INTFLAG	31:24	NSCHK							
		23:16								
		15:8	EXTINT[15:8]							
		7:0	EXTINT[7:0]							
0x18	ASYNCH	31:24								
		23:16								
		15:8	ASYNCH[15:8]							
		7:0	ASYNCH[7:0]							
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[15:8]							
		7:0	DEBOUNCEN[7:0]							
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	PINSTATE[15:8]							
		7:0	PINSTATE[7:0]							
0x3C	NSCHK	31:24	NMI							
		23:16								
		15:8	EXTINT[15:8]							
		7:0	EXTINT[7:0]							

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x40	NONSEC	31:24	NMI							
		23:16								
		15:8	EXTINT[15:8]							
		7:0	EXTINT[7:0]							

30.7.1. Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00
Property: PAC Write-Protection, Enable-Protected, Write-Synchronized

Notes:
1. Access to this register is limited to 32-bit width. Byte level access is not allowed.
2. Reserved bits must always be written as ‘0’.

Table 30-3. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
				CKSEL			ENABLE	SWRST
Access				R/W			R/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 kHz is required for filtering) or by CLK_ULP32K (when power consumption is the priority).
This bit is not Write-Synchronized.

Value	Description
0	The EIC is clocked by GCLK_EIC.
1	The EIC is clocked by 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.
This bit is not Enable-Protected.
This bit is Write-Synchronized.
Note: This bit is write protected and can only be written when CTRLA.ENABLE = 0.

Important: Enable-protected bits in the CTRLA register can be written at the same time when setting CTRLA.ENABLE to '1', but not at the same time when CTRLA.ENABLE is being cleared.

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.
Due to synchronization there is a delay from writing CTRLA.SWRST until the Reset is complete. CTRLA.SWRST and SYNCBUSY.SWRST will both be cleared when the Reset is complete.
This bit is not Enable-Protected.
This bit is Write-Synchronized.

Value	Description
0	There is no ongoing reset operation.
1	The reset operation is ongoing.

30.7.2. Non-Maskable Interrupt Control

Name: NMICTRL
Offset: 0x01
Reset: 0x00
Property: PAC Write-Protection

Table 30-4. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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

Bit 4 – NMIASYNCH Non-Maskable Interrupt Asynchronous Edge Detection Mode
The NMI edge detection can be operated synchronously or asynchronously to the EIC clock.

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

30.7.3. Non-Maskable Interrupt Flag Status and Clear

Name: NMIFLAG
Offset: 0x2
Reset: 0x00
Property: -

Note: Access to this register is limited to 32-bit width. Byte level access is not allowed.

Table 30-5. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
								NMI
Access								R/W
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.

30.7.4. Synchronization Busy

Name: SYNCBUSY
Offset: 0x04
Reset: 0x00000000
Property: -

Note: Access to this register is limited to 32-bit width. Byte level access is not allowed.

Table 30-6. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

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

Bit 1 – ENABLE Enable Synchronization Busy Status

Value	Description
0	Write synchronization for CTRLA.ENABLE bit is complete.
1	Write synchronization for CTRLA.ENABLE bit is ongoing.

Bit 0 – SWRST Software Reset Synchronization Busy Status

Value	Description
0	Write synchronization for CTRLA.SWRST bit is complete.
1	Write synchronization for CTRLA.SWRST bit is ongoing.

30.7.5. Event Control

Name: EVCTRL
Offset: 0x08
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Note: This register is write protected and can only be written when CTRLA.ENABLE = 0.

Table 30-7. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Important: On devices with security attribution 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 Enabled-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
	EXTINTEO[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
	EXTINTEO[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 – EXTINTEO[15:0] External Interrupt Event Output Enable
The bit x of EXTINTEO enables the event associated with the EIC_EXTINTx pin.

Value	Description
0	Event from pin EIC_EXTINTx is disabled.
1	Event from pin EIC_EXTINTx is enabled and will be generated when EIC_EXTINTx pin matches the external interrupt sensing configuration.

30.7.6. Interrupt Enable Clear

Name: INTENCLR
Offset: 0x0C
Reset: 0x00000000
Property: PAC Write-Protection

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set register (INTENSET).
Notes:
1. Access to this register is limited to 32-bit width. Byte level access is not allowed.
2. Reserved bits must always be written as ‘0’.
3. Interrupt flags must be cleared and then read back to confirm the clear before exiting the ISR to avoid double interrupts.

Table 30-8. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Important: On devices with security attribution 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	R/W							
Reset	0							

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	EXTINT[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
	EXTINT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 31 – NSCHK Non-secure Check Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the NSCHK Interrupt Enable bit. For devices without TrustZone support, this location is reserved.

Value	Description
0	The external interrupt EIC_EXTINTx is disabled.
1	The external interrupt EIC_EXTINTx is enabled.

Bits 15:0 – EXTINT[15:0] External Interrupt Enable

The bit x of EXTINT enables the interrupt associated with the EIC_EXTINTx pin.
Writing a '0' to bit x has no effect.
Writing a '1' to bit x will clear the External Interrupt Enable bit x, which disables the external interrupt EIC_EXTINTx pin(s).

Value	Description
0	The external interrupt EIC_EXTINTx is disabled.
1	The external interrupt EIC_EXTINTx is enabled.

30.7.7. Interrupt Enable Set

Name: INTENSET
Offset: 0x10
Reset: 0x00000000
Property: PAC Write-Protection

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear (INTENCLR) register.
Notes:
1. Access to this register is limited to 32-bit width. Byte level access is not allowed.
2. Reserved bits must always be written as ‘0’.

Table 30-9. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	NSCHK							
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
	EXTINT[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
	EXTINT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 31 – NSCHK Non-secure Check Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the NSCHK Interrupt Enable bit. For devices without TrustZone support, this location is reserved.

Value	Description
0	The external interrupt EIC_EXTINTx is disabled.
1	The external interrupt EIC_EXTINTx is enabled.

Bits 15:0 – EXTINT[15:0] External Interrupt Enable
The bit x of EXTINT enables the interrupt associated with the EIC_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 EIC_EXTINTx pin(s).

Value	Description
0	The external interrupt EIC_EXTINTx is disabled.
1	The external interrupt EIC_EXTINTx is enabled.

30.7.8. Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x14
Reset: 0x00000000
Property: -

Notes:

1. Interrupt flags must be cleared and then read back to confirm the clear before exiting the ISR to avoid double interrupts.
2. Access to this register is limited to 32-bit width. Byte level access is not allowed.
3. Reserved bits must always be written as ‘0’.

Table 30-10. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	NSCHK							
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
	EXTINT[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
	EXTINT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 31 – NSCHK Non-secure Check Interrupt Enable

For devices with TrustZone support, the 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. For devices without TrustZone support, this location is reserved.

Value	Description
0	The external interrupt EIC_EXTINTx is disabled.
1	The external interrupt EIC_EXTINTx is enabled.

Bits 15:0 – EXTINT[15:0] External Interrupt

The flag bit x is cleared by writing a '1' to it.
This flag is set when the EIC_EXTINTx pin matches the external interrupt sense configuration and will generate an interrupt request if INTENCLR.EXTINT[x] or 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.

30.7.9. External Interrupt Asynchronous Mode

Name: ASYNCH
Offset: 0x18
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Notes:
1. This register is write protected and can only be written when CTRLA.ENABLE = 0.
2. Access to this register is limited to 32-bit width. Byte level access is not allowed.
3. Reserved bits must always be written as '0'.

Table 30-11. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
	ASYNCH[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
	ASYNCH[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 – ASYNCH[15:0] Asynchronous Edge Detection Mode
The bit x of ASYNCH set the Asynchronous Edge Detection mode for the interrupt associated with the EIC_EXTINTx pins.

Value	Description
0	The EIC_EXTINTx edge detection is synchronously operated.
1	The EIC_EXTINTx edge detection is asynchronously operated.

30.7.10. External Interrupt Sense Configuration

Name: CONFIG0
Offset: 0x1C
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Notes:
1. This register is write protected and can only be written when CTRLA.ENABLE = 0.
2. Access to this register is limited to 32-bit width. Byte level access is not allowed.
3. Reserved bits must always be written as '0'.

Table 30-12. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	FILTEN7	SENSE7[2:0]			FILTEN6	SENSE6[2:0]		
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	FILTEN5	SENSE5[2:0]			FILTEN4	SENSE4[2:0]		
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	FILTEN3	SENSE3[2:0]			FILTEN2	SENSE2[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
	FILTEN1	SENSE1[2:0]			FILTEN0	SENSE0[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 3, 7, 11, 15, 19, 23, 27, 31 – FILTENx Filter Enable x [x=7..0]

Value	Description
0	Filter is disabled for the EIC_EXTINTx pin input.
1	Majority Vote (best 2 out 3). Filter is enabled for the EIC_EXTINTx pin input.

Note:
1. If corresponding EXTINTx EIC.CONFIGn.SENSEx is set, then the EIC.CONFIGn.FILTERx bit must not be set.

Bits 0:2, 4:6, 8:10, 12:14, 16:18, 20:22, 24:26, 28:30 – SENSEx Input Sense Configuration x [x=7..0]
These bits define on which edge or level the interrupt or event for the EIC_EXTINTx pins 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

Note:
1. If corresponding EXTINTx EIC.CONFIGn.SENSEx is set, then the EIC.CONFIGn.FILTERx bit must not be set.

30.7.11. External Interrupt Sense Configuration

Name: CONFIG1
Offset: 0x20
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Notes:
1. This register is write protected and can only be written when CTRLA.ENABLE = 0.
2. Access to this register is limited to 32-bit width. Byte level access is not allowed.
3. Reserved bits must always be written as '0'.

Table 30-13. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	FILTEN15	SENSE15[2:0]			FILTEN14	SENSE14[2:0]		
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	FILTEN13	SENSE13[2:0]			FILTEN12	SENSE12[2:0]		
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	FILTEN11	SENSE11[2:0]			FILTEN10	SENSE10[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
	FILTEN9	SENSE9[2:0]			FILTEN8	SENSE8[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 3, 7, 11, 15, 19, 23, 27, 31 – FILTENx Filter Enable x [x=7..0]

Value	Description
0	Filter is disabled for EIC_EXTINTx pin input.
1	Majority Vote, (best 2 out 3). Filter is enabled for EIC_EXTINTx pin input.

Note:
1. If corresponding EXTINTx EIC.CONFIGn.SENSEx is set, then EIC.CONFIGn.FILTERx bit must not be set.

Bits 0:2, 4:6, 8:10, 12:14, 16:18, 20:22, 24:26, 28:30 – SENSEx Input Sense Configuration x [x=7..0]
These bits define on which edge or level the interrupt or event for EIC_EXTINTx pin(s) 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

Note:
1. If corresponding EXTINTx EIC.CONFIGn.SENSEx is set, then EIC.CONFIGn.FILTERx bit must not be set.

30.7.12. Debouncer Enable

Name: DEBOUNCEN
Offset: 0x30
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Notes:
1. This register is write protected and can only be written when CTRLA.ENABLE = 0.
2. Access to this register is limited to 32-bit width. Byte level access is not allowed.
3. Reserved bits must always be written as '0'.

Table 30-14. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
	DEBOUNCEN[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
	DEBOUNCEN[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 – DEBOUNCEN[15:0] Debouncer Enable
The bit x of DEBOUNCEN set the Debounce mode for the interrupt associated with the EIC_EXTINTx pin(s).

Value	Description
0	The EIC_EXTINTx edge input is not debounced.
1	The EIC_EXTINTx edge input is debounced.

30.7.13. Debouncer Prescaler

Name: DPRESCALER
Offset: 0x34
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Notes:
1. This register is write protected and can only be written when CTRLA.ENABLE = 0.
2. Access to this register is limited to 32-bit width. Byte level access is not allowed.
3. Reserved bits must always be written as ‘0’.

Table 30-15. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
								TICKON
Access								R/W
Reset								0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	STATES1	PRESCALER1[2:0]			STATES0	PRESCALER0[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 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 – STATESn Debouncer Number of States [n=0,1]
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+(8n):8n].

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 – PRESCALERn Debouncer Prescaler [n=0,1]

These bits select the debouncer low frequency clock for pins EXTINT[7+(8n):8n].

Value	Name	Description
0x0	F/2	EIC clock divided by 2
0x1	F/4	EIC clock divided by 4
0x2	F/8	EIC clock divided by 8
0x3	F/16	EIC clock divided by 16
0x4	F/32	EIC clock divided by 32
0x5	F/64	EIC clock divided by 64
0x6	F/128	EIC clock divided by 128
0x7	F/256	EIC clock divided by 256

30.7.14. Pin State

Name: PINSTATE
Offset: 0x38
Reset: 0x00000000
Property: PAC

Note: Access to this register is limited to 32-bit width. Byte level access is not allowed.

Table 30-16. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
	PINSTATE[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
	PINSTATE[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – PINSTATE[15:0] Pin State of external interrupts EIC_EXTINTx
These bits return the valid pin state of the debounced external interrupt pin, EIC_EXTINTx.

30.7.15. 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.

Table 30-17. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	NMI							
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
	EXTINT[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
	EXTINT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 31 – 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 15:0 – EXTINT[15:0] 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.

30.7.16. 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.

Table 30-18. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	NMI							
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
	EXTINT[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
	EXTINT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 31 – 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 15:0 – EXTINT[15:0] 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, INTENCLR, INTENSET, INTFLAG and CONFIG registers.

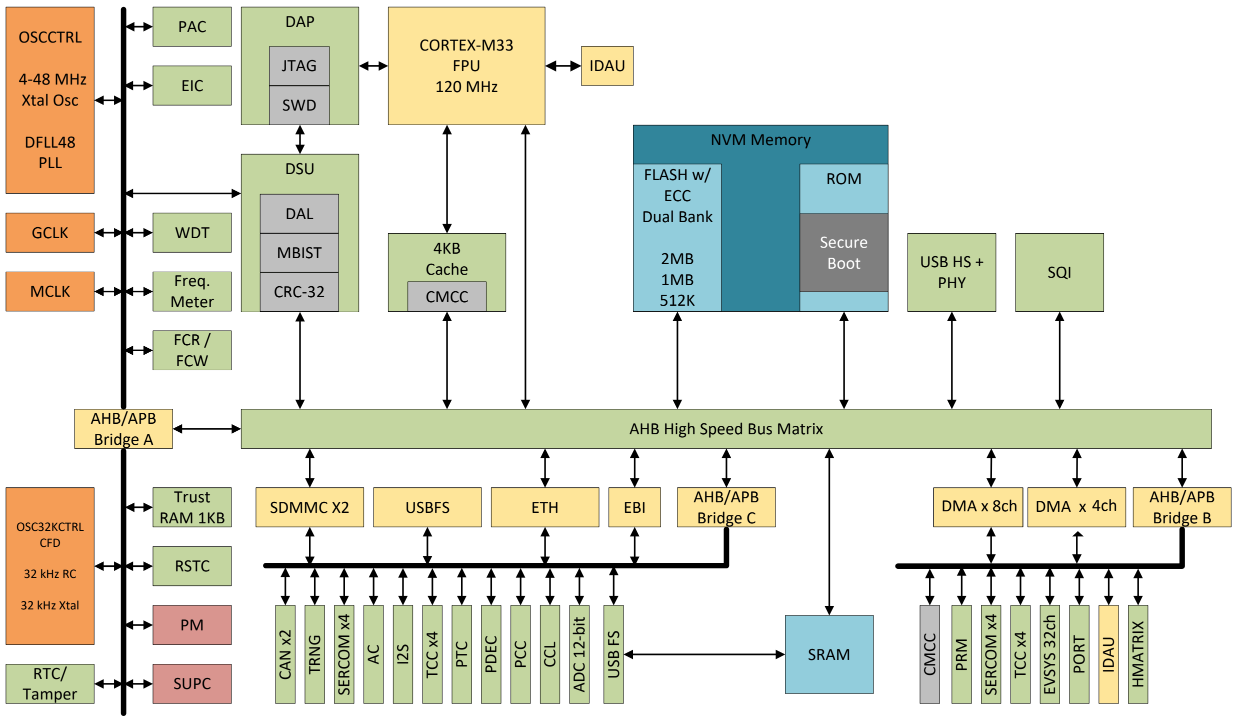
Value	Description
0	EXTINTx is secure.
1	EXTINTx is non-secure.

31. Non-Volatile Memory Controller (NVMCTRL)

31.1. Block Diagram

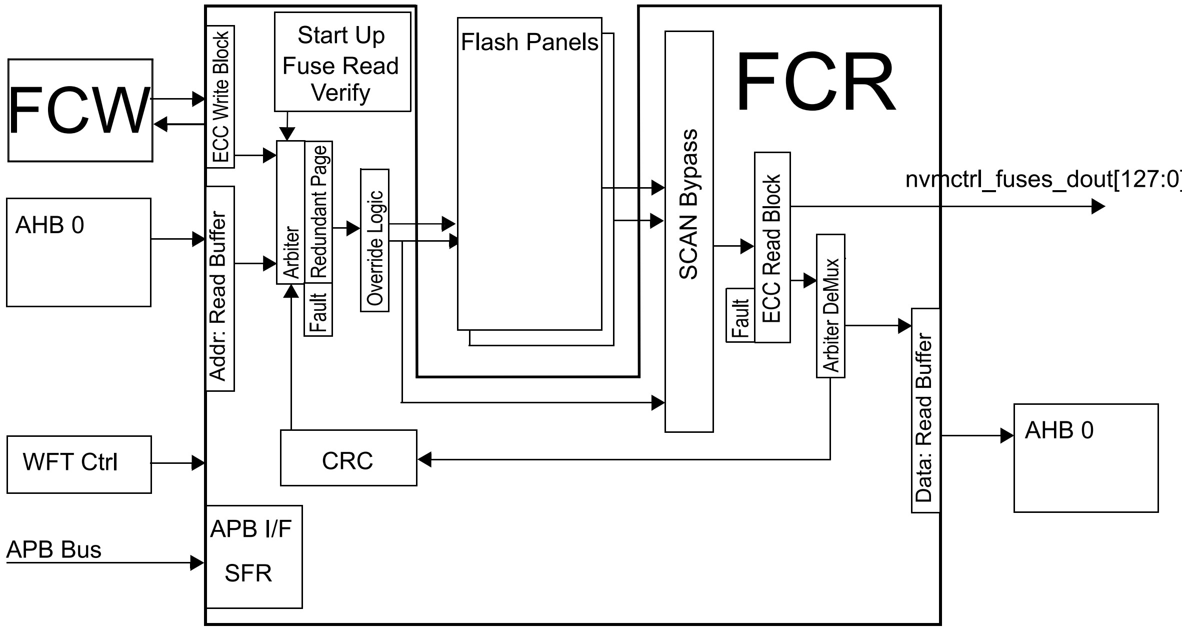
The following figure shows the NVM Controller within the context of the system's block diagram. There are two bus interfaces, one to the AHB Bus Matrix and one to Peripheral Bus A. The NVM Controller is built from two separate modules described below: the Flash Controller, Write (FCW) and Flash Controller, Read (FCR).

Figure 31-1. NVM Controller Location in the System Block Diagram



The FCW's main interface is with the FCR, as shown in the following figure. The FCR provides the interface to the AHB Bus Matrix and Peripheral Bus A. It also provides the interface to the two Flash panels in support of the rest of the device.

Figure 31-2. NVM Controller Detailed Block Diagram



31.2. Flash Controller, Write

31.2.1. Features

The following are key feature of the NVMCTRL module:

- Support Flash partitions
  - Program Flash Memory (PFM)
  - Boot Flash Memory (BFM)
  - Configuration Flash Memory (CFM)
- Mission Mode Write/Erase Features:
  - PFM Erase
  - Page Erase
  - Single Write (64-bit) + simple parity
  - Quad Write (256-bit) + SECDED ECC
  - Row Write (1KB) + SECDED ECC
    - Built-in DMA of data from Data RAM
  - All write operations support pre-program to increase endurance and retention
- BFM Address Swap
  - Supports Dual Boot
  - Support Live Update
- PFM Address Swap
- Debug Erase features are implemented by Boot ROM:
  - Identical to Mission Mode – PLUS:
    - Chip Erase: PFM and unprotected NVR pages
      - available to the Boot ROM only

    - NVR special pages must be unprotected to erase
    - NVR Test pages cannot be erased
    - ROM must lock feature not user accessible before it exits
- Debug Programming Features
  - Identical to Mission Mode
  - Set DAL commands – SDAL*x*
- Program Flash Memory (PFM) Write Protect, with 8 protection regions
  - Region defined by Base and Size
  - Lockable
- Boot Flash Memory (BFM) Write Protect
  - Individual (4 KB) Page write protect regions
  - Compatible with Dual Boot
  - Lockable
- Flash based User OTP pages
  - Not erased by Chip Erase or Page Erase
  - Four Lockable Write Protect regions per Page
  - Supports storage of user calibration values
- Boot ROM Support: Boot CFG and User CFG Write Protection

31.2.2. Peripheral Dependencies

Peripheral Name	Base Address	NVIC IRQ Index:Source	MCLK AHBx/APBx **Clock Enable Mask Bit**	PAC Peripheral ID **(PAC.WRCTRL.PERIDx)**	Power Domain
FCW	0x4400_4000	2 : MAIN	AHB : MCLK.CLKMSK0[5] APB : MCLK.CLKMSK1[2]	2	VDDREG

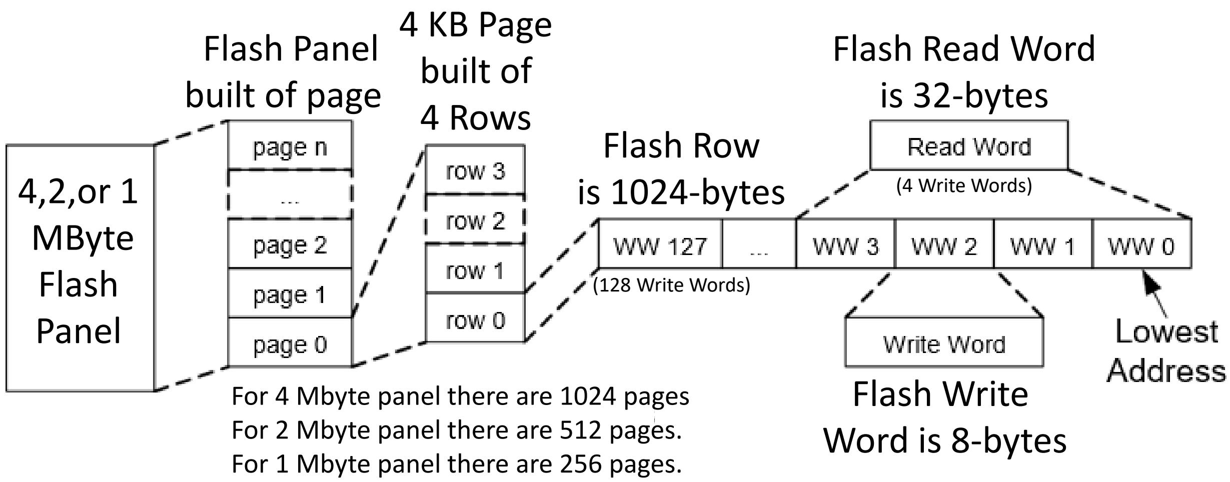
31.2.3. Definitions

The Flash Controller, Write (FCW) supports programming operations (writes/erases) for on-chip flash memory. This device's flash memories have 256-bit read data width and 64-bit write data width. There are extra bits which the FCW uses to implement Error Correction Code (ECC) for single error correct (SEC) and double error detect (DED).

A Flash panel is a single flash memory block. This device has a dual panel flash memory system, consisting of two identical 1/4, 1/2, or 1-Mbyte panels. The structure of each flash panel is shown in the following figure.

For the purpose of the Flash Controllers a "Word" is defined as 4-bytes or 32-bits. A Write Word (WW), the native write width of the Flash, is a Double Word, DWord, of 8-bytes. A Read Word, the native read width of the Flash, is a Quad Double Word, QDWord, of four Write Words or 32-bytes. "Flash Words" as described in this section are Read Words.

Figure 31-3. Flash Module Construction



Flash panels contain 16 extra bits per Read Word to support ECC. These bits are among the four Write Words of a Read Word (i.e. 4 extra bits/Write Word). Therefore, a Write Word is really 64 data bits and 4 ECC bits. Referring to DWords, Flash Words, QDWords implies the associated ECC bits also.

A Flash panel is divided into three logical Flash partitions:

- Program Flash Memory (PFM)
- Boot Flash Memory (BFM)
- Configuration Flash Memory (CFM)

The dual panel system provides two sets of logical Flash partitions, of which the PFM and BFM are identical, but CFM usage varies slightly between panels.

31.2.4. Flash Architecture

Each Flash partition is built up of several pages, also called sectors. The controller works with panels made from 4KB pages with each page containing 4 rows of Flash data. A row is the largest selectable region for contiguous programming of Write Words. A Row contains 128 Write Words.

A page of Flash is the smallest unit of memory that can be erased in a single operation. A panel's Program Flash Memory (PFM) space can also be erased in a single operation. All other erases use multiple operations.

A Write Word is the only unit of memory that can be programmed at a time. All other programming operations are made up of several contiguous Write Word Program operations. The FCW supports:

- **Row Write:** Write Word by Write Word programming until the whole Row is written. Data is read by the FCW from System SRAM.
- **Quad Write:** The FCW performs 4 writes of data from holding registers
- **Single Write:** The FCW performs one write of data from holding register(s)

31.2.4.1. Program Flash Memory (PFM)

PFM is the largest section of Flash memory. It is where the main application code resides. When a series of parts have the same feature but different Flash size, the PFM is the only partition that changes size.

Each panel contains PFM. The natural order is for Panel 1 PFM to exist in the lower address range and Panel 2 PFM to exist in the upper address range. The order in which each PFM exist in the Flash memory space can be controlled by software. The PFM address range is contiguous across both panels.

Dual panel systems support Live Update which allows reading from one panel while write to the other. It does not matter which of the logical regions are being accessed as long as they are in different panels.

31.2.4.2. Boot Flash Memory (BFM)

Boot Flash Memory is meant to support sophisticated boot loaders and therefore have 16 pages in each panel's BFM.

PIC32C devices have a Boot ROM to control the loading of configuration and to provide a root of trust for secure boot. When the Boot ROM is finished it sets up the CPU to start execution from the base of BFM.

With a dual panel system, each panel has BFM. The BFM can be used in either Dual Boot or Single Boot. Dual Boot designates each panel's BFM as a boot source. This allows the safe updating of one boot image while the other image stays intact. Single boot allows the boot image to span both panel's BFM space providing a larger boot code space.

31.2.4.3. Configuration Flash Memory (CFM)

CFM contains 16 pages (8 in each panel) dedicated to configuration of the device. (See the following table.) These pages have hardware-imposed restrictions on their usage. They are not for code and may not be accessible after the device is protected.

One page contains factory calibration data including the unique ID of the device . This page cannot be written or erased.

The Boot Configuration (Boot CFG) page contains pre-boot options to be configured by the Boot ROM. This page can be read protected and write/erase protected by the Flash system.

The User-OTP page implements a Flash-based One Time Programmable regions. It is always erase protected by hardware such that its content survives a Chip Erase operation. It is intended to store calibration values for external devices but is fully user defined.

The User Configuration (User CFG) page stores pre-boot options that have a different protection model than Boot CFG. This page can be write/erase locked.

The second panel's CFM contains additional pages to support dual boot. Boot CFG 2 and User CFG 2 allow safe updates of configuration when also updating application code in PFM or boot code in BFM.

Table 31-1. Flash CFM Configuration Address Map

Panel:	Address:		Size:	Contents:	Notes:
	Start:	End:			
1	0x0A00_0000	0x0A00_0FFF	4 KByte	User CFG-1	-
1	0x0A00_1000	0x0A00_1FFF	4 KByte	User OTP-1	-
1	0x0A00_2000	0x0A00_2FFF	4 KByte	BOOT CFG-1	-
1	0x0A00_3000	0x0A00_3FFF	4 KByte	RSVD	No access.
1	0x0A00_4000	0x0A00_4FFF	4 KByte	DAL CFG	-
1	0x0A00_5000	0x0A00_5FFF	4 KByte	RSVD	No access.
1	0x0A00_6000	0x0A00_6FFF	4 KByte	RSVD	No access.
1	0x0A00_7000	0x0A00_7FFF	4 KByte	CAL-OTP	Cannot be written or erased.
2	0x0A00_8000	0x0A00_8FFF	4 KByte	User CFG-2	-
2	0x0A00_9000	0x0A00_9FFF	4 KByte	User OTP-2	-
2	0x0A00_A000	0x0A00_AFFF	4 KByte	BOOT CFG-2	-
2	0x0A00_B000	0x0A00_BFFF	4 KByte	RSVD	No access.
2	0x0A00_C000	0x0A00_CFFF	4 KByte	RSVD	No access.
2	0x0A00_D000	0x0A00_DFFF	4 KByte	RSVD	No access.

Table 31-1. Flash CFM Configuration Address Map (continued)

Panel:	Address:		Size:	Contents:	Notes:
	Start:	End:			
2	0x0A00_E000	0x0A00_EFFF	4 KByte	RSVD	No access.
2	0x0A00_F000	0x0A00_FFFF	4 KByte	RSVD	No access.

Note: All RSVD addresses are "Address Holes" and therefore generate a bus error back to the initiator.

31.2.5. Erase and Word Write Flow

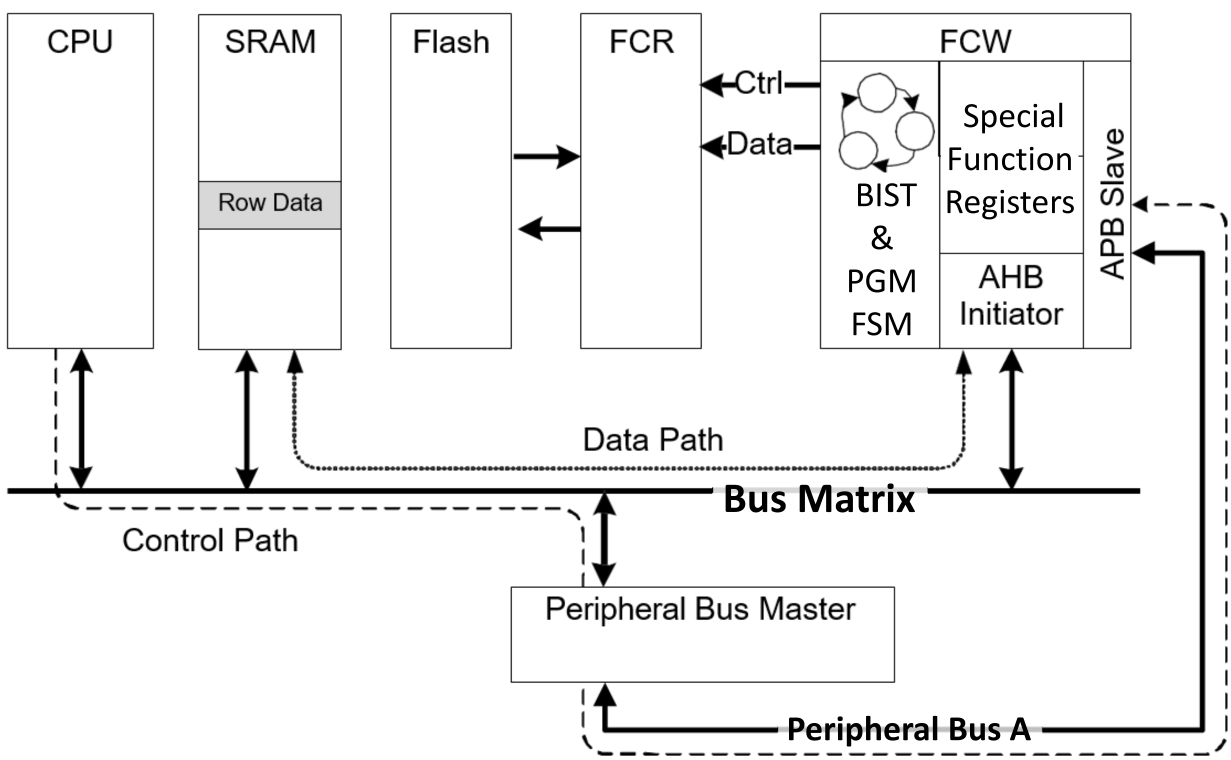
The address for erases and the address/data for word writes is stored by the CPU in the Special Function Registers (SFRs) of the FCW. The CPU stores this data using Peripheral Bus A. Flash operations are executing using the FCR (Flash Controller, Read). This is shown in the following figure.

Figure 31-4. Erase and Word Write Flow System Diagram

31.2.6. Row Write Flow

Row writes of Flash do not use data already stored in the FCW's registers. Instead the data is read directly from data RAM using the AHB bus matrix. This is shown in the following figure.

Figure 31-5. Row Write System Diagram



31.2.7. Flash Retention and Endurance

Flash panels have retention and endurance values native to the process technology in which they are designed. Two mechanisms exist to increase the endurance of a panel while maintaining retention: ECC, and Pre-Program. ECC masks native error rates by detection and correction. Since the ECC can correct one error per Flash Word it can double the endurance of a panel. When enabled, the ECC adds to the Flash read access time to perform the correction calculation. Pre-Program reduces stress when programming bit cells thereby doubling endurance. Pre-Program adds about 20 percent to the Flash programming time. Using both ECC and Pre-Program can yield 4x endurance improvement.

31.2.8. ECC Support

The FCW always writes ECC values for Quad Word and Row Write operations. The FCW always writes Parity for Single Write operations. The Flash ECC settings only determine if Single Write operations are allowed. Single Write operations are NOT allowed when Flash ECC settings are set to ECC Writes with ECC Reads. (For more information on ECC settings, see Section 3 FCR below.) For all Dynamic Write ECC selections, the FCW allows Single Write operations.

31.2.9. Clocks

The FCW peripheral bus clock (CLK_FCW_APB) can be enabled and disabled in the Main Clock Controller.

The FCW data bus clock (CLK_FCW_AHB) can be enabled and disabled in the Main Clock Controller.

The FCW also requires an on-chip 8 MHz clock source that is automatically configured without application assistance. The CLK_GEN_FCW is derived by dividing the 48 MHz trimmed internal RC oscillator by 6.

31.2.10. Security Features

Contact a local Microchip representative to obtain the details of this chapter.

31.2.11. Debug Access Level (DAL)

The FCW supports setting the DAL for code protection. DAL has two levels, “Access to All Memory” (DAL2) and “Limited Access” (DAL0). The controller only allows DAL to be programmed to a more restrictive level, that is a lower value. The factory default is DAL2. The NVMOP SDAL sets the Debug Access Level to the value in CTRLB.SDALCPU*x*, for *x* = 0 (M33 Core)). The DAL value must be the same or lower.

For SDAL, the steps are as follows:

1. Write CFGKEY to KEY.KEY.
2. Write required DAL in CTRLB.SDALCPU*x*.
3. Write WRKEY to KEY.KEY.
4. Write CTRLA.NVMOP to <Desired SDAL NVMOP>.
5. The FCW generates an interrupt when it clears STATUS.BUSY and sets INTFLAG.DONE.

The SDAL NVMOP performs an erase followed by a program of the new DAL value.

31.2.12. SFR Write Protect Features

A number of mechanisms exist within the device to ensure that inadvertent writes to program Flash do not occur. They are :

1. PAC write protection.
2. An unlock value (KEY) to allowing writing critical registers.
3. Local SFR bits that provide write locks for other SFR bits.
4. SFR bits that provide write/erase protection of Flash memory regions.

31.2.12.1. PAC SFR Write Protection

Registers with write access can be optionally write-protected by the *Peripheral Access Controller (PAC)*. Write protection does not extend to external debugger accesses. Note that PAC write protection is not applied during debug. The register descriptions below will show “PAC Write Protection” under “Properties:” for those registers that can be write protected using the PAC module.

31.2.12.2. SFR Write Unlock Values

The KEY register provides write protection to critical NVM registers. KEY uses multiple KEY Values to unlock access to classes of registers. Writing the specific KEY Value to the KEY register unlocks access to the defined set of registers. When the KEY Value protects a multitude of registers the unlock remains in effect until SW clears it. When the KEY Value protects a single register KEY clears after a successful write.

KEY is always cleared by a write to a protected register. This causes the INTFLAG.KEYERR bit to be set. SW can clear the unlock state by writing any invalid key to it (recommend all 0's). The INTFLAG.KEYERR is not set by this case.

Setting KEY.KEY = <CFGKEY> unlocks CTRLB, PWP*x* (*x* = 0,1,...,7), LBWP, UBWP, UOWP, and CWP. Setting KEY.KEY = <WRKEY> unlocks CTRLA. Setting KEY.KEY = <SWAPKEY> unlocks SWAP.

31.2.12.3. Local Lock Bits

Some SFR bits have associated local lock bits. These bits lock updates of the associated feature until the next reset. Lock bits are found in SWAP, PWP*x*, LBWP, UBWP, UOWP, and CWP.

31.2.13. FCW Sequencer User Model

31.2.13.1. Sequencer Enable Control

KEY.KEY must be written with the WRKEY value before CTRLA can be written to start a Write/Erase sequence. Once CTRLA is written, the KEY register returns to 0's (locks) and all registers in the FCW are write protected (STATUS.BUSY=1) until the operation finishes. Write protecting the registers prevents addresses, data and configuration changing in mid sequence.

31.2.13.2. Start Sequencer

Once the NVM system is configured the only registers necessary for programming are CTRLA, KEY, ADDR, and either DATAx or SRCADDR. Page Erase does not need DATAx or SRCADDR. Other Erases do not need ADDR.

For Write/Erase, the steps are as follows:

1. Lock the hardware write mutex by setting the LOCK bit to '1' and the OWNER field to '01' simultaneously to the MUTEX register. Ensure that these bits are set correctly before proceeding, if the OWNER field is not '01' and the LOCK bit is '1' another system has ownership of the hardware write mutex and this operation must be attempted again when that system releases the mutex.
2. Setup ADDR and if programming either DATAx (Single/Quad) or SRCADDR (Row Write).
3. Write WRKEY to KEY.KEY.
4. Write CTRLA.NVMOP to <Desired NVMOP>.
5. The FCW generates an interrupt when it clears STATUS.BUSY and sets INTFLAG.DONE.

When CTRLA is written, the write/erase sequence starts and the CPU is unable to execute from the selected Flash panel for the duration of the sequence.

31.2.14. Flash Write Protect Features

Flash write protect features are for safety not security. They protect against inadvertent erases or updates due to execution of errant code.

31.2.14.1. PFM Write Protect (PWP)

The PWP register provides write/erase protection for a region in PFM using PWP*x*.BASE and PWP*x*.SIZE. BASE is the flash physical address of the start of the region to be protected. BASE is aligned to the page size of the flash panel. SIZE is the number of pages to be protected. There are eight regions that can be protected.

Protection can be enabled or disabled using PWP*x*.PWPEN bit. However, any address range not defined by PWP*x* is not protected. When PWP regions with different protections overlap, the one with the more restrictive wins. Because of this, overlapping regions are not useful.

See the Interrupt Flag Summary Table for the effects of write protect errors on INTFLAG.

31.2.14.1.1. PFM Write Protect Mirror

With dual panel systems PWP can be mirrored from the Lower PFM to the Upper PFM OR from the Upper PFM to the Lower PFM. When PWP*x*.PWPMIR=1, PWP*x* is applied to both Upper and Lower PFM regions. The check simply masks the most significant address bit (BASE, SIZE, ADDR) used in the compare.

Individual mirror bits per PWP*x* regions allows the lower PFM to be protected while the upper region is not (or vice versa).

31.2.14.1.2. PWP Example

For example, to protect a 104KB region starting at 0x0134_7000:

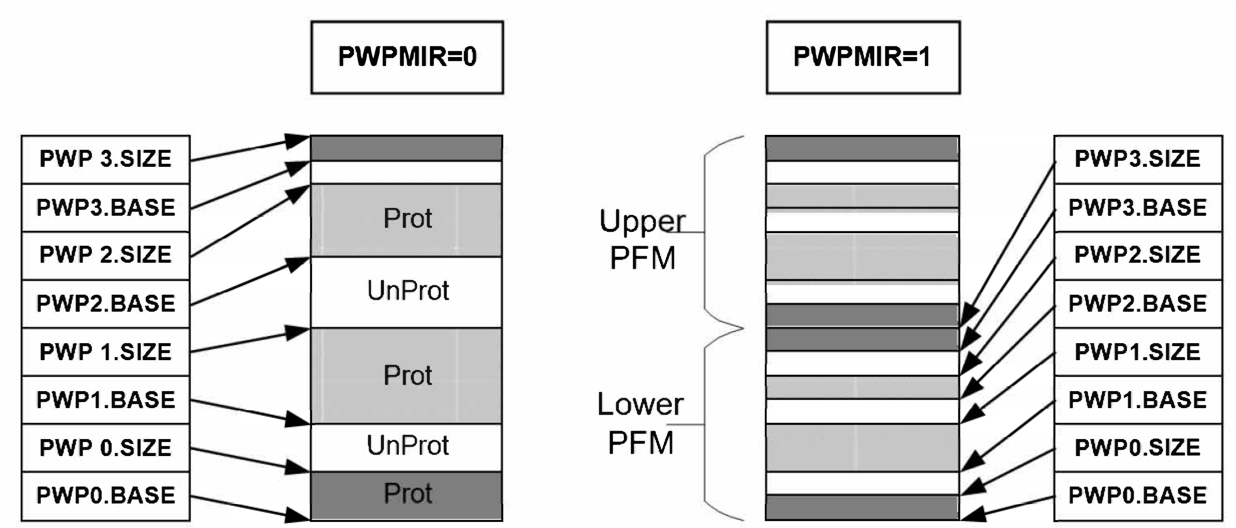
BASE = 0x0134_7000 << 8 = 0x3470_0000 SIZE = 104/4 = 26 = 0x1A

PWPEN = 0x0000_8000

PWPn = BASE | PWPEN | SIZE PWPn = 0x3470_801A

The following figure shows a graphical representation of Program Write Protect (PWP). Shaded regions are write/erase protected. Non-shaded regions are not protected.

Figure 31-6. PWP Example: No Mirror versus Mirror



31.2.14.1.3. PWPx Lock/Unlock Sequence

PWP*x* registers require a lock/unlock sequence as described in SFR Write Unlock Values. Any attempt to write to the PWP*x* register when locked has no effect.

31.2.14.1.4. Local Lock Bits

The user may optionally select to prevent further writes to a PWP*x* register by also setting the local LOCK bit (in the same register) when writing the PWP*x* values.

When the local lock bit is set (LOCK = 1), subsequent writes to that register (even if the register unlock sequence is followed) have no effect, creating a “write once’ register. Local LOCK bits revert to the unlocked state at reset.

31.2.14.2. Boot Write Protect

Note: This section uses BWP to refer to both LBWP and UBWP.

The BWP registers are used to write protect individual pages within BFM. The write protect bits are grouped into Lower Boot (LBWP[15:0]) and Upper Boot (UBWP{15:0]). Each group is directed to Panel 1 or Panel 2 depending upon which panel is determined to contain the specific Boot region.

At reset, all BWP bits are set to logic ‘1’, write protecting all user accessible BFM pages. Clearing a BFM page bit within its register removes write protect from the corresponding page.

See the Interrupt Flag Summary Table for the effects of write protect errors on INTFLAG.

31.2.14.2.1. BWP Lock/Unlock Sequence

LBWP and UBWP require a lock/unlock sequence as described in SFR Write Unlock Values. Any attempt to write to these registers when locked will have no effect.

31.2.14.2.2. Local Lock Bits

The user may optionally select to prevent further writes to a BWP register by also setting the local LOCK bit (in the same register) when writing the BWP value.

When the local lock bit is set (LOCK = 1), subsequent writes to that register (even if the unlock sequence is followed) have no effect, creating a “write once’ register. The local LOCK bit reverts to the unlocked state at reset.

31.2.14.3. Configuration Write Protect

The CWP register is used to write/erase protect individual pages within CFM. A single register supports the CFM in Panel 1 and Panel 2 for the dual Boot model (see Flash Write Protect Features). CFM does not change addressing when swapping BFM order.

At reset, all CWP bits are set to logic ‘1’, write protecting all user accessible CFM pages. Clearing a CFM page bit within its register removes write protect from the corresponding page.

The CWP register contains the User CFG and Boot CFG protection control. User-OTP is protected by other registers. DAL CFG is always protected for write/erase commands.

31.2.14.3.1. CWP Lock/Unlock Sequence

This register requires a lock/unlock sequence as described in SFR Write Unlock Values. Any attempt to write to this register when locked will have no effect.

31.2.14.3.2. Local Lock Bits

There are four separate “Lock” bits for ((boot,user) x (page 1, page2)) that can be locked in place to prevent disabling erase and write protection. Writing the “Lock” bit at the same time as the “Write Protect” bit creates a “write once” register field. The “Lock” bit, once set, reverts only after a reset.

31.2.14.4. User-OTP Write Protect

One page in the Flash boot region (for each panel) is implemented as One-Time Programmable NVM called “User-OTP”. It can only be written and cannot be erased (by page or chip erase). It is intended for the storage of user system calibration data that must survive a Flash erase.

User-OTP can be used to store and preserve identification values such as an Ethernet MAC address, OEM/ODM version numbers, board version numbers, etc. Another use would be for calibration values of circuits external to the chip but constant for a board design in which the chip exists.

The UOWP.UO*n*WP [3:0] (*n* = 1 or 2) field is used to write protect each of the four rows in the User-OTP page separately.

31.2.14.4.1. UOWP Lock/Unlock Sequence

The UOWP register requires a lock/unlock sequence as described in SFR Write Unlock Values. Any attempt to write to this register when locked has no effect.

31.2.14.4.2. Local Lock Bits

The user may optionally select to prevent further writes to the write protect by also setting associated UOWP.UO*n*WPRLOCK bit (in the same register) when writing the UO*n*WP value.

When a local lock bit is set, subsequent writes to change the corresponding UO*n*WP bit (even if the unlock sequence is followed) have no effect, creating a “write once’ register. Local lock bits revert to the unlocked state at reset.

31.2.15. Write Sequences

Reads from the panel being programmed stall until the programming sequence finishes. This is because FCW programming requests have the higher priority than read requests.

Note: Performing any programming sequence when power is lost results in an undefined increased wear of the panel. Loss of power is defined as receiving a POR event.

31.2.15.1. Pre-Program

The Flash panels in this device support an option to Flash programming that increase endurance of the panel. This feature is called Pre-Program. When selected the controller automatically initiates two programming cycles per write. The first is a short cycle and the second is a full cycle. This option adds about 20% to the programming time.

To use this feature set the CTRLA.PREPG bit when writing CTRLA.NVMOP. In all other respects the SFR setup is identical. Pre-Program typically doubles the native Endurance and Retention of the panel. It is recommended to either **always** use or **never use** Pre-Program. Mixed use of PREPG yields undefined endurance and retention of the panel.

It is highly recommended to always have CTRLA.PREPG = 1.

31.2.15.2. Single Write Sequence

The smallest block of data that can be programmed with a single write is one DWord (aka a Write Word). The data to be programmed must be written to DATA0 and DATA1 before the programming sequence is initiated. The FCW programs the DATA*x* into the DWord location pointed to by ADDR. ADDR is a byte address, so the lower 3 bits are ignored for DWord programming.

A DWord Program sequence typically comprises the following steps:

1. <Desired NVMOP> is Single Write.
2. Follow the Start Sequencer from Start Sequencer.
3. Wait for NVM Interrupt from Interrupts.
4. Check the INTFLAG bits to ensure that the program sequence completed successfully, and then clear all bits in INTFLAG. See Errors and Flags about error flags.
5. Unlock the hardware write mutex by setting the LOCK bit to ‘0’ and the OWNER field to ‘00’ simultaneously to the MUTEX register.

A Word/DWord within the Flash can be programmed if its associated page write protection is not enabled, see Debug Access Level.

31.2.15.2.1. Single Write Timing

DWord Program timing is dominated by setup (Tnvs + Tpgs), program time (Tprog) and recovery (Trcv) delays. Using the timing shown in Non-Volatile Memory Controller (NVM) Electrical Specifications, the total time to program one word is roughly shown as follows:

DWord Program time = T_{FW}

If Pre-Programming, the first step adds roughly: DWord Program time = Tpreprog+ Tprepgh + Tprepgs

31.2.15.2.2. Single Write and ECC

Single Writes are not allowed when ECC mode is set to 'ECC Writes with ECC Reads'. The Interrupt Flag Summary table defines the interrupt flags that are set.

31.2.15.3. Quad Write Sequence

The Quad Write operation implements a Quad Double Word programming. The eight 32-bit instruction/data words to be programmed must be written to DATA*x*, *x* = 0,1,...,7, before the programming sequence is initiated. The Flash instruction word at the location pointed to by ADDR is programmed. The program operation must be aligned to a Flash word address, so the lower 5 bits are ignored for Quad Write.

The Quad Write sequence typically comprises the following steps

1. <Desired NVMOP> is Quad Write.
2. Follow the Start Sequencer from Start Sequencer.
3. Wait for NVM Interrupt from Interrupts.
4. Check the INTFLAG bits to ensure that the program sequence completed successfully, and then clear all bits in INTFLAG. See Errors and Flags about error flags.
5. Unlock the hardware write mutex by setting the LOCK bit to '0' and the OWNER field to '00' simultaneously to the MUTEX register.

A flash address can be written if its associated page write protection is not enabled, see Debug Accesss Level.

31.2.15.3.1. Quad Write Timing

Quad Write timing is dominated by setup (Tnvs + Tpgs), program time (Tprog) and recovery (Trcv) delays. Using the specs shown in Non-Volatile Memory Controller (NVM) Electrical Specifications, the total time to program one QWord/QDWord is roughly:

Quad Write Time = Tnvs + Tpgs + 4*Tprog + Trcv + Trw

If Pre-Programming, the first step adds roughly: Quad Write Time = 4*Tpreprog + Tprepgh + Tprepgs +

Quad Write asserts programming voltage throughout the sequence to avoid duplicating setup and recovery delays. Consequently, a Quad Write operation takes less time than 4 Word/DWord Program operations.

31.2.15.3.2. Quad Write and ECC

Note: All configuration data held in flash and read before user boot up must be programmed, by software, using Quad Write or it will generate an ECC error when read.

31.2.15.4. Row Write Sequence

The largest block of data that can be programmed by a single NVMOP command is a row. As shown in Flash Module Construction, a row is 1024 bytes of data. ADDR is the row aligned address where the Flash address starts programming the data. The controller ignores the sub-row address bits and ALWAYS starts programming at the beginning of a row.

A Row Write sequence comprises the following steps:

1. Write the entire row of data to be programmed into system SRAM. The source address must be word aligned and secure/non-secure zone consistent with the transaction type.
2. <Desired NVMOP> is Row Write.
3. Follow the Start Sequencer from Start Sequencer.
4. Wait for NVM Interrupt from Interrupts.
5. Check the INTFLAG bits to ensure that the program sequence completed successfully, and then clear all bits in INTFLAG. See Errors and Flags about error flags.
6. Unlock the hardware write mutex by setting the LOCK bit to '0' and the OWNER field to '00' simultaneously to the MUTEX register.

A row of Flash can be programmed if its associated page is not write protection see Debug Access Level.

The data transferred from RAM is buffered within the FCW. All data values must be in place before the ECC can be derived and the programming operation can start.

The FCW automates programming the data into the Flash using four contiguous Single Write operations. The data is double buffered, such that each programming sequence of four programming operations may execute while the next data are read from the internal system SRAM.

Notes:

1. The base address held in the SRCADDR register is not changed during the Row Write sequence.
2. The DATA*x*, *x* = 0,1,...,7, registers are not used by the Row Write sequence., since the data is read from internal system SRAM.

31.2.15.4.1. Row Write Timing

Row Write timing is dominated by setup (Tnvs + Tpgs), program time (Tprog) and recovery (Trcv) delays.

The number of writes needed per row, *n*, is 128. So, the total time to program one row is roughly:

Row Write Time = Tnvs + Tpgs + *n**Tprog + Trcv + Trw

If Pre-Programming, the first step adds roughly: Row Write Time = *n**Tpreprog + Tprepgh + Tprepgs

Row Write asserts programming voltage throughout the sequence to avoid duplicating setup and recovery delays.

31.2.15.4.2. Row Write and ECC

Since Row Write uses multiple Quad Writes to update a row, it has the same features and restrictions as Quad Write.

31.2.16. Erase Sequences

Note:
Any access to the panel containing the page or panel being erased stalls reads from all panels until the erase finishes.

31.2.16.1. Page Erase Sequence

A page erase performs an erase of a single page of PFM or BFM or select pages in CFM (User, Boot). The page to be erased is selected using ADDR. The lower bits of the address given by ADDR are ignored in page selection.

Note: It is the responsibility of the user code to ensure that no bus initiators, including the CPU, access the Flash region, the entire PFM of the panel, or the entire Boot of the panel, that is to be or being erased.

31.2.16.1.1. Page Erase

A Page Erase sequence comprises the following steps:

1. <Desired NVMOP> is Page Erase.
2. Follow the Start Sequencer from Start Sequencer.
3. Wait for NVM Interrupt from Interrupts.
4. Check the INTFLAG bits to ensure that the program sequence completed successfully, and then clear all bits in INTFLAG. See Errors and Flags about error flags.
5. Unlock the hardware write mutex by setting the LOCK bit to ‘0’ and the OWNER field to ‘00’ simultaneously to the MUTEX register.

A page of Flash can be erased if its associated page write protection is not enabled, see Debug Access Level.

31.2.16.1.2. Page Erase Timing

Page Erase timing is dominated by setup (Tnvs), erase time (Terase) and recovery (Trcv) delays. Using the timing shown in Non-Volatile Memory Controller (NVM) Electrical Specifications, the total time to erase one page is:

Page Erase = T_{FEP}

31.2.16.2. PFM Single Panel Erase

Note: Single Panel Erase obeys the values of the PWP*x* registers.

The PFM within a single panel may be erased using the respective Higher Addressed or Lower Addressed Panel Erase NVMOP command. These commands leave the BFM and CFM pages intact and are intended to be used by a field upgradeable device. To guard against accidental erasure user code can setup PWP for protection.

The ability to erase a single panel makes feasible the concept of executing from one panel while erasing (or writing) the other, without the need to complete many Page Erase operations.

A Panel Erase sequence comprises the following steps:

1. <Desired NVMOP> is Upper or Lower Program Erase.
2. Follow the Start Sequencer from Start Sequencer.
3. Wait for NVM Interrupt from Interrupts.
4. Check the INTFLAG bits to ensure that the program sequence completed successfully, and then clear all bits in INTFLAG. See Errors and Flags about error flags.
5. Unlock the hardware write mutex by setting the LOCK bit to ‘0’ and the OWNER field to ‘00’ simultaneously to the MUTEX register.

A single panel can be erased if its associated page write protection is not enabled, see Debug Access Level.

If in mission mode, the application must NOT be executing from the erased region. If any initiator reads from the panel containing the region being erased, all Flash accesses are stalled until the erase completes.

31.2.16.3. PFM Dual Panel Erase

All the PFM (both panels) may be erased using the Program Erase NVMOP command. This mode leaves all BFM and CFM pages intact and is intended to be used by a field upgradeable device. To guard against accidental erasure user code should setup PWP*x* appropriately.

The PFM Erase operation is not address based (i.e. the FCW directly selects the target flash panel). The ADDR register is ignored so does not need to be initialized for this NVMOP.

A Program Erase sequence comprises the following steps:

1. <Desired NVMOP> is Program Erase.
2. Follow the Start Sequencer from Start Sequencer.
3. Wait for NVM Interrupt from Interrupts.
4. Check the INTFLAG bits to ensure that the program sequence completed successfully, and then clear all bits in INTFLAG. See Errors and Flags about error flags.
5. Unlock the hardware write mutex by setting the LOCK bit to ‘0’ and the OWNER field to ‘00’ simultaneously to the MUTEX register.

All PFM can be erased if all associated page write protection is not enabled, see Debug Access Level.

If in mission mode, the application must NOT be executing from the erased region. If any initiator reads from the panel containing the region being erased, all Flash accesses are stalled until the erase completes.

31.2.16.3.1. Program Erase Timing

Program Erase timing is dominated by setup (Tnvs), erase time (Terase) and recovery (Trcv) delays. Using the timing shown in Non-Volatile Memory Controller (NVM) Electrical Specifications:

Program Erase = 2 * (Tnvs + Terase + Trcv + Trw)

31.2.17. Support for Boot Code

The NVM Controller supports:

- Dual Boot regions together with an automated means to select the Lower Boot area out of reset
- A Single Boot region using up to the maximum number of Boot pages available
- A Secure Boot page within each Boot region in both Single and Dual Boot modes

31.2.17.1. Dual Panel BFM Organization Notes

It is worth restating that the pages in CFM do not swap address locations when the Boot Swap changes. The usage of the User-OTP can impact code running from upper or lower regions based on the swap setting for BFM and PFM.

31.2.17.2. Dual Boot Mode

The Dual Boot scheme allows the old boot code to be preserved in one flash panel until the new boot code is successfully programmed and verified in the other Flash panel. Which Flash panel is used is controlled by the BootROM.

Each panel's BFM region has a sequence number (SeqNum) located in the USER_CFG page for that panel. After reset and during the Flash configuration period (i.e. prior to system execution from BFM), SeqNum for each BFM region is read by the Boot ROM and compared. The Boot region with the higher SeqNum is selected as the Lower Boot region.

Each sequence number stored into the Flash consists of a ‘true’ and ‘complement’ 16-bit (halfword) value held in a single 32-bit word. The ‘true’ portion is held in the LS-halfword and the complement portion is held in the MS-halfword.

Note: All configuration data held in Flash and read by the Boot ROM at start up must be programmed, by software, using Quad Write in order to for the configuration data to have ECC protection.

In the event of an invalid Boot sequence number, the Boot ROM defaults to map the panel with the valid sequence number to Lower Boot. A Boot sequence number is considered to be invalid if:

- TrueValue != ~(ComplementValue) or TrueValue^ComplementValue != 0xFFFF

Or

An ECC DED error has occurred

In the following cases:

- The sequence numbers are equal
- Both sequence numbers are invalid

The Boot ROM selects Panel 1 to be the Lower Boot region.

The Flash Controllers, FCR (for reads) and the FCW (for writes/erases), use SWAP.BFSWAP to map the BFM of each panel into the Lower and Upper regions of the device memory map (refer to the example in *Device Memory Map Example*) according to the settings implemented by the BootROM.

31.2.17.2.1. Boot Loader Operation

Updating boot code is the responsibility of the Boot Loader. The Upper Boot region is always the target of the update. The Boot Loader must verify the erase status, typically done by erasing the Upper Boot region before programming it with the new Boot code.

If the new code successfully verifies, the Boot Loader assigns a sequence number, SeqNum, higher than that of the old boot code. On a subsequent reset, the Boot ROM selects the new Boot code to be the Lower Boot region (and the old Boot code appears in the Upper Boot region).

If a failure occurs during the process of updating the Upper Boot region with new boot code, the old boot code remains valid in the Lower Boot region. As long as the new boot code does not receive a successful write of a SeqNum greater than the old boot code's, the old boot code is selected on a reset.

31.2.17.2.2. Dual Boot and Boot Page Protection

Although the BWP registers allow for different protection of Upper and Lower boot, in a Dual Boot situation it is desirable for them to protect the same page offsets.

31.2.17.2.3. Boot Panel Manual Swap

As discussed in Dual Boot Mode, the Dual Panel Lower Boot region selection is handled automatically at reset/boot time. The bit SWAP.BFSWAP shows the SeqNum controlled state of the Lower Boot order.

In addition to automatic control of boot, the user has the ability to change the Lower Boot panel manually with BFSWAP. The likely scenario for manual override is when both boot regions are programmed with identical code (which is recommended). Doing so allows write and erase of either Lower or Upper PFM without interruption of access to the boot region. For example, if Lower Boot is in Panel 1 and the Lower PFM address region is also in Panel 1, programming Panel 1's PFM region causes accesses to the Lower Boot region to be stalled until the programming operation finishes. If both Boot regions are programmed identically, the user can simply invert BFSWAP so boot code is fetched from the other panel.

31.2.17.3. Single Boot Model

Using a Single Boot region model allows the user to make use of almost all the Boot memory provided. The Boot ROM does not differentiate between Dual and Single Boot models, so Boot ROM always reads and tests the sequence numbers from each panel. Consequently, it is the user's responsibility to ensure that the sequence number, SeqNum, for each panel remains consistent with the desired panel order. It is highly recommended that the default Lower Boot region have a higher SeqNum than the default Upper Boot region, therefore preserving their natural order (See Device Memory Map Example).

31.2.17.3.1. Single Boot and Boot Page Protection

When operating with a Single Boot model, it is recommended that all Boot pages containing code be write protected. This could include pages within both the Lower and Upper address regions. It is also recommended to lock the BFSWAP bit (SWAP.BFSLOCK=1).

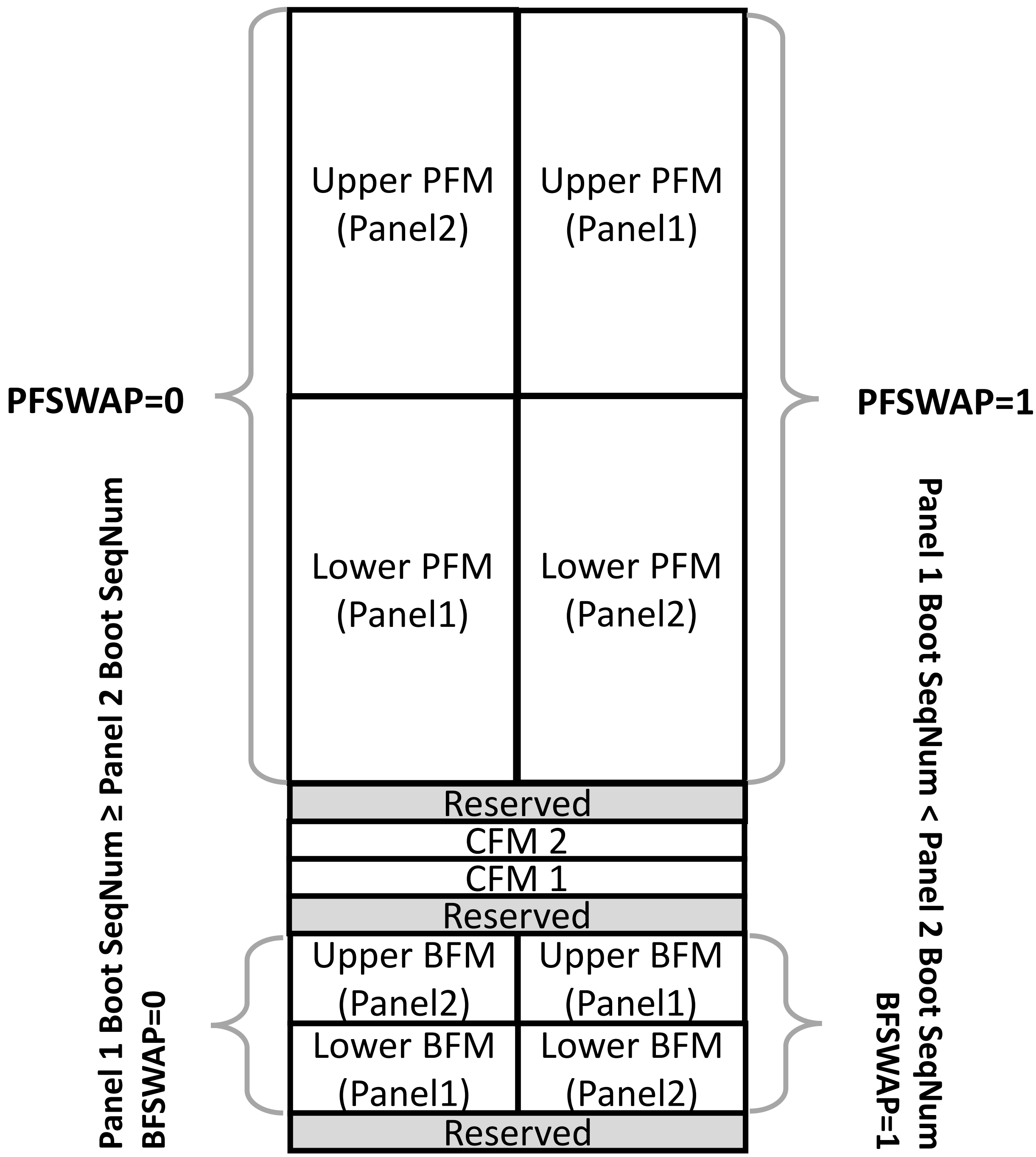
31.2.17.4. PFM Panel Manual Swap

Unlike Boot Swap, the memory region order of the PFM panels is entirely the user's responsibility. The control bit, SWAP.PFSWAP, provides the manual control to swap the order of the PFM panels. In order to properly swap the PFM panel regions, there must not be any accesses to either panel when changing the PFSWAP bit. Accesses to Boot regions or SRAM is allowed.

A likely method to implement PFM panel swap in boot code is to follow the same procedure as the FCW uses for Boot Swap. The code would be in the Lower Boot region but the sequence numbers, SeqNum, would be read from each panel.

An example device memory map is shown in the following figure with the PFSWAP bit set or cleared.

Figure 31-7. Device Memory Map Example



31.2.17.5. Swap Protection

It may be desirable to prevent swapping BFM and/or PFM after they are setup. The SWAP.BFSLOCK and SWAP.PFSLOCK allow BFSWAP and PFSWAP, respectively, to be write locked, preventing them from changing. Each lock bit also locks itself. Lock is in effect until the next reset.

31.2.18. Operation Timing

The FCW supports only self-timed operations using an internal clock source. The timing values for writing and erasing the Flash are predefined based on the Flash implemented in the system. No user involvement is need for timing value selection.

The AHB bus host initiator operates independently of the FSM, transferring data to the FCW data buffer only when requested. Consequently, there are no timing restrictions with respect to AHB and the FSM. However, for Row Write operations, the AHB interface must have sufficient bandwidth to SRAM to keep up and therefore not starve the FCW for data.

31.2.19. DMA

The FCW does not use the DMA module. It has its own built-in DMA that uses the AHB Bus Matrix directly when executing Row Writes.

31.2.20. Interrupts

The FCW has two sets of interrupts, which are serviced by the FCW interrupt vector. The main set of INTENCLR, INTENSET, and INTFLAG are for the system CPU.

The module has an interrupt output for CPU initiated NVMOPs that is controlled by INTENSET and INTENCLR. The interrupt status is reported via INTFLAG. A read of either INTENSET or INTENCLR returns the enable state for the specific interrupt condition. The Flash interrupt is asserted if a condition is enabled and its associated bit is set in INTFLAG. Interrupt flags are cleared by writing a 1 to its bit position in the INTFLAG register.

It is recommended to always enable DONE as interrupt source in INTENSET. Using DONE masks side effects of WRERR, RSTERR, HTDPGM, BUSERR, and FIFOERR being set before the FCW has finished cleaning up after an error.

STATUS.BUSY indicates if the FCW is performing an operation (from the CPU). INTFLAG.DONE indicates that the FCW has completed an NVMOP for the CPU.

Note: The FCW does not clear DONE automatically. Also, DONE=1 does not prevent NVM operations. Therefore, software must clear DONE before the next NVM operation is started.

31.2.21. Events

The FCW does not use events or generate them.

31.2.22. Errors and Flags

Several non-typical events can occur that affect the operation of the FCW causing Write/Erase operations to be aborted. Setup errors can occur that prevent the start of an NVMOP command. All these errors are reported via INTFLAG. Some flags are updated before the FSM clears BUSY. However, the FSM always clears BUSY and sets DONE on the same clock edge.

The following table describes the types of errors the FCW detects and what flags they affect.

Table 31-2. Interrupt Flag Summary

INTFLAGs:	Source:	Effect on FCW Operations:
WRERR, DONE	Any Error Event during NVMOP	Last address programmed may be corrupted, or last erase may not be complete.
RSTERR, DONE	BOR circuit not Ready before NVMOP written	NVMOP not started[2] SW should enable BOR and wait for it to be ready.
RSTERR[1]	Reset or Low Voltage Event before FSM starts NVMOP	NVMOP not started[2]
RSTERR[1], WRERR	Reset or Low Voltage Event after FSM starts NVMOP	Last address programmed may be corrupted, or last erase may not be complete. Software should verify results.
HTDPGM, DONE	High Temp Event before FSM starts NVMOP	NVMOP not started[2]

Table 31-2. Interrupt Flag Summary (continued)

INTFLAGs:	Source:	Effect on FCW Operations:
WRERR, SECERR, DONE	Security module or tamper event preempted operation	Last address programmed may be corrupted, or last erase may not be complete.
KEYERR	KEY not valid for SFR write	SFR not updated
CFGERR, DONE	ADDR not in flash memory space Disallowed CPUDAL setting.	NVMOP is not started (2)
CFGERR, OPERR, DONE	NVMOP not allowed because of ECC, i.e. Single Write attempted when ECC Write with ECC Read is selected	NVMOP is not started (2)
WRERR, FIFOERR, DONE	Row Write FIFO under run error	NVMOP is aborted(2) Subsequent addresses not programmed
BUSERR, FIFOERR(3), DONE	AHB Bus Host error before FIFO filled Possible SRCADDR out of range	NVMOP is not started(2)
BUSERR, FIFOERR(3), WRERR, DONE	AHB Bus Host error Possible SRCADDR out of range or access permission violation	NVMOP is aborted(2) Current and subsequent addresses not programmed
WPERR, DONE	ADDR is write protected by a WP register (i.e BWP, PWP, etc)	NVMOP is not started(2)
WPERR, SECERR, DONE	ADDR is write protected by a different security association(4)	NVMOP is not started(2)
OPERR, DONE	NVMOP is not available (e.g. RSVD)	NVMOP is not started(2)
SECERR	Security Violation	SFR access blocked Does not generate a bus error

Notes:
1. If BOR causes a reset, DONE will not be set because it is cleared by the reset.
2. NVMOP is generally considered started when the panel is enabled.
3. Error code may or may not be set.

The following table summarizes events that cause the Write/Erase sequence to not occur but do not set the error flag.

Table 31-3. Conditions Without an Error Flag

Source:	Effect on FCW Operations:
Writing KEY with an invalid value.	Locks all registers protected by KEY.
Attempted write of an SFR while locked for programming (BUSY=1) from the same security association.	Action is ignored by FSM and SFRs. Reports a bus error.
Attempted read or write of an SFR address that is not implemented.	Action is ignored by FSM and SFRs. Reports a bus error.

31.2.22.1. Write Error (INTFLAG.WRERR)

The FCW Finite State Machine (FSM) set the WRERR flag only if the FSM aborts a Write/Erase operation (i.e. an NVMOP command). The FSM sets the flag on the detection of an error which is before it clears BUSY and sets DONE. WRERR is only set if there is possible corruption of the flash due to the NVMOP abort.

31.2.22.2. Reset and Low Voltage (BOR) Error (INTFLAG.RSTERR)

If a reset or BOR event occurs while an NVMOP command is underway, the FCW sets flags as per the Interrupt Flag Summary table and terminates the command. A BOR event is likely to cause a system reset, preventing an interrupt from occurring. However, the affected interrupt flags maintain state through a reset caused by a BOR event. Therefore, software can check the prior state of the FCW. The devices RCAUSE register can be checked to see if the error is from a general reset or from a BOR. However, if the system shows a POR occurred, software has no guaranteed way of knowing if a write/erase operation was underway at the time the POR occurred.

31.2.22.3. Power-on Reset

A Power-on Reset (POR) causes immediate termination of any FCW operation. A POR is a full and hard reset of all states of the FCW and therefore no error flags originating from a prior operation survive the reset.

Several SFR bits reset by POR survive a BOR or user reset (WDT Reset, RESET pin, GCLK.CTRLA.SWRST, etc.,). Doing so allows NVM operations to report errors due to reset aborts. This also prevents values like address from changing when cleaning up an aborted operation.

31.2.22.4. Reset

A user reset during an NVM operation can but does not necessarily cause an error. Atomic operations like single or Quad Write, and page or upper/lower PFM erase finish before reset. The completion of these operations likely delays system exit from reset. Non-atomic operations like SDAL, Row Write, and PFM Erase may early terminate with an error flag set, typically WRERR.

Several SFR bits reset by POR survive a user reset (WDT Reset, RESET pin, GCLK.CTRLA.SWRST, etc). Doing so allows NVM operations to report errors due to reset aborts. This also prevents values like address from changing when cleaning up an aborted operation.

31.2.23. Register Summary

For descriptions and definitions of both Register and bitfield properties, refer to Register Properties.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRLA	31:24								
		23:16								
		15:8								
		7:0	PREPG				NVMOP[3:0]			
0x04	CTRLB	31:24								
		23:16								
		15:8								
		7:0							SDALCPU0[1:0]	
0x08 ... 0x0B	Reserved									
0x0C	INTENCLR	31:24								
		23:16								
		15:8			WRERR	RSTERR				
		7:0	SECERR	OPERR	WPERR	BUSERR	FIFOERR	CFGERR	KEYERR	DONE
0x10	INTENSET	31:24								
		23:16								
		15:8			WRERR	RSTERR				
		7:0	SECERR	OPERR	WPERR	BUSERR	FIFOERR	CFGERR	KEYERR	DONE
0x14	INTFLAG	31:24								
		23:16								
		15:8			WRERR	RSTERR				
		7:0	SECERR	OPERR	WPERR	BUSERR	FIFOERR	CFGERR	KEYERR	DONE
0x18	STATUS	31:24								
		23:16								
		15:8								
		7:0								BUSY
0x1C	KEY	31:24	KEY[31:24]							
		23:16	KEY[23:16]							
		15:8	KEY[15:8]							
		7:0	KEY[7:0]							
0x20	ADDR	31:24	ADDR[31:24]							
		23:16	ADDR[23:16]							
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0x24	SRCADDR	31:24	SRCADDR[31:24]							
		23:16	SRCADDR[23:16]							
		15:8	SRCADDR[15:8]							
		7:0	SRCADDR[7:0]							
0x28	DATA0	31:24	DATA31	DATA30	DATA29	DATA28	DATA27	DATA26	DATA25	DATA24
		23:16	DATA23	DATA22	DATA21	DATA20	DATA19	DATA18	DATA17	DATA16
		15:8	DATA15	DATA14	DATA13	DATA12	DATA11	DATA10	DATA9	DATA8
		7:0	DATA7	DATA6	DATA5	DATA4	DATA3	DATA2	DATA1	DATA0
0x2C	DATA1	31:24	DATA31	DATA30	DATA29	DATA28	DATA27	DATA26	DATA25	DATA24
		23:16	DATA23	DATA22	DATA21	DATA20	DATA19	DATA18	DATA17	DATA16
		15:8	DATA15	DATA14	DATA13	DATA12	DATA11	DATA10	DATA9	DATA8
		7:0	DATA7	DATA6	DATA5	DATA4	DATA3	DATA2	DATA1	DATA0
0x30	DATA2	31:24	DATA31	DATA30	DATA29	DATA28	DATA27	DATA26	DATA25	DATA24
		23:16	DATA23	DATA22	DATA21	DATA20	DATA19	DATA18	DATA17	DATA16
		15:8	DATA15	DATA14	DATA13	DATA12	DATA11	DATA10	DATA9	DATA8
		7:0	DATA7	DATA6	DATA5	DATA4	DATA3	DATA2	DATA1	DATA0
0x34	DATA3	31:24	DATA31	DATA30	DATA29	DATA28	DATA27	DATA26	DATA25	DATA24
		23:16	DATA23	DATA22	DATA21	DATA20	DATA19	DATA18	DATA17	DATA16
		15:8	DATA15	DATA14	DATA13	DATA12	DATA11	DATA10	DATA9	DATA8
		7:0	DATA7	DATA6	DATA5	DATA4	DATA3	DATA2	DATA1	DATA0

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x38	DATA4	31:24	DATA31	DATA30	DATA29	DATA28	DATA27	DATA26	DATA25	DATA24
		23:16	DATA23	DATA22	DATA21	DATA20	DATA19	DATA18	DATA17	DATA16
		15:8	DATA15	DATA14	DATA13	DATA12	DATA11	DATA10	DATA9	DATA8
		7:0	DATA7	DATA6	DATA5	DATA4	DATA3	DATA2	DATA1	DATA0
0x3C	DATA5	31:24	DATA31	DATA30	DATA29	DATA28	DATA27	DATA26	DATA25	DATA24
		23:16	DATA23	DATA22	DATA21	DATA20	DATA19	DATA18	DATA17	DATA16
		15:8	DATA15	DATA14	DATA13	DATA12	DATA11	DATA10	DATA9	DATA8
		7:0	DATA7	DATA6	DATA5	DATA4	DATA3	DATA2	DATA1	DATA0
0x40	DATA6	31:24	DATA31	DATA30	DATA29	DATA28	DATA27	DATA26	DATA25	DATA24
		23:16	DATA23	DATA22	DATA21	DATA20	DATA19	DATA18	DATA17	DATA16
		15:8	DATA15	DATA14	DATA13	DATA12	DATA11	DATA10	DATA9	DATA8
		7:0	DATA7	DATA6	DATA5	DATA4	DATA3	DATA2	DATA1	DATA0
0x44	DATA7	31:24	DATA31	DATA30	DATA29	DATA28	DATA27	DATA26	DATA25	DATA24
		23:16	DATA23	DATA22	DATA21	DATA20	DATA19	DATA18	DATA17	DATA16
		15:8	DATA15	DATA14	DATA13	DATA12	DATA11	DATA10	DATA9	DATA8
		7:0	DATA7	DATA6	DATA5	DATA4	DATA3	DATA2	DATA1	DATA0
0x48	SWAP	31:24								
		23:16								
		15:8							PFSLOCK	PFSWAP
		7:0							BFSLOCK	BFSWAP
0x4C	PWP0	31:24					PWPBASE[11:8]			
		23:16	PWPBASE[7:0]							
		15:8	PWPEN	PWPLOCK	PWPMIR		PWPSIZE[11:8]			
		7:0	PWPSIZE[7:0]							
0x50	PWP1	31:24					PWPBASE[11:8]			
		23:16	PWPBASE[7:0]							
		15:8	PWPEN	PWPLOCK	PWPMIR		PWPSIZE[11:8]			
		7:0	PWPSIZE[7:0]							
0x54	PWP2	31:24					PWPBASE[11:8]			
		23:16	PWPBASE[7:0]							
		15:8	PWPEN	PWPLOCK	PWPMIR		PWPSIZE[11:8]			
		7:0	PWPSIZE[7:0]							
0x58	PWP3	31:24					PWPBASE[11:8]			
		23:16	PWPBASE[7:0]							
		15:8	PWPEN	PWPLOCK	PWPMIR		PWPSIZE[11:8]			
		7:0	PWPSIZE[7:0]							
0x5C	PWP4	31:24					PWPBASE[11:8]			
		23:16	PWPBASE[7:0]							
		15:8	PWPEN	PWPLOCK	PWPMIR		PWPSIZE[11:8]			
		7:0	PWPSIZE[7:0]							
0x60	PWP5	31:24					PWPBASE[11:8]			
		23:16	PWPBASE[7:0]							
		15:8	PWPEN	PWPLOCK	PWPMIR		PWPSIZE[11:8]			
		7:0	PWPSIZE[7:0]							
0x64	PWP6	31:24					PWPBASE[11:8]			
		23:16	PWPBASE[7:0]							
		15:8	PWPEN	PWPLOCK	PWPMIR		PWPSIZE[11:8]			
		7:0	PWPSIZE[7:0]							
0x68	PWP7	31:24					PWPBASE[11:8]			
		23:16	PWPBASE[7:0]							
		15:8	PWPEN	PWPLOCK	PWPMIR		PWPSIZE[11:8]			
		7:0	PWPSIZE[7:0]							
0x6C	LBWP	31:24	LBWPLOCK							
		23:16								
		15:8	LBWP15	LBWP14	LBWP13	LBWP12	LBWP11	LBWP10	LBWP9	LBWP8
		7:0	LBWP7	LBWP6	LBWP5	LBWP4	LBWP3	LBWP2	LBWP1	LBWP0
0x70	UBWP	31:24	UBWPLOCK							
		23:16								
		15:8	UBWP15	UBWP14	UBWP13	UBWP12	UBWP11	UBWP10	UBWP9	UBWP8
		7:0	UBWP7	UBWP6	UBWP5	UBWP4	UBWP3	UBWP2	UBWP1	UBWP0

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x74	UOWP	31:24					UO2WPRLOCK3	UO2WPRLOCK2	UO2WPRLOCK1	UO2WPRLOCK0
		23:16					UO1WPRLOCK3	UO1WPRLOCK2	UO1WPRLOCK1	UO1WPRLOCK0
		15:8					UO2WP3	UO2WP2	UO2WP1	UO2WP0
		7:0					UO1WP3	UO1WP2	UO1WP1	UO1WP0
0x78	CWP	31:24						BC2WPLOCK		UC2WPLOCK
		23:16						BC1WPLOCK		UC1WPLOCK
		15:8						BC2WP		UC2WP
		7:0						BC1WP		UC1WP

31.2.23.1. CTRLA - NVM Write Control Register

Name: CTRLA
Offset: 0x0000
Reset: 0x00000000
Property: PAC Write Protection, KEY.KEY[7:0] = WRKEY to unlock when unprotected

Table 31-4. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

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

Bit 7 – PREPG NVM Pre-Program Configuration Bit

Notes:
1. This field can only be modified when PAC Write protection is disabled, STATUS.BUSY = 0, and KEY = <WRKEY Value> - same as NVMOP below.
2. It is recommended to either always use or never use PREPG.

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

Bits 3:0 – NVMOP[3:0] NVM Operation
1111-1011 = Reserved
1010 = SDAL: Set DAL per SDALCPUn - must be same or lower value than existing DAL value.(1)
Operation is reserved if system debug access level is elevated for the CPU.
For all Write/Erase Operations below, the entire target range must not be write protected.
0111 = PFM Erase Operation: Upper & Lower PFM Erase
0110 = Upper PFM Erase Operation:
0101 = Lower PFM Erase Operation:
0100 = Page Erase Operation: Erases page selected by ADDR
0011 = Row Write Operation: Programs row selected by ADDR

0010 = Quad (DWord) Program Operation: Programs flash word selected by ADDR
0001 = Single (DWord) Program Operation: Programs word selected by ADDR (2)
0000= No Operation

Notes:

1. This field can only be updated (and execute the operation) when PAC Write protection is disabled, STATUS.BUSY = 0, and KEY = <WRKEY>.
2. If FCR.ECCCTRL.ECCCTL[1:0] = 2'b00, this operation performs a No-Op but does not affect WRERR or RSTERR.

31.2.23.2. Control B

Name: CTRLB
Offset: 0x0004
Reset: 0x00000007
Property: PAC Write Protection, KEY.KEY[7:0] = CFGKEY to unlock when unprotected

Table 31-5. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
							SDALCPU0[1:0]	
Access							R/W	R/W
Reset							1	1

Bits 1:0 – SDALCPU0[1:0] Set DAL for CPU 0 (M33 Core)
Note: This field can only be updated when PAC Write protection is disabled, STATUS.BUSY=0, and KEY=<CFGKEY>.

Value	Description
11	Do Not Change Setting.
10	DAL2: Access to all memory. Can debug main CPU code.
01	DAL1: Reserved by convention (evaluates to DAL0).
00	DAL0: Access to very limited features.

31.2.23.3. NVM Interrupt Enable Clear Register

Name: INTENCLR
Offset: 0x000C
Reset: 0x00000000
Property: PAC Write-Protection

Table 31-6. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
			WRERR	RSTERR				
Access			R/W	R/W				
Reset			0	0				

Bit	7	6	5	4	3	2	1	0
	SECERR	OPERR	WPERR	BUSERR	FIFOERR	CFGERR	KEYERR	DONE
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 13 – WRERR Write Error Interrupt Disable Bit
Writing a '0' to this bit has no effect.
Writing a 1 to this bit will disable the Write Error as an interrupt request.
Reading this bit returns whether this interrupt is enabled (=1 ≥ enabled).
Note: This field can only be modified when STATUS.BUSY=0.

Bit 12 – RSTERR Reset or Brown Out Detect Error Interrupt Disable Bit
Writing a '0' to this bit has no effect.
Writing a 1 to this bit will disable the Reset or Brown Out Detect Error as an interrupt request.
Reading this bit returns whether this interrupt is enabled (=1 ≥ enabled).
Note: This field can only be modified when STATUS.BUSY=0.

Bit 7 – SECERR Security Violation Error Interrupt Disable Bit
Writing a '0' to this bit has no effect.
Writing a 1 to this bit will disable the Security Violation Error as an interrupt request.
Reading this bit returns whether this interrupt is enabled (=1 ≥ enabled).
Note: This field can only be modified when STATUS.BUSY=0.

Bit 6 – OPERR NVMOP Error Interrupt Disable Bit
Writing a '0' to this bit has no effect.

Writing a 1 to this bit will disable the NVMOP Error as an interrupt request.
Reading this bit returns whether this interrupt is enabled (=1 ≥ enabled).
Note: This field can only be modified when STATUS.BUSY=0.

Bit 5 – WPERR Write Protection Error Interrupt Disable Bit
Writing a '0' to this bit has no effect.
Writing a 1 to this bit will disable the Write Protection Error as an interrupt request.
Reading this bit returns whether this interrupt is enabled (=1 ≥ enabled).
Note: This field can only be modified when STATUS.BUSY=0.

Bit 4 – BUSERR AHB Bus Error During Row Write Interrupt Disable Bit
Writing a '0' to this bit has no effect.
Writing a 1 to this bit will disable the AHB Bus Error During Row Write as an interrupt request.
Reading this bit returns whether this interrupt is enabled (=1 ≥ enabled).
Note: This field can only be modified when STATUS.BUSY=0.

Bit 3 – FIFOERR FIFO Underrun During Row Write Interrupt Disable Bit
Writing a '0' to this bit has no effect.
Writing a 1 to this bit will disable the FIFO Underrun During Row Write as an interrupt request.
Reading this bit returns whether this interrupt is enabled (=1 ≥ enabled).
Note: This field can only be modified when STATUS.BUSY=0.

Bit 2 – CFGERR Configuration Error Interrupt Disable Bit
Writing a '0' to this bit has no effect.
Writing a 1 to this bit will disable the Configuration Error as an interrupt request.
Reading this bit returns whether this interrupt is enabled (=1 ≥ enabled).
Note: This field can only be modified when STATUS.BUSY=0.

Bit 1 – KEYERR Key Error Interrupt Disable Bit
Writing a '0' to this bit has no effect.
Writing a 1 to this bit will disable the Key Error as an interrupt request.
Reading this bit returns whether this interrupt is enabled (=1 ≥ enabled).
Note: This field can only be modified when STATUS.BUSY=0.

Bit 0 – DONE NVM Operation Done Interrupt Disable Bit
Writing a '0' to this bit has no effect.
Writing a 1 to this bit will disable the NVM Operation Done as an interrupt request.
Reading this bit returns whether this interrupt is enabled (=1 ≥ enabled).
Note: This field can only be modified when STATUS.BUSY=0.

31.2.23.4. NVM Interrupt Enable Set Register

Name: INTENSET
Offset: 0x0010
Reset: 0x00000000
Property: PAC Write-Protection

Table 31-7. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
			WRERR	RSTERR				
Access			R/W	R/W				
Reset			0	0				

Bit	7	6	5	4	3	2	1	0
	SECERR	OPERR	WPERR	BUSERR	FIFOERR	CFGERR	KEYERR	DONE
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 13 – WRERR Write Error Interrupt Enable Bit
Writing a '0' to this bit has no effect.
Writing a 1 to this bit will enable the Write Error as an interrupt request.
Reading this bit returns whether this interrupt is enabled (=1 ≥ enabled).
Note: This field can only be modified when STATUS.BUSY=0.

Bit 12 – RSTERR Reset or Brown Out Detect Error Interrupt Enable Bit
Writing a '0' to this bit has no effect.
Writing a 1 to this bit will enable the Reset or Brown Out Detect Error as an interrupt request.
Reading this bit returns whether this interrupt is enabled (=1 ≥ enabled).
Note: This field can only be modified when STATUS.BUSY=0.

Bit 7 – SECERR Security Violation Error Interrupt Enable Bit
Writing a '0' to this bit has no effect.
Writing a 1 to this bit will enable the Security Violation Error as an interrupt request.
Reading this bit returns whether this interrupt is enabled (=1 ≥ enabled).
Note: This field can only be modified when STATUS.BUSY=0.

Bit 6 – OPERR NVMOP Error Interrupt Enable Bit
Writing a '0' to this bit has no effect.

Writing a 1 to this bit will enable the NVMOP Error as an interrupt request.
Reading this bit returns whether this interrupt is enabled (=1 ≥ enabled).
Note: This field can only be modified when STATUS.BUSY=0.

Bit 5 – WPERR Write Protection Error Interrupt Enable Bit
Writing a '0' to this bit has no effect.
Writing a 1 to this bit will enable the Write Protection Error as an interrupt request.
Reading this bit returns whether this interrupt is enabled (=1 ≥ enabled).
Note: This field can only be modified when STATUS.BUSY=0.

Bit 4 – BUSERR AHB Bus Error During Row Write Interrupt Enable Bit
Writing a '0' to this bit has no effect.
Writing a 1 to this bit will enable the AHB Bus Error During Row Write as an interrupt request.
Reading this bit returns whether this interrupt is enabled (=1 ≥ enabled).
Note: This field can only be modified when STATUS.BUSY=0.

Bit 3 – FIFOERR FIFO Underrun During Row Write Interrupt Enable Bit
Writing a '0' to this bit has no effect.
Writing a 1 to this bit will enable the FIFO Underrun During Row Write as an interrupt request.
Reading this bit returns whether this interrupt is enabled (=1 ≥ enabled).
Note: This field can only be modified when STATUS.BUSY=0.

Bit 2 – CFGERR Configuration Error Interrupt Enable Bit
Writing a '0' to this bit has no effect.
Writing a 1 to this bit will enable the Configuration Error as an interrupt request.
Reading this bit returns whether this interrupt is enabled (=1 ≥ enabled).
Note: This field can only be modified when STATUS.BUSY=0.

Bit 1 – KEYERR Key Error Interrupt Enable Bit
Writing a '0' to this bit has no effect.
Writing a 1 to this bit will enable the Key Error as an interrupt request.
Reading this bit returns whether this interrupt is enabled (=1 ≥ enabled).
Note: This field can only be modified when STATUS.BUSY=0.

Bit 0 – DONE NVM Operation Done Interrupt Enable Bit
Writing a '0' to this bit has no effect.
Writing a 1 to this bit will enable the NVM Operation Done as an interrupt request.
Reading this bit returns whether this interrupt is enabled (=1 ≥ enabled).
Note: This field can only be modified when STATUS.BUSY=0.

31.2.23.5. NVM Interrupt Flag Register

Name: INTFLAG
Offset: 0x0014
Reset: 0x00000000
Property: PAC Write-Protection

Notes:
1. The interrupt flag bits of this register are set by hardware only.
2. Interrupt flags must be cleared and then read back to confirm the clear before exiting the ISR to avoid double interrupts.

Table 31-8. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
			WRERR	RSTERR				
Access			R/W	R/W				
Reset			0	0				

Bit	7	6	5	4	3	2	1	0
	SECERR	OPERR	WPERR	BUSERR	FIFOERR	CFGERR	KEYERR	DONE
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 13 – WRERR Write Error Flag Bit
Writing a '0' to this bit has no effect.
Writing a 1 to this bit will clear the flag.
Note: While this bit and its associated interrupt enable bit are 1, the NVM interrupt remains asserted.

Note: This field can only be modified when STATUS.BUSY=0.

Value	Description
0	The Write/Erase sequence completed normally
1	The Write/Erase sequence did not complete successfully

Bit 12 – RSTERR Reset or Brown Out Detect Error Flag Bit
The error is only captured during Write/Erase operations. Check the system RCAUSE register to see if this error was caused by a BOR event.
Writing a '0' to this bit has no effect.

Writing a 1 to this bit will clear the flag.
Note: While this bit and its associated interrupt enable bit are 1, the NVM interrupt remains asserted.

Note: This field can only be modified when STATUS.BUSY=0.

Value	Description
0	No Reset and Voltage level OK during write/erase
1	A reset or Low Voltage Detected (possible data corruption, verify data)

Bit 7 – SECERR Security Violation Error Bit
Attempted operation violates security configuration.
Writing a '0' to this bit has no effect.
Writing a 1 to this bit will clear the flag.
Note: While this bit and its associated interrupt enable bit are 1, the NVM interrupt remains asserted.

Note: This field can only be modified when STATUS.BUSY=0.

Value	Description
0	No Security Violation Error
1	Security Violation Error

Bit 6 – OPERR NVMOP Error Flag Bit
Writing a '0' to this bit has no effect.
Writing a 1 to this bit will clear the flag.
Note: While this bit and its associated interrupt enable bit are 1, the NVM interrupt remains asserted.

Note: This field can only be modified when STATUS.BUSY=0.

Value	Description
0	No NVMOP Error
1	Selected Operation is Disabled Error

Bit 5 – WPERR Write Protection Error Flag Bit
Writing a '0' to this bit has no effect.
Writing a 1 to this bit will clear the flag.
Note: While this bit and its associated interrupt enable bit are 1, the NVM interrupt remains asserted.

Note: This field can only be modified when STATUS.BUSY=0.

Value	Description
0	No Write Protection Error
1	Write Protection Error

Bit 4 – BUSERR AHB Bus Error During Row Write Flag Bit
Writing a '0' to this bit has no effect.
Writing a 1 to this bit will clear the flag.
Note: While this bit and its associated interrupt enable bit are 1, the NVM interrupt remains asserted.

Note: This field can only be modified when STATUS.BUSY=0.

Value	Description
0	No Bus Error
1	Bus Error

Bit 3 – FIFOERR FIFO Underrun During Row Write Flag Bit
Writing a '0' to this bit has no effect.
Writing a 1 to this bit will clear the flag.
Note: While this bit and its associated interrupt enable bit are 1, the NVM interrupt remains asserted.

Note: This field can only be modified when STATUS.BUSY=0.

Value	Description
0	No FIFO Error
1	FIFO Error

Bit 2 – CFGERR Configuration Error Flag Bit
Attempted Write/Erase when disallowed by a configuration setting.
Writing a '0' to this bit has no effect.
Writing a 1 to this bit will clear the flag.
Note: While this bit and its associated interrupt enable bit are 1, the NVM interrupt remains asserted.

Note: This field can only be modified when STATUS.BUSY=0.

Value	Description
0	No CFG Error
1	CFG Error

Bit 1 – KEYERR Key Error Flag Bit
Attempted to write to an SFR bit without first enabling it via KEY.
Writing a '0' to this bit has no effect.
Writing a 1 to this bit will clear the flag.
Note: While this bit and its associated interrupt enable bit are 1, the NVM interrupt remains asserted.

Note: This field can only be modified when STATUS.BUSY=0.

Value	Description
0	No Key Error
1	Key Error

Bit 0 – DONE NVM Operation Done Flag Bit
When NVMOP completes the FSM clears BUSY and sets Done.
Writing a '0' to this bit has no effect.
Writing a 1 to this bit will clear the flag.
Note: While this bit and its associated interrupt enable bit are 1, the NVM interrupt remains asserted.

Note: This field can only be modified when STATUS.BUSY=0.

Value	Description
0	NVMOP Not Done
1	NVMOP Done

31.2.23.6. NVM Status Register

Name: STATUS
Offset: 0x0018
Reset: 0x00000000
Property: –

Table 31-9. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

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

Bit 0 – BUSY NVM Busy Status
Note: This bit is read-only. It is set and cleared by hardware.

Value	Description
0	NVM Not Busy.
1	NVM Busy - SFR bits are not writable.

31.2.23.7. Key - SFR Unlock Register

Name: KEY
Offset: 0x001C
Reset: 0x00000000
Property: PAC Write-Protection

Table 31-10. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	KEY[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
	KEY[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
	KEY[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
	KEY[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 – KEY[31:0] NVM SFR Key Bits

This register is used to lock and unlock write access to other FCW SFR bits.
The KEY state remains active until it is written to a Valid Lock Code or CTRLA is written.

KEY[31:8] - KEY Code

0x91C32C= Valid Unlock code - Unlocks selected SFR(s)
All Others = Valid Lock Code - Locks all SFR protected by KEY register

KEY[7:0] KEY Value.

0x00= no selection, invalid selection, or invalid Key - Locks all SFR protected by KEY register
0x01= WRKEY Value: Unlock SFR bits associated with Write/Erase
0x02= SWAPKEY Value: Unlock SFR bits associated with Panel Swapping
0x04= CFGKEY Value: Unlock SFR bits associated with general Flash configuration
all other values: Invalid selection - Key Value set to 00 - Locks all SFR protected by KEY register

Notes:

1. KEY can only be written when STATUS.BUSY=0.
2. KEY resets to 0 (a valid lock code) on a successful write to CTRLA, SWAP or an unsuccessful write to any register (any write that sets KEYERR).
3. KEY allows for multiple writes to configuration registers when using CFGKEY. To lock, write a Valid Lock Code to KEY.

31.2.23.8. Flash Address Register

Name: ADDR
Offset: 0x0020
Reset: 0x00000000
Property: PAC Write-Protection

Table 31-11. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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

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

Bit	15	14	13	12	11	10	9	8
	ADDR[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

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

Bits 31:0 – ADDR[31:0] Flash Address used by CTRLA.NVMOP

This is a system byte address that the FCW aligns (by dropping lower ordered bits) to the minimum resolution of the NVMOP.

<u>Bulk/Chip/PFM Erase:</u>
Address is ignored
<u>Page Erase:</u>
Address identifies the page to erase
<u>Row Write:</u>
Address identifies the row to write
<u>Single Write:</u> (64-bits)
Address identifies the DWord to write. ADDR[2:0] are ignored
<u>Quad Write:</u> (256-bits)
Address identifies the Quad DWord to write. ADDR[4:0] are ignored

Notes:

1. This field can only be modified when STATUS.BUSY=0.
2. For 32-bit aligned memory the bottom two bits, ADDR[1:0], are ignored.
   For 64-bit (double word) writes the bottom four bits, ADDR[3:0], are ignored.

31.2.23.9. Source Data Address Register

Name: SRCADDR
Offset: 0x0024
Reset: 0x00000000
Property: PAC Write-Protection

Table 31-12. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	SRCADDR[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
	SRCADDR[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
	SRCADDR[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
	SRCADDR[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 – SRCADDR[31:0] Source Data (Word) Address
This is the system physical address of the data to be programmed into the flash when CTRLA.NVMOP is set to Row Write.
Notes:
1. This field can only be modified when STATUS.BUSY=0.
2. For 32-bit aligned memory the bottom two bits, SRCADDR[1:0], are ignored.
   For 64-bit (double word) writes the bottom four bits, SRCADDR[3:0], are ignored.

31.2.23.10. Flash Write Data Register

Name: DATAx
Offset: 0x28 + x*0x04 [x=0..7]
Reset: 0x00000000
Property: PAC Write-Protection

Table 31-13. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	DATA31	DATA30	DATA29	DATA28	DATA27	DATA26	DATA25	DATA24
Access	R/W	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
	DATA23	DATA22	DATA21	DATA20	DATA19	DATA18	DATA17	DATA16
Access	R/W	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
	DATA15	DATA14	DATA13	DATA12	DATA11	DATA10	DATA9	DATA8
Access	R/W	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
	DATA7	DATA6	DATA5	DATA4	DATA3	DATA2	DATA1	DATA0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – DATAx Flash Write Data

The value in this register(s) is written to flash when a Write operation is commanded.
Single Write: (64-bit data)
Writes DATA0 to ADDR[31:3] with address bits [2:0] = 000
Writes DATA1 to ADDR[31:3] with address bits [2:0] = 100
Quad Write: (256-bit data)
Writes DATA0 to ADDR[31:5], with address bits[4:0] = 0_0000
Writes DATA1 to ADDR[31:5], with address bits[4:0] = 0_0100
Writes DATA2 to ADDR[31:5], with address bits[4:0] = 0_1000
Writes DATA3 to ADDR[31:5], with address bits[4:0] = 0_1100
Writes DATA4 to ADDR[31:5], with address bits[4:0] = 1_0000
Writes DATA5 to ADDR[31:5], with address bits[4:0] = 1_0100
Writes DATA6 to ADDR[31:5], with address bits[4:0] = 1_1000
Writes DATA7 to ADDR[31:5], with address bits[4:0] = 1_1100
Note: This field can only be modified when STATUS.BUSY=0.

31.2.23.11. NVM Panel Swap Register

Name: SWAP
Offset: 0x0048
Reset: 0x00000000
Property: PAC Write Protection, KEY.KEY[7:0] = SWAPKEY to unlock when unprotected

Table 31-14. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
							PFSLOCK	PFSWAP
Access							R/W	R/W
Reset							0	0

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

Bit 9 – PFSLOCK PFM Swap Lock Bit

Writing a 0 to this bit has no effect.
Writing a 1 to this bit will set the lock bit.

Notes:

1. Once set, PFSLOCK can only be cleared by a reset.
2. This field can only be modified when STATUS.BUSY=0, PFSLOCK=0, and KEY.KEY=<SWAPKEY Value>.

Value	Description
1	PFSLOCK and PFSWAP cannot be written
0	PFSLOCK and PFSWAP can be written

Bit 8 – PFSWAP PFM Swap Status/Control Bit

Note: This field can only be modified when STATUS.BUSY=0, PFSLOCK=0, and KEY.KEY=<SWAPKEY Value>.

Value	Description
1	Panel 2 PFM is mapped into the Lower PFM region
0	Panel 1 PFM is mapped into the Lower PFM region

Bit 1 – BFSLOCK BFM Swap Lock Bit

Writing a 0 to this bit has no effect.
Writing a 1 to this bit will set the lock bit.

Notes:

1. Once set, BFSLOCK can only be cleared by a reset.
2. This field can only be modified when STATUS.BUSY=0, BFSLOCK=0, and KEY.KEY=<SWAPKEY Value>.

Value	Description
1	BFSLOCK and BFSWAP cannot be written
0	BFSLOCK and BFSWAP can be written

Bit 0 – BFSWAP BFM Swap Status/Control Bit

Notes:

1. The BFSWAP value is determined by the values the user programmed SEQNUM in each boot panel.
2. This field can only be modified when STATUS.BUSY=0, BFSLOCK=0, and KEY.KEY=<SWAPKEY Value>.

Value	Description
1	Panel 2 Boot is mapped into the Lower Boot region
0	Panel 1 Boot is mapped into the Lower Boot region

31.2.23.12. PFM Write Protect Region x Register

Name: PWPx
Offset: 0x4C + x*0x04 [x=0..7]
Reset: 0x00000000
Property: PAC Write Protection, KEY.KEY[7:0] = CFGKEY to unlock when unprotected

Table 31-15. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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

Bit	23	22	21	20	19	18	17	16
	PWPBASE[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
	PWPEN	PWPLOCK	PWPMIR		PWPSIZE[11:8]			
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
	PWPSIZE[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 27:16 – PWPBASE[11:0] PFM Write Protect Region x Base Address - 4KB Page Aligned
The Region Base Address is the concatenation of PWPBASE[23:12] and 0x000.
When PWPEN=1, the region from PWPBASE to PWPBASE+PWPSIZE+1 is write/erase protected.
Note: This is a byte address force to align to page boundaries.

Note: This field can only be modified when PWPLOCK=0, STATUS.BUSY=0 and KEY.KEY=<CFGKEY Value>.

Bit 15 – PWPEN PFM Write Protect Region x Enable Bit
Note: This field can only be modified when PWPLOCK=0, STATUS.BUSY=0 and KEY.KEY=<CFGKEY Value>.

Value	Description
1	Region is Enabled
0	Region is Not Enabled

Bit 14 – PWPLOCK PWP Region x Lock Bit
Note: PWPLOCK can be set the same time as PWPBASE and PWPSIZE are written. Once set, PWPLOCK can only be cleared by a reset (Writing zero has no effect.).

Note: This field can only be modified when PWPLOCK=0, STATUS.BUSY=0 and KEY.KEY=<CFGKEY Value>.

Value	Description
1	PWPx register is Locked and cannot be modified
0	PWPx register is Not Locked and can be modified

Bit 13 – PWPMIR Mirror PWP Bit

Mirrors Lower PFM settings to Upper or Upper PFM settings to Lower. This feature can be used to maintain Write Protect (WP) consistency between Upper and Lower PFM when using PFSWAP.
Note: When Mirrored, the PWPBASE address bit that distinguishes between panels is treated as a “Don’t Care”, meaning it can be 0 or 1.

Note: This field can only be modified when PWPLOCK=0, STATUS.BUSY=0 and KEY.KEY=<CFGKEY Value>.

Value	Description
1	PWP settings are Mirrored
0	PWP settings are NOT Mirrored

Bits 11:0 – PWPSIZE[11:0] PWP Region x Size in 4KB

Pages Region Size is (PWPSIZE+1) *4KB.
`0x000`= 4KB
`0x001`= 8KB
...
`0x3FF`= 16MB
Note: This field can only be modified when PWPLOCK=0, STATUS.BUSY=0 and KEY.KEY=<CFGKEY Value>.

31.2.23.13. Lower BFM Write Protect Register

Name: LBWP
Offset: 0x006C
Reset: 0x0000FFFF
Property: PAC Write Protection, KEY.KEY[7:0] = CFGKEY to unlock when unprotected

Table 31-16. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	LBWPLOCK							
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
	LBWP15	LBWP14	LBWP13	LBWP12	LBWP11	LBWP10	LBWP9	LBWP8
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
	LBWP7	LBWP6	LBWP5	LBWP4	LBWP3	LBWP2	LBWP1	LBWP0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bit 31 – LBWPLOCK LBWP Lock Bit

Notes:

1. This field can be set at the same time as writing to LBWP. Once set, LBWPLOCK can only be cleared by a reset (Writing zero has no effect.).
2. This field can only be modified when LBWPLOCK=0, STATUS.BUSY=0 and KEY.KEY=<CFGKEY Value>.

Value	Description
1	LBWP & LBWPLOCK fields are Locked and cannot be modified
0	LBWP & LBWPLOCK fields are NOT Locked and can be modified

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 – LBWP Lower Boot Pages Write Protect Bits

Note:

1. This field can only be modified when LBWPLOCK=0, STATUS.BUSY=0 and KEY.KEY=<CFGKEY Value>.

Value	Description
1	Erase and Write Protection for Lower Boot Page *n* is Enabled
0	Erase and Write Protection for Lower Boot Page *n* is Disabled

31.2.23.14. Upper BFM Write Protect Register

Name: UBWP
Offset: 0x0070
Reset: 0x0000FFFF
Property: PAC Write Protection, KEY.KEY[7:0] = CFGKEY to unlock when unprotected

Table 31-17. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	UBWPLOCK							
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
	UBWP15	UBWP14	UBWP13	UBWP12	UBWP11	UBWP10	UBWP9	UBWP8
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
	UBWP7	UBWP6	UBWP5	UBWP4	UBWP3	UBWP2	UBWP1	UBWP0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bit 31 – UBWPLOCK UBWP Lock Bit

Notes:

1. This field can be set at the same time as writing to UBWP. Once set, UBWPLOCK can only be cleared by a reset (Writing zero has no effect.).
2. This field can only be modified when UBWPLOCK=0, STATUS.BUSY=0 and KEY.KEY=<CFGKEY Value>.

Value	Description
1	UBWP & UBWPLOCK fields are Locked and cannot be modified
0	UBWP & UBWPLOCK fields are NOT Locked and can be modified

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 – UBWP Upper Boot Pages Write Protect Bits

Note:

1. This field can only be modified when UBWPLOCK=0, STATUS.BUSY=0 and KEY.KEY=<CFGKEY Value>.

Value	Description
1	Erase and Write Protection for Upper Boot Page *n* is Enabled
0	Erase and Write Protection for Upper Boot Page *n* is Disabled

31.2.23.15. User OTP Write Protect Register

Name: UOWP
Offset: 0x0074
Reset: 0x00000F0F
Property: PAC Write Protection, KEY.KEY[7:0] = CFGKEY to unlock when unprotected

Table 31-18. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
					UO2WPRLOCK3	UO2WPRLOCK2	UO2WPRLOCK1	UO2WPRLOCK0
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	23	22	21	20	19	18	17	16
					UO1WPRLOCK3	UO1WPRLOCK2	UO1WPRLOCK1	UO1WPRLOCK0
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
					UO2WP3	UO2WP2	UO2WP1	UO2WP0
Access					R/W	R/W	R/W	R/W
Reset					1	1	1	1

Bit	7	6	5	4	3	2	1	0
					UO1WP3	UO1WP2	UO1WP1	UO1WP0
Access					R/W	R/W	R/W	R/W
Reset					1	1	1	1

Bits 24, 25, 26, 27 – UO2WPRLOCK User OTP Page 2 Write Protect Row Lock Bit
Writing a 0 to a bit has no effect.
Writing a 1 to a bit will enable the corresponding lock.
Notes:
1. This field can be set at the same time as writing to UO2WPR[n]. Once set, UO2WPRLOCK can only be cleared by a reset.
2. This field can only be modified when UO2WPRLOCK[n]=0, STATUS.BUSY=0,and KEY.KEY=<CFGKEY Value>.

Value	Description
1	UO2WPR[*n*] & UO2WPRLOCK[*n*] bits are Locked and cannot be modified
0	UO2WPR[*n*] & UO2WPRLOCK[*n*] bits are NOT Locked and can be modified

Bits 16, 17, 18, 19 – UO1WPRLOCK User OTP Page 1 Write Protect Row Lock Bit
Writing a 0 to a bit has no effect.
Writing a 1 to a bit will enable the corresponding lock.

Notes:

1. This field can be set at the same time as writing to UO1WPR[n]. Once set, UO1WPRLOCK can only be cleared by a reset.
2. This field can only be modified when UO1WPRLOCK[n]=0, STATUS.BUSY=0,and KEY.KEY=<CFGKEY Value>.

Value	Description
1	UO1WPR[*n*] & UO1WPRLOCK[*n*] bits are Locked and cannot be modified
0	UO1WPR[*n*] & UO1WPRLOCK[*n*] bits are NOT Locked and can be modified

Bits 8, 9, 10, 11 – UO2WP User OTP Page 2 Write Protect Row Bit

Writing a 0 to a bit has no effect.
Writing a 1 to a bit will enable the corresponding write protect.

Notes:

1. Each bit protects 1/4 of the page regardless of the number of actual rows in a page.
2. This field can only be modified when UO2WPRLOCK[n]=0, STATUS.BUSY=0, and KEY.KEY=<CFGKEY Value>.

Value	Description
1	Write Protection for User OTP Page 2 Row *n* is Enabled
0	Write Protection for User OTP Page 2 Row *n* is Disabled

Bits 0, 1, 2, 3 – UO1WP User OTP Page 1 Write Protect Row Bit

Writing a 0 to a bit has no effect.
Writing a 1 to a bit will enable the corresponding write protect.

Notes:

1. Each bit protects 1/4 of the page regardless of the number of actual rows in a page.
2. This field can only be modified when UO1WPRLOCK[n]=0, STATUS.BUSY=0, and KEY.KEY=<CFGKEY Value>.

Value	Description
1	Write Protection for User OTP Page 1 Row *n* is Enabled
0	Write Protection for User OTP Page 1 Row *n* is Disabled

31.2.23.16. CFM Page Write Protect Register

Name: CWP
Offset: 0x0078
Reset: 0x00000505
Property: PAC Write Protection, KEY.KEY[7:0] = CFGKEY to unlock when unprotected

Table 31-19. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
						BC2WPLOCK		UC2WPLOCK
Access						R/W		R/W
Reset						0		0

Bit	23	22	21	20	19	18	17	16
						BC1WPLOCK		UC1WPLOCK
Access						R/W		R/W
Reset						0		0

Bit	15	14	13	12	11	10	9	8
						BC2WP		UC2WP
Access						R/W		R/W
Reset						1		1

Bit	7	6	5	4	3	2	1	0
						BC1WP		UC1WP
Access						R/W		R/W
Reset						1		1

Bit 26 – BC2WPLOCK Boot Configuration Page 2 Write Protect Lock Bit

Writing a 0 to a bit has no effect.
Writing a 1 to a bit will enable the corresponding lock.
Notes:

1. This field can be set at the same time as writing to BC2WP.
2. This field can only be modified when BC2WPLOCK=0, STATUS.BUSY=0 and KEY.KEY=<CFGKEY Value>.

Value	Description
1	BC2WP & BC2WPLOCK fields are Locked and cannot be modified
0	BC2WP & BC2WPLOCK fields are NOT Locked and can be modified

Bit 24 – UC2WPLOCK User Configuration Page 2 Write Protect Lock Bit

Writing a 0 to a bit has no effect.
Writing a 1 to a bit will enable the corresponding lock.
Notes:

1. This field can be set at the same time as writing to UC2WP.
2. This field can only be modified when UC2WPLOCK=0, STATUS.BUSY=0 and KEY.KEY=<CFGKEY Value>.

Value	Description
1	UC2WP & UC2WPLOCK fields are Locked and cannot be modified
0	UC2WP & UC2WPLOCK fields are NOT Locked and can be modified

Bit 18 – BC1WPLOCK Boot Configuration Page 1 Write Protect Lock Bit
Writing a 0 to a bit has no effect.
Writing a 1 to a bit will enable the corresponding lock.
Notes:
1. This field can be set at the same time as writing to BC1WP.
2. This field can only be modified when BC1WPLOCK=0, STATUS.BUSY=0 and KEY.KEY=<CFGKEY Value>.

Value	Description
1	BC1WP & BC1WPLOCK fields are Locked and cannot be modified
0	BC1WP & BC1WPLOCK fields are NOT Locked and can be modified

Bit 16 – UC1WPLOCK User Configuration Page 1 Write Protect Lock Bit
Writing a 0 to a bit has no effect.
Writing a 1 to a bit will enable the corresponding lock.
Notes:
1. This field can be set at the same time as writing to UC1WP.
2. This field can only be modified when UC1WPLOCK=0, STATUS.BUSY=0 and KEY.KEY=<CFGKEY Value>.

Value	Description
1	UC1WP & UC1WPLOCK fields are Locked and cannot be modified
0	UC1WP & UC1WPLOCK fields are NOT Locked and can be modified

Bit 10 – BC2WP Boot Configuration Page 2 Write Protect Bit
Note:
1. This field can only be modified when BC2WPLOCK=0, STATUS.BUSY=0 and KEY.KEY=<CFGKEY Value>.

Value	Description
1	Erase and Write Protection for Boot Configuration Page 2 is Enabled
0	Erase and Write Protection for Boot Configuration Page 2 is Disabled

Bit 8 – UC2WP User Configuration Page 2 Write Protect Bit
Note:
1. This field can only be modified when UC2WPLOCK=0, STATUS.BUSY=0 and KEY.KEY=<CFGKEY Value>.

Value	Description
1	Erase and Write Protection for User Configuration Page 2 is Enabled
0	Erase and Write Protection for User Configuration Page 2 is Disabled

Bit 2 – BC1WP Boot Configuration Page 1 Write Protect Bit
Note:
1. This field can only be modified when BC1WPLOCK=0, STATUS.BUSY=0 and KEY.KEY=<CFGKEY Value>.

Value	Description
1	Erase and Write Protection for Boot Configuration Page 1 is Enabled
0	Erase and Write Protection for Boot Configuration Page 1 is Disabled

Bit 0 – UC1WP User Configuration Page 1 Write Protect Bit

Note:

1. This field can only be modified when UC1WPLOCK=0, STATUS.BUSY=0 and KEY.KEY=<CFGKEY Value>.

Value	Description
1	Erase and Write Protection for User Configuration Page 1 is Enabled
0	Erase and Write Protection for User Configuration Page 1 is Disabled

31.3. Flash Controller, Read

31.3.1. Overview

The Flash Controller, Read (FCR) contains one AHB system read target for the Flash, ECC calculation module, automatic CRC scanning of Flash, and Fault Injection. It also provides arbitration between the FCW and the system, ready timers for startup from POR/sleep, and multiple panel address decoding.

The FCR provides a user interfaces to Flash read control features. The FCR contains Flash wait state control, ECC fault reporting, ECC Fault Injection, and Flash CRC. The Fault and CRC features that can be used to meet certain functional safety requirements for invariant memories. For additional information, refer to the Definitions and Flash Architecture.

31.3.2. Features

The following are key features of the module:

- Supports flash partitions
  - Program Flash Memory (PFM)
  - Boot Flash Memory (BFM)
  - Configuration Flash Memory (CFM)
- Flash-Based Configuration Retention Support:
  - Read and update external module Calibration and Configuration data
  - Read and update Flash IP Calibration data
  - Read and update Page Redundancy Records (locally)
- Flash Memory Read Protection
  - Boot CFG BoCoR Read Protection (for CE Key privacy)
- AHB Reads of Flash Memory
  - One AHB Read port
  - Round Robin or Fixed Arbitration via SFR bit
  - One Read Buffer per port
- CRC Logic for In-variant memory tests
- Fault Logic for error injection and testing

31.3.3. Peripheral Dependencies

Peripheral Name	Base Address	NVIC IRQ Index:Source	MCLK AXI/APB Clocks Index:Name [1]	PAC Peripheral ID (PAC.WRCTRL), PAC.INTFLAGx[n]	Power Domain
FCR	0x4400_2000	0 : ECERR 1 : FAULT or CRCERR	4 : CLK_FCR_AHB 33: CLK_FCR_APB	1, INTFLAGA[1]	VDDREG

Note:
1. Register Field: MCLK.CLKMSK{index/32}.MASK[index mod 32].

31.3.4. Clocks

The FCR peripheral bus clock (CLK_FCR_APB) can be enabled and disabled in the Main Clock Controller.

The FCR data bus clock (CLK_FCR_AHB) can be enabled and disabled in the Main Clock Controller.

The FCR also requires an on-chip 8 MHz clock source that is automatically configured without application assistance. The CLK_GEN_FCW is derived by dividing the 48 MHz trimmed internal RC oscillator by 6. This clock is used by the Period Clock (PerCLK) to provide additional delay between the stop and start of the CRC engine.

31.3.5. Concurrent Access

31.3.5.1. Read While Write

In order to permit code execution from one Flash panel while writing (or erasing) another, the FCR detects and arbitrates access conflicts between the Flash Write Controller (FCW), System read requests, and internal read requests.

When a Flash panel can no longer be read because it is actively being written or erased, the FCR's Arbiter is signaled accordingly. Read accesses from the system bus are compared to the panel address ranges and allocated accordingly within the panel arbiter.

31.3.5.2. Concurrent Reads

The FCR's Arbiter permits system reads to access any panel, provided it is not being actively written or erased. The panel arbiter also permits concurrent read accesses to different panels.

If the system (CPU or other peripheral initiator) attempts to access a panel that is being actively written or erased, then the FCR's Arbiter delays the data ready response to the initiator until such time that the FCW has completed its sequence (i.e. the FCW always has priority).

31.3.6. Flash Address Map Swap

31.3.6.1. PFM Address Map Swap

The FCR handles PFM address map swapping for read operations based on the value of FCW.SWAP.PFSWAP in the FCW. See the SWAP Register for more information on PFSWAP.

31.3.6.2. BFM Address Map Swap

The FCR handles BFM address map swapping for read operations based on the value of FCW.SWAP.BFSWAP in the FCW. See the SWAP Register for more information on BFSWAP.

31.3.6.3. CFM Address Map Swap

The Configuration Flash Memory address map cannot be swapped.

31.3.7. Power Management

The Flash system supports multiple power mode for reducing or managing power. These include Auto-Standby, Flash power down, and controlling power during sleep.

31.3.7.1. Flash Auto Standby

The Flash panel enters a standby mode automatically after a read or write operation. Each Flash panel enters its Auto-Standby independently of the other. After a read request completes as defined by the read access time (T_{ACC}), the panel enters Auto-Standby. For writes/erases this after the recovery time. When a panel is in this mode read and write requests incur no access time penalty.

31.3.7.2. Sleep Interaction

The Flash system holds off entry into sleep modes lower than Idle Sleep to cleanly stop Flash operations. The following operations delay entry into sleep:

- In-progress read from any source (system, CRC, etc)
- Requested program or erase operation, etc.

Note: Row programming and erase operations can significantly delay entry into a sleep mode.

31.3.8. HSM Read Security

Please contact a local Microchip representative for details of how the FCR module supports HSM read security.

31.3.9. Flash Read Control

31.3.9.1. Read Control Overview

The FCR provides read access to the Flash system for the CPU and peripheral hosts. It implements one AHB client. This interface runs at the same frequency as the system bus and use the same Flash access timing control.

31.3.9.2. CTRLA SFR Description

The SFR fields that control read operations to the Flash can be changed at any time, provided the FCR (CTRLA specifically) is not write locked by the PAC module. New wait state values are used on the next Flash read. RDBUFWS changes on the next update to the associated read buffer. A new arbitration (ARB) value is used at the next available arbitration cycle on a per panel basis.

31.3.9.2.1. ARB

ARB controls read requester arbitration to the Flash. Requesters include each AHB interface (numbered 0 to 2) and the CRC Finite State Machine. The main CPU always uses AHB0 to access the Flash. FCW requests take precedence over read requests. Arbitration is per Flash panel.

When ARB = 0, round-robining arbitration is used for all read requests to the Flash. When ARB = 1, fixed arbitration is used for all reads to the Flash. In this mode, AHB0 has the highest priority followed by AHB1, etc., and CRC has lowest priority.

31.3.9.2.2. FWS[3:0] - Flash Wait States

The FWS[3:0] field controls the number of extra clock cycles it takes to access the Flash memory and provide data back (to the bus or internal requester). The number of wait states needed is typically Taws/(AHB Clock Period) rounded up to the nearest integer. If ECC is enabled an extra wait state may be needed. Each device characterizes and states the frequency range for each wait state selection and for ECC on/off. See *FLASH CHARACERIZATION*.

31.3.9.2.3. ADRWS - Address Wait States

The ADRWS field controls the number of extra clock cycles for address setup to the Flash memory. Address wait state occurs before requesting arbitration, so it is in between the bus and the request to Flash. In most system the path starts at the initiator of the request.

When ADRWS = 0, no extra cycle clocks are added to address setup. When ADRWS = 1, one extra clock is added to address setup. ADRWS only affects AHB interfaces. It does nor affect internal read requests such as from CRC logic if present.

31.3.9.2.4. AUTOWS - Auto Wait States

The AUTOWS field controls the use of automatic wait state for the Flash access time (Taws). This is useful when varying clock frequencies.

When AUTOWS=1, the total Flash wait states are ADRWS+Taws. When AUTOWS=0 the total Flash wait states are ADRWS+FWS. Non-CPU ports may incur an extra wait state. Note that the bus may add pipeline delays to either the address phase or the data phase.

31.3.9.2.5. RDBUFWS[3:0] - Read Buffer Wait States

The RDBUFWS field controls the determinism of accesses from the read buffer. Each bit corresponds to AHB Interface *n*, *n*= 0 to 2 When RDBUFWS[*n*] = 0, AHB interface *n* returns data from the read buffer in zero wait states. When RDBUFWS[*n*] = 1, AHB interface *n returns* data with wait states to match the Flash access of ADRWS+FWS such that the initiator sees a constant access time.

The setting of RDBUFWS is ignored when AUTOWS = 1. For this case, all data return from the read buffer return with zero wait states.

31.3.10. Cyclic Redundancy Check (CRC) Support

31.3.10.1. CRC Overview

The Flash system supports automated CRC of any contiguous Flash region. It supports both 16-bit and 32-bit CRC with a programmable polynomial, initial value, final XOR value, and checksum compare value. The CRC can run on-command or continuously with auto repeat.

All control registers must be setup prior to enabling CRC operation. Once setup, writing CRCCTRL.CRCEN = 1 loads the Initial Value into the Accumulator, the Message Length into the Message Length counter, and the Period into the Period counter. After the Period counter decrements to zero the logic reads the data at the Message Address. For each byte shifted through the LFSR, it decrements the byte counter.

Bytes shift into the LFSR lowest addressed to highest addressed. The Reflected Input bit, CRCCTRL.RIN, determines if it is MSbit first or LSbit first. During the shifting the Accumulator bits toggle showing the result. Once the state machine completely shifts in the last byte from the current Flash read, it reloads the Period counter and starts counting.

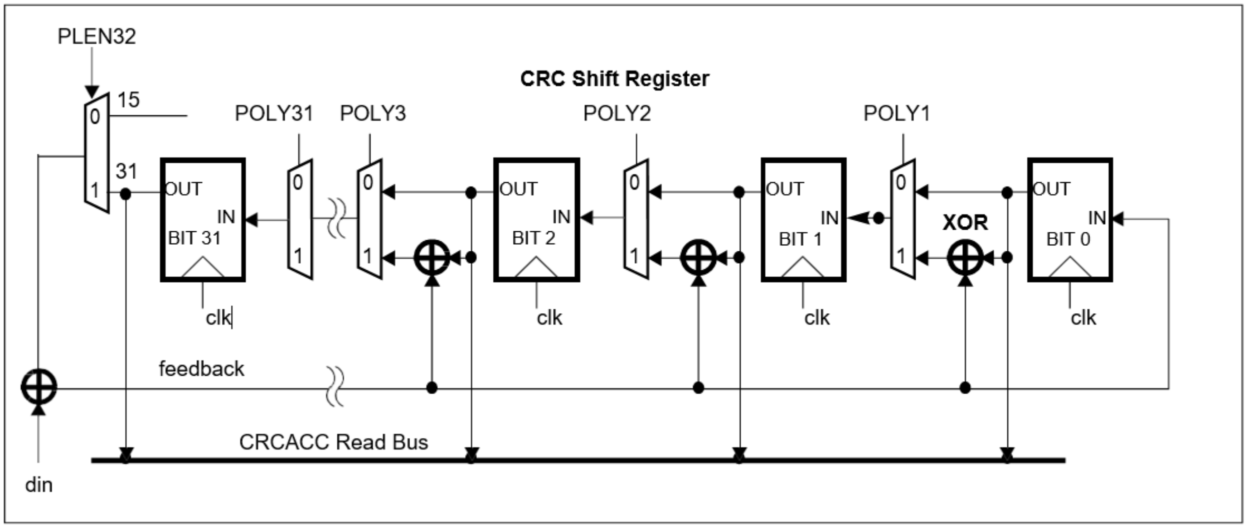
The CRC logic continues reading, shifting, and waiting until the Message Length counter reaches zero. The final step of the CRC operation XORs the Accumulator with the Final XOR value before comparison to the Check Sum value. Reflected Bit Order bit, CRCCTRL.REFOUT, determines if the Accumulator is reversed or not before the Final XOR.
Note: The Accumulator is not updated with the Final XOR value. If the comparison fails, logic sets the INTFLAG.CRCERR flag and stops with INTFLAG.CRCDONE flag also set.

If no error occurs and CRCCTRL.AUTOR is not set, the CRC logic sets INTFLAG.CRCDONE and INTFLAG.CRCERR remains cleared. If CRCCTRL.AUTOR is set, then the CRC logic reloads all initial settings are restarts the CRC calculation without setting INTFLAG.CRCDONE.

31.3.10.2. CRC Accumulator Topology

Other CRC modules have functions that support two selectable accumulator topologies. The topologies differ based on if PLEN+1 trailing zeros in the byte stream exist or not. Since this CRC is specific to Flash-based tests and not streaming data, FCR's CRC topology does not require "extra" trailing zeros in the message. This topology is shown in the following figure.

Figure 31-8. CRC LFSR Topology



31.3.10.3. Manual Usage Model

When the Auto Repeat bit selects manual mode (CRCCTRL.AUTOR=0), the last step of the CRC operation always sets the INTFLAG.CRCDONE bit. The state of INTFLAG.CRCERR reflects the validity of the message (Flash memory region).

If checking the same region (i.e. message) again, simply clear CRCEN, then CRCDONE and CRCERR before setting CRCEN again. The IV loads into the ACC and the operation begins again.

31.3.10.4. Auto Repeat Usage Model

When the Auto Repeat bit selects repeat mode (CRCCTRL.AUTOR=1), the CRC operation only sets INTFLAG.CRCDONE at the end of a loop if there is a comparison error. The error causes the logic to set both INTFLAG.CRCDONE and INTFLAG.CRCERR. To restart the CRC, simply clear CRCCTRL.CRCEN, then INTFLAG.CRCDONE and INTFLAG.CRCERR before setting CRCCTRL.CRCEN again.

If there is no comparison error the logic restarts the CRC operation by loading the initial value into ACC, the MLEN into the length counter, and Period into the Period counter. CRCDONE, CRCERR, and CRCEN remain unchanged. As mentioned above, each read waits the Period count including the first read of a loop. The CRC operation continues looping until it detects an error or is stopped manually.

31.3.10.5. CRC Interrupt Event

The INTFLAG.CRCDONE bit reflects the state of the interrupt event. When it is DONE the interrupt event is asserted. Clearing INTFLAG.CRCDONE de-asserts the event. The system level interrupt controller (NVIC) determines if the CPU see the event.

31.3.10.6. Performance

To process each byte of data (8-bits), the calculation takes one peripheral clock. However, it reads a multi-byte Flash word and operates on all those bytes each read. Since the Flash Read Word, or Flash Word in Bytes (FWB), is 32 bytes long it takes 32 clocks to calculate a CRC.

Since the CRC is reading the Flash it must also use the system Flash Wait States, FWS. The time to get the Flash data into the shift register is FWS+1. Also, if the system continues to access flash, the CRC logic is lowest priority and must wait until there is an idle cycle.

The logic periodically reads the flash to supply data to the calculation. The PERIOD is programmable from 0 to 4095 extra Period Clock (PerCLK) counts from the end of the calculation. (The PerCLK is an 8MHz fixed frequency clock derived from the 48 MHz Precision Internal RC Oscillator.) Though selecting a period of zero clocks will not completely cut off the system flash bandwidth, it could be severely limiting. System bandwidth can be less than half under such conditions. Low period selections are useful during light loads or scheduled checks. When running with auto repeat, an appropriate period value is one that finishes the calculation before to the maximum user allotted time.

Lastly, the CRC needs to know the Message Length, MLEN, in bytes over which to perform the calculation. Note that neither the starting address of the message nor the message length need to be aligned to Flash Read Bytes.

31.3.10.6.1. CRC Performance Equation

First, let:

T_{MCALC} = CRC message calculation time

T_{STEP} = CRC step time (time to calculate each step in the CRC)

T_{ACC} = Flash Access Time

T_{PER} = Period between CRC accesses

P_{APB} = Peripheral Bus Period = 1/Peripheral Bus speed

P_{AHB} = AHB Period = 1/AHB Clock speed

P_{PerCLK} = Period of Period Clock = 1/8 MHz = 125 ns

FWB = Flash Word in Bytes = 32

Then:

T_{MCALC} = (TSTEP + 2*TACC + T_{PER})*MLEN/FWB

Where:

T_{STEP} = FWB*P_{APB}

T_{ACC} = (FWS+1)*Pahb

T_{PER} = P_{PerCLK} * PERIOD + 2 * P_{AHB}, if PERIOD > 0

= 0, if PERIOD = 0.

The equation for T_{MCALC} includes the term 2*T_{ACC} instead of just T_{ACC} because the CPU is likely reading, forcing the CRC engine to wait before reading. The factor MLEN/FWB represents the number of accesses needed to read the message.

If T_{MCALC} is known, we can solve for PERIOD:

PERIOD = 8e6 * { (FWB/MLEN)*T_{MCALC} – [(FWB*P_{APB} + (2*FWS + 2)P_{AHB} + 2*P_{AHB}] }

31.3.10.6.2. CRC Performance Example

Assume the application must calculate the CRC on an 80 Kbyte message every 800 ms. Further, assume that both the AHB and APB are running at 100 MHz and that Flash needs four wait states. Then:

T_{MCALC} = 800 ms = 800,000,000 ns

MLEN = 80 Kbytes = 80*1024 = 81920 Bytes, so MLEN/FWB = 2560

P_{APB} = 1/100 MHz = 10 ns

P_{AHB} = 1/100 MHz = 10 ns

FWS = 4

Solving for the value of PERIOD,

PERIOD = 8e6 Hz* { (800,000,000 ns)/2560 – [32*10ns + (8 + 2)*10 ns + 2*10 ns] }

= 8e6 Hz * { 312,060 ns } = 2496.48

= 2496 (rounding down)

31.3.10.7. CRC Examples

The following table provides CRC configuration with a polynomial size of 32. For each term (xn) of the polynomial, a corresponding bit in CRCPOLY is set to ‘1’. This polynomial implements CRC-32-IEEE 802.3.

Table 31-20. Example 32-Bit Polynomial CRC Setup

Field Name	Bit Value	Description
CRCCTRL.PLEN	1	32-bit polynomial length
CRCPOLY	`0000 0100 1100 0001 0001 1101 1011 011-`	CRC polynomial: x32 + x26 + x23 + x22 + x16 + x12 + x11 + x10 + x8 + x7 + x5 + x4 + x2 + x +1

The following table provides CRC configuration with the polynomial size of 16. The value of CRCPOLY shows the 12th bit and the 5th bit set to ‘1’, as required by the equation. The 0th bit required by the equation is always XOR-ed.

Table 31-21. Example 16-Bit Polynomial CRC Setup

Field Name	Bit Value	Description
CRCCTRL.PLEN	0	16-bit polynomial length
CRCPOLY	`0000 0000 0000 0000 0001 0000 0010 000-`	CRC polynomial: x16 + x12 + x5 + 1

31.3.10.8. Effects of ECC on CRC

CRC reads of the Flash obey the selected ECC mode of the device. If ECC is enabled the CRC uses the corrected data. If ECC is not enabled (either bypass or simple parity) the CRC uses the uncorrected data. ECC errors, either SEC or DED, are not reported to the system for CRC reads. CRC reads of the Flash also do not affect the FLT features or SFR bits.

31.3.10.9. Wait States Used by CRC

CRC always uses system wait states. If the panel is placed in a mode that uses other wait states, then CRC must be disabled until returning to system wait states.

31.3.10.10. Effects of Programming on CRC

The CRC logic pauses automatically when a programming or erase operation prevents it from reading the Flash. No user intervention is needed.

31.3.10.11. Effects of DBGCTRL on CRC

When DBGCTRL.CRCRUN = 1 the CRC Logic continues operation during Debug Mode. When DBGCTRL.CRCRUN = 0 the CRC Logic halts operation during Debug Mode.

31.3.10.12. Effects of Sleep Mode on CRC

When CRCCTRL.RUNSTDBY = 1, the CRC Logic continues operation during Standby mode. Otherwise, CRC logic halts operations. When CRCCTRL.RUNSTDBY = 0 the CRC Logic halts operation during Standby mode. Unlike PAUSE, the counters also stop.

31.3.10.13. CRCCTRL SFR Description

31.3.10.13.1. CRCRST

CRCRST reset the CRC SFR bits and returns the FSM to its idle state. The FSM clears CRCRST before returning to idle.

31.3.10.13.2. CRCEN

CRCEN is essentially the “enable” and “go” bit. All other CRC related SFR bits and fields must be setup prior to writing CRCEN to one. When CRCEN = 1 all other CRC bits are write protected except CRCEN, CRCRST, and PAUSE.

31.3.10.13.3. PLEN32

The PLEN32 bit determines the Polynomial Length, either CRC-16 (when zero) or CRC-32 (when one). PLEN32 also defines the effective length of the bits in CRCIV, CRCACC, CRCPOLY, CRCFXOR, and CRCSUM.

Other CRC functions may have a fully programmable PLEN from 0 to 31. These values define CRC lengths from 1 to 32. So PLEN32=0 and PLEN32=1 are equivalent to PLEN=15 and PLEN=31, respectively, on a fully programmable CRC.

31.3.10.13.4. AUTOR

The AUTOR selects between Manual mode and Auto Repeat mode. Manual mode requires software to start the CRC engine for each check of the message allowing. Manual mode allows software to check several different messages. Auto Repeat mode does not require software to restart the CRC engine to check the message. However, this mode is limited to checking the same message repeatedly, only stopping on an error or by software disable of the CRC.

31.3.10.13.5. ROUT

The ROUT, Reflected Output, controls the bit order output of the CRC Accumulator. When ROUT = 0, the FSM XORS the CRCACC with CRCFXOR directly. When ROUT = 1, the FSM reverses the bit order of the CRCACC before it XORS it with CRCFXOR.

31.3.10.13.6. RIN

The RIN, Reflected Input, controls the shift direction of each byte. When RIN = 0, the FSM feeds the LSFR LSbit first. When RIN = 1, the MSbit is first into the LFSR. RIN does not swap bytes in a (32-bit) word or a Flash word, bytes flow from lowest addressed to highest addressed.

31.3.10.13.7. PERIOD[11:0]

The PerCLK is an 8MHz fixed frequency clock. The PERIOD field defines the number of PerCLK counts that the CRC FSM waits from the last bit shifted in to the LFSR until the next read from Flash. The request to count is synchronized between the PerCLK domain and the AHB clock domain. Therefore, up to two PerCLK and two AHB clock periods are added to the period.

31.3.10.14. CRCPAUSE SFR Description

The PAUSE bit halts CRC reads of Flash memory. The CRC FSM only checks the PAUSE bit when the PERIOD counter (being zero) triggers a read. The counter and the LFSR do not pause. When the FSM exits pause with the counter being zero, it requests a read of Flash to feed the LFSR.

Software set and clear of the PAUSE bit may occur without causing the FSM to actually pause operation. This type of interaction can occur during the time that the PERIOD counter is non-zero.

31.3.10.15. CRCMADR SFR Description

CRCMADR is the byte address of the first byte in the message to be checked. The CRC logic requires a system physical address to determine the region of Flash targeted: boot vs main and Flash Panel 1 vs Flash Panel 2.

31.3.10.16. CRCMLEN SFR Description

The CRCMLEN field defines the message length in bytes. The Message Length is limited to the size of the largest contiguous region of Flash memory.

31.3.10.17. CRCIV SFR Description

The CRCIV defines the Initial Value which is loaded into the CRCACC at the beginning of the CRC calculation. Both Manual and Auto Repeat mode use the IV. CRCCTRL.PLEN32 controls the final Initial Value size. When selecting CRC-16, bit 16 and above are not used.

31.3.10.18. CRCACC SFR Description

The CRCACC, CRC Accumulator, reports the LFSR state which changes during the calculation for each byte shifted in. Section CRCSUM SFR Description defines how the compare equation uses CRCACC when the CRC calculation completes. PLEN32 controls the accumulator size. When selecting CRC-16, bit 16 and above are not used.

31.3.10.19. CRCPOLY SFR Description

The CRCPOLY holds the desired Polynomial for the CRC. The highest bit of the polynomial is not contained in CRCPOLY but is always set to one in the CRC logic. The lowest bit, CRCPOLY[0], is always one and not writable. PLEN32 controls the polynomial size. When selecting CRC-16, bit 16 and above are not used.

31.3.10.20. CRCFXOR SFR Description

CRCFXOR contains the value to XOR with CRCACC in the final calculation before comparing to CRCSUM. Section ???? below defines how the compare equation uses CRCFXOR when the CRC calculation completes. PLEN32 controls the Final XOR size. When selecting CRC-16, bit 16 and above are not used.

31.3.10.21. CRCSUM SFR Description

CRCSUM contains the checksum value. CRCCTRL.PLEN32 controls the checksum size. When selecting CRC-16, bit 16 and above are not used.

The CRC Compare Equation is given by:

Compare True = CRCSUM == (CRCACC xor CRCFXOR)

31.3.10.22. CRC Bits in INTFLAG

31.3.10.22.1. CRCDONE

For manual CRC, the CRCDONE bit indicates that the operation is complete with all bits in the message acted upon. The CRCERR bit may or may not be set. For auto repeat CRC, the CRCDONE bit is only set when there is an error so CRCERR is also set. This condition also causes the looping to stop.

31.3.10.22.2. CRCERR

The CRCERR bit reports the status of the compare at the end of the CRC when all bits in the message finish shifting into the LFSR.

CRCERR = ![CRCSUM = (CRCACC xor CRCFXOR)]

31.3.11. Error Correction Code (ECC) Support

31.3.11.1. Read While Write (RWW, also known as *Live Update*)

With two Flash panels in the system, it is possible to execute from one Flash panel while programming another (see Concurrent Access). Consequently, the ECC generation logic is duplicated to support ECC parity bit generation for data writes and ECC syndrome generation for data reads.

31.3.12. Error Correction Logic

There are two device configuration fields (i.e. fuses) stored in the Flash that are used for ECC control. The first is FECCCTL which is the reset value of the SFR based ECCCTRL.ECCCTL[1:0]. The second is FECCUNLCK which is the reset value of ECCCTRL.ECCUNLCK.

31.3.12.1. ECCCTRL SFR Description

31.3.12.1.1. ECCUNLCK

The ECC mode of the Flash can be locked for the duration of the program lifetime in Flash. When FECCUNLCK is 0, ECCUNLCK is also 0 and the selected ECC mode cannot be changed until Flash is updated. This option prevents undesired changing of the mode. When FECCUNLCK is 1 (the default erased state of the flash), ECCUNLCK and ECCCTL can be modified.

Note: If ECCUNLCK is 0, debug mode cannot override the ECC or error reporting via DBGCTRL.

31.3.12.1.2. ECCCTRL

The field ECCCTL determines how the parity bits are used for Flash reads and writes. The four options, ECC, Dynamic, Dynamic w/o Bus Error and Bypass affect reads and writes differently.

For all ECC modes, writes to the Flash update the Flash ECC Control Bits, CTL[2:0], which store whether ECC or Simple Parity was calculated on the data. The Control Bits exists per Flash word (256-bit data). If the FCW performs a Single Write then the CTL is written with 3'b111 (i.e. not changed from the default erase value of the bits) for Parity. If the FCW performs a Quad Write then the CTL is written as 3'b000 for ECC. CTL[2:0] must be 3'b111 for Single Writes using Simple Parity since all Flash ECC Control Bits (CTL) are not updated with a Single Write. CTL[2:0] is updated for Quad Writes so 3'b000 works for selecting ECC.

The ECC Control Bits table shows the allowed operations based on ECCCTL and its effect on the CTL[2:0] field.

31.3.12.1.3. ECC Writes with ECC Reads

If ECCCTL[1:0]= 2'b00, ECC hardware is always active. This mode is also referred to as ECC On.

Only Quad Write programming is allowed in this mode, which always calculates and stores ECC values for the data. The FCW disables the Single Write Program command by making it a no-op if an attempt is made to execute it. Note Row Write uses multiple Quad Writes for programming.

Reads in this mode ignore Flash ECC Control bits CTL[2:0] and always perform ECC error checking and correction. This mode can generate ECC SEC events or ECC DED events.

Note: The FCR forces ECC to be enabled for all calibration word reads during the reset sequence.

31.3.12.1.4. Dynamic Writes with Dynamic Reads

If ECCCTL[1:0]= 2'b01, ECC hardware dynamically switches between ECC and Parity. This mode is also referred to as Dynamic ECC.

Both Quad Write and Single Write programming are allowed in this mode.

Reads in this mode obey Flash ECC Control bits CTL[2:0] to determine whether ECC or Parity is calculated on the data. This mode can generate ECC SEC events, and ECC DED events, and Parity Errors that are reported via the ECC DED event path.

Note that Flash ECC Control bits CTL[2:0] uses a majority detect during reads to make it tolerant to single bit failures. A single bit failure of CTL[2:0] is reported via the ECC SEC event path, and only occurs in this mode. If an ECC SEC occurs at the same time as a CTL SEC, the error is reported on ECC SEC event path since both single bit failures get corrected. They also only count as one error with respect to SECCNT.

31.3.12.1.5. Dynamic Writes with Dynamic Reads but w/o Bus Error (for DED or Parity)

If ECCCTL[1:0]= 2'b10, ECC hardware dynamically switches between ECC and Parity. However, Neither DED nor Parity Errors cause a Bus Error. The DERR bit is still set for uncorrectable errors. Otherwise, this mode behaves the same as in Dynamic Writes with Dynamic Reads, above.

31.3.12.1.6. Dynamic Writes with No Error Checks

If ECCCTL[1:0] = 2'b11, ECC hardware is bypassed (for reads). This mode is also referred to as ECC Bypass.

Both Quad Write and Single Write programming are allowed in this mode.

Reads in this mode ignore Flash ECC Control bits CTL[2:0] and never checks ECC or Parity. This mode never generates error events either SEC, DED, or single word parity error on DED.

31.3.12.2. ECC Double Error Detected (INTFLAG.DERR)

When ECC is active (i.e. not OFF/Bypass) and a read from Flash memory results in an ECC DED, the FCR returns a bus error to the initiator. The error is returned in-band with the data and the initiator determines how to handle the corrupted data using the initiator's bus error exception handler.

For the ECC Double Error Detect (DED), the FCR sets the INTFLAG.DERR bit, therefore the Bus Error exception handler can check this bit to determine that a double error has been detected by the ECC. A properly constructed bus error exception handler should be able to decode and report the address that triggered the double bit error. The data is loaded into the read buffer with a DED tag so subsequent reads of the address also causes bus errors.

31.3.12.3. Single Error Corrected (SERR)

When ECC is active (i.e. not OFF/Bypass) and a read from Flash memory results in an ECC Single Error Corrected (SEC), the FCR reports it via an interrupt since it is not a critical error. Data in the read buffer is correct and no further ECC events are generated for reads that hit the buffer as long as that data is in the buffer.

Each read of the Flash that results in an SEC causes the FCR to set INTFLAG.SERR when ECCCTRL.SECCNT = 0. If the count in SECCNT is non-zero, the FCR decrements it and does not set SERR. The FCR does not reload SECCNT when it is zero. Software must write the desired count each time it services the SEC interrupt event.

When Dynamic ECC is active, an SERR can be caused by a bit error in CTL. If this is the case, a DERR and an SERR can be caused by the same read.

31.3.12.4. Simple Parity Error

When Dynamic ECC is active and the read is from an address where Flash ECC Control bits CTL selects simple parity, a parity error causes the same behavior as an ECC DED. The error is reported in-band (if ECCCTL[1:0]= 2’b01) and the INTFLAG.DERR bit is set. Unlike a DED, a Bus Error exception is not returned to the initiator.

31.3.13. ECC Fault Control

The flash system contains a fault injection mechanism to allow periodic testing of the ECC logic and error event generation. It can be setup to test Parity, ECC SEC, and ECC DED data faults. For Dynamic mode, it can also inject faults in the ECC Control Bits. The registers ECCCTRL, FFLTCTRL, FFLTPTR, and FFLTADR provide control, while FFLTCAP, FFLTPAR and FFLTSYN provide status. Flash Fault registers are named FFLT* to distinguish them for other memory fault registers.

Note that if the system level ECCCTL bits are set for Bypass mode, the fault logic behaves as if it is in Dynamic Mode so that the functional safety code can be developed without generating system level errors.

The ECC Fault logic has two modes of operation: Injection and Capture. Injection mode allows user code to test the ECC logic by forcing faults into the Flash read or write paths. Fault injection behaves slightly differently depending on the ECCCTL mode. Capture mode allows user code to monitor read faults by capturing the first address at which an ECC or Parity error occurs.

31.3.13.1. Flash Panel ECC Organization

The following table shows the ECC Calculation Vector bit order with respect to flash data bits and parity bits. The first column is the ECC Calculation Vector. The next two columns map the Read/Write Data and ECC Parity bits to the Calculation Vector. V[n] defines the bit which is selected by FFLT*PTR, so n=FFLT*PTR.

Table 31-22. Flash ECC Vector

Vector Bits	Data Bits	ECC Parity Bits
V[n]	D[n]	EP[n]
0	-	0
1	-	1
2	-	2
3	0	-
4	-	3

Table 31-22. Flash ECC Vector (continued)

Vector Bits	Data Bits	ECC Parity Bits
V[n]	D[n]	EP[n]
7:5	3:1	-
8	-	4
15:9	10:4	-
16	-	5
31:17	25:11	-
32	-	6
63:33	56:26	-
64	-	7
127:65	119:57	-
128	-	8
136:129	127:120	-
255:137	246:128	-
256	-	9
265:257	255:247	-

The following table shows the ECCCTL and CTL field decode. The CTL field is stored in Flash and the fault logic can inject errors into it based on FFLTCTRL.CTLFLT.

Table 31-23. ECC Control Bits

Error Correction Mode	ECCCTL[1:0]	Write Value CTL[2:0]	Read Values CTL[2:0]	Operation
Bypass Mode	2'b11	3'b000	Don't Care	Quad Write w/ ECC Read w/o ECC Check
		3'b111	Don't Care	Single Write w/ Parity Read w/o Parity Check
Dynamic Mode w/o Bus Error	2'b10	3'b000	3'b000 3'b001 3'b010 3'b100	Quad Write w/ ECC Read w/ ECC but w/o DED bus error DERR valid
		3'b111	3'b111 3'b110 3'b101 3'b011	Single Write w/ Parity Read w/o Parity buserror DERR valid
Dynamic Mode	2'b01	3'b000	3'b000 3'b001 3'b010 3'b100	Quad Write w/ ECC Read w/ ECC
		3'b111	3'b111 3'b110 3'b101 3'b011	Single Write w/ Parity Read w/ Parity
ECC Mode	2'b00	3'b000	Don't Care	Quad Write w/ ECC Read w/ ECC
		N/A	N/A	Single Write is not available in this mode. All reads use ECC.

Note: If switching modes, Single Writes in Bypass or Dynamic cause SEC and DED errors in ECC mode. It is highly recommend to select and use only one Error Correction Mode.

31.3.13.2. Usage Model for Fault Injection in ECC Mode

Fault Injection occurs at only the Flash address selected by the SFR FFLTADR[31:0]. The FFLTCTRL field FLTMD[2:0] determines the type of fault injected, (single or double) and if it is for reads or writes of Flash. The two fields in the FFLTPTR register, FLT1PTR and FLT2PTR, point to the Vector bit or bits (as ordered in Table 2-3) to invert. Single fault injection always uses FLT1PTR.

Fault Injection always occurs between the ECC logic and the Flash. For writes this means that errors are inject after the ECC/Parity calculation but prior to the data write to Flash. For reads this means that errors are injected after the read from Flash but prior to ECC/Parity calculation.

In FFLTPAR, the SFR fields SECIN and DEDIN capture the parity bits as they are read from the Flash. They have no meaning for writes as the bits are always driven to zeros (during reads and writes) for the calculation. The SFR fields SECOUT and DEDOUT capture the calculated value for either reads or writes.

In FFLTSYN, the SFR field SECSYN captures the syndrome of the read. This is the XOR of the SECIN with SECOUT. The SFR field DEDSYN captures the overall parity of the values read from Flash. If there is NO Overall Parity change this bit is ZERO. If there is an Overall Parity change this bit is ONE. The following table shows how each of the four conditions are determined. For SEC, the SECSYN points to the bit that was in ERROR.

Though the ECC/Parity mode bits CTL are not used in this mode, the CTLFLT bits are still effective. They will alter writes and reads of that field. However, since CTL is not used in ECC Mode, errors in CTL have no effect on the ECC calculation or correction.

In ECC Mode, the fields PERR, CTLSTAT and CERR in FFLTSYN are meaningless.

Table 31-24. Error Decode

SECSYN	DEDSYN	SERR	DERR	Condition
Zero	0	0	0	No Error
Zero	1	1	0	DED parity Bit Error
Non-Zero	0	0	1	Double Error
Non-Zero	1	1	0	SEC – Data bit error corrected

31.3.13.3. Usage Model for Fault Injection in Dynamic Model

When ECCCTRL.ECCCTL selects Dynamic ECC mode, the FFLTCTRL.CTLFLT bits can also be used to inject a fault in the Dynamic ECC CTL bits. This operation is performed in parallel with data fault injection.

Though more than one CTL bit can be inverted, the logic is only capable of correcting single bit errors in the CTL field. For writes, the selected bit is stored in Flash inverted from the normal value. For reads the selected bit is inverted before it is used to determine the ECC mode.

The CTLSTAT field in FFLTSYN reports the captured values of CTL from the read or write that hit FFLTADR. ECC Control Bits shows the values of CTL for which the Flash system interprets ECC vs Parity reads and generates the values for writes. Note that injecting an error on the CTL value for a write does not change whether the write uses ECC or Simple Parity.

When performing a Quad Write operation the fault logic, control and status behave the same as in ECC Mode. When performing a read and FFLTSYN.CTLSTAT indicates “Read w/ ECC”, the fault logic, control and status also behave as in ECC Mode. However, FFLTSYN.CERR indicates if CTLSTAT has a bit failure.

When performing Single Write operations, the fault logic injects faults the same way as in ECC Mode. However, some parity bits are not used for Simple Parity (see Table 2-3) and do not affect the calculation when read. Also, the SEC* and DED* fields plus PERR, CLSTAT, CERR, DERR and SERR are “Don’t Care”.

When performing a read and FFLTSYN.CTLSTAT indicates “Read w/ Parity”, SEC*, DED*, DERR, SERR are “Don’t Care”. The value in FFLTSYN.PERR reflects error status of each Write Word. If a PERR = 1, then a bus error is generated on the DED error path.

31.3.13.4. Usage Model for Fault Capture

When FFLTCTRL.FLTMD selects Fault Capture Mode, logic captures ECC/Parity status information. Capture only occurs when the INTFLAG.FLTCAP = 0, and never when it is ONE. In other words, once a fault capture has occurred any additional faults are ignored until INTFLAG.FLTCAP is cleared. Capture occurs for a read only on the detection of a fault which also cause logic to set INTFLAG.FLTCAP = 1. Capture occurs for each flash write but the INTFLAG.FLTCAP bit remains ZERO. This allows write ECC logic to be checked without interfering with fault events causing a read capture. Note that the write ECC/Parity logic is incapable of generating an error interrupt.

If there are simultaneous read errors, the event associated with AHB0 is captured. If AHB0 is not involved, then the event associated with Panel 1 is captured and the event associated with Panel 2 is dropped. If the simultaneous capture is between a read and a write, the read event is captured.

Only fields pertinent to the ECC/Parity Mode are valid. FFLTCAP.FLTADR always captures the address at which the fault occurs (or for any write). FFLTSYN.CTLSTAT and FFLTSYN.CERR are valid only for Dynamic ECC mode. If the access uses ECC then control logic updates the FFLTSYN fields SECSYN, DEDSYN, SERR, and DERR, plus all fields in FFLTPAR. If the access uses Simple Parity then control logic updates FFLTSYN.PERR.

31.3.13.5. System Effects on Fault Injection

Entry to Debug Mode or an Interrupt Service Routine can cause discrepancy between what the Fault Logic sees and what the CPU/system sees. This does not cause an erroneous condition for the system. However, the Fault Logic can show an event when it is not manifested in the system.

Because entry to these states are event driven and not part of the program flow the CPU can abort an instruction (or data) fetch in order to service the interruption. Fault code is susceptible to this condition if it happens to perform a read of Flash at this juncture.

31.3.13.5.1. Interrupts

When FFLTCTRL.FLTMD selects one of the Fault Injection modes, fault logic does NOT set INTFLAG.FLTCAP. User code is expected to know it is accessing the Flash at FFLTADR causing a fault to be injected. However, if the access is a read, user code can expect to experience a system error/interrupt associated with the fault (that is, either an SEC interrupt or a bus error related to DED). Note that faults detected in Bypass Mode do not cause system error/interrupts.

When FLTMD selects a Capture mode, fault logic sets INTFLAG.FLTCAP = 1 when it detects a CTL, SEC, or DED fault (captured and reported respectively by the FFLTSYN bits CERR, SERR, or DERR) from the Flash read data path. Note that CTL faults detected in ECC Mode (ECCCTRL.ECCCTL[1:0]=2’b00) do not set INTFLAG.FLTCAP.

When INTFLAG.FLTCAP = 1, the Fault Logic halts operation allowing user code to service the interrupt. When INTFLAG.FLTCAP = 0, the Fault Logic resumes/continues operation.

31.3.13.5.2. Effects on DBGCTRL on FLT

ECC and Fault capture are not controlled by DBGCTRL.CRCRUN. When the CPU is halted in debug mode the Flash ECC and Fault capture works as defined by DBGECC field in DBGCTRL.

31.3.13.6. FFLTCTRL SFR Description

31.3.13.6.1. FLTEN

When FLTEN = 0, the Fault Injection Logic is disabled and user code can safely update the SFR control bits.

When FLTEN = 1 the Fault Injection Logic performs the operation selected by FLTMD. Modifying any other control bit(s) produce(s) undefined results.

The operation description of all subsequent bits is predicated on the Fault Logic being enabled.

31.3.13.6.2. CTLFLT

When FLTMD selects one of the Fault Injection modes, CTLFLT[2:0] controls fault injection on the Flash ECC Control bits CTL[2:0]. When CTLFLT[n] = 1 logic injects a fault in the associated bit position of CTL[n]. When CTLFLT[n] = 0 logic does not inject a fault on to the associated bit in CTL[n].

Control faults cannot be injected separately from parity faults. In all other FLTMD modes, CTLFLT[2:0] is unused. Also, CTLFLT is only meaningful for Dynamic ECC mode.

31.3.13.6.3. FLTMD

There are four basic fault modes: Disabled, Capture, Read Fault, and Write Fault. The selected mode takes affect when FLTEN = 1.

Fault Injection Disabled provides the ability to complete disable and lock out the Fault Injection logic. Applications that never want or need to test ECC logic can use this mode to ensure a fault is never injected into the Flash read or write path. The lock out remains until the next system reset.

Fault Capture Mode allows the application to monitor the Flash for errors and take the appropriate action. Often times bit errors detected by ECC SEC are soft failures. They may be retention failures or read disturb failures which are correctable by re-writing the data. An application that wishes to decrease the likelihood of an uncorrectable ECC DED fault can reserve a spare page in Flash. The spare page provides a location to copy the offending page while correcting the SEC bit error(s). The offending page would then be erased and re-written with the original data intact. Fault Capture has a lock mode to prevent accidentally switching to Fault Injection Mode.

Fault Injection mode allows user code to periodically test the ECC logic to verify it is working correctly. There are four options. The first two inject a single or double fault for reads. The second two inject a single or double fault for writes. Depending on the fault testing requirements the write option may not be need for single panel system since reads and writes share the same ECC calculation module. However, dual panel systems with Live Update never share the same ECC module.

31.3.13.7. FFLTPTR SFR Description

31.3.13.7.1. FLT1PTR[8:0]

The Fault Pointer inverts the selected bit in the ECC Calculation Vector, V. For Single or Double Fault injection, logic inverts bit V[FLT1PTR], as defined in the Flash ECC Vector table. The table shows the mapping between the calculation vector and Data, ECC and Parity bits. For a 128-bit data Flash FLT1PTR[8] is unused.

31.3.13.7.2. FLT2PTR[8:0]

The Fault Pointer inverts the selected bit in the ECC Calculation Vector. For Double Fault injection, logic inverts bit V[FLT2PTR], as defined in the Flash ECC Vector table. The table shows the mapping between the calculation vector and Data, ECC and Parity bits.

31.3.13.8. FFLTADR SFR Description

FFLTADR.FLTADR contains the System Physical Flash Address at which a Fault Injection occurs.

31.3.13.9. FFLTCAP SFR Description

FFLTCAP.FLTADR contains the System Physical Flash Address at which a Fault Capture occurred.

31.3.13.10. FFLTPAR SFR Description

This register contains the values input to and output from the ECC Calculation Module. For reads the values are post fault injection but for writes they are pre-fault injection. This register does not contain valid information when using Simple Parity.

Note: In Capture Mode faults are not injected.

31.3.13.10.1. SECIN[8:0]

This is the SEC Parity for data read from the Flash panel. For a write this field is always ZERO. For a 128-bit data Flash SECIN[8] is unused.

31.3.13.10.2. DEDIN

This is the Overall Parity for data read from the Flash panel. For a write this field is always ZERO.

31.3.13.10.3. SECOUT[8:0]

This is the SEC Parity calculated on either the read or write data.

31.3.13.11. FFLTSYN SFR Description

This register contains Simple Parity and ECC Parity status. For reads the values are post fault injection but for writes they are pre-fault injection.

Note: In Capture Mode faults are not injected.

31.3.13.11.1. SECSYN[8:0]

SEC Syndrome represents the bitwise XOR of SECIN and SECOUT. SECSYN is not valid for any write or Simple Parity mode.

31.3.13.11.2. DEDSYN

DED Syndrome represents the overall parity from all data bits, SEC Parity bits and the DED Parity bit read from Flash. DEDSYN is not valid for any write or Simple Parity mode.

31.3.13.11.3. DERR and SERR

Double Error and Single Error status indicate the read state of the data: No Error, Single Error, or Double Error. The Error Decode table shows how DEDSYN and SECSYN determine their values. These bits are not valid for any write or Simple Parity mode.

31.3.13.11.4. CERR

Control Error indicates the single bit error status of the CTL[2:0] field. CERR is only meaningful for reads in Dynamic ECC or Bypass ECC modes.

31.3.13.11.5. CTLSTAT[2:0]

CTLSTAT[2:0] reports the read value of the CTL [2:0] field stored in Flash for the addressed (FLTADR) Flash word (256-bit data). CTLSTAT is only meaningful for reads in Dynamic ECC or Bypass ECC modes.

31.3.13.11.6. PERR[3:0]

Parity Error captures the difference between each word's read parity and its calculated parity. The word size is dependent on the Flash and is either 32-bit or 64-bit. PERR[0] represents the error status of the lowest addressed word in of the Flash read, while PERR[3] is for the highest addressed word. PERR is only meaningful for reads in Dynamic ECC or Bypass ECC modes.

31.3.14. Flash Read Protect Features

Flash read protection features are for security of protected information. A page containing boot code can be locked and cannot be read until the next reset.

Read protection controlled by the Flash system is directly protected within Flash system and does not require setup external to the FCR.

31.3.14.1. CFG Read Protect (CRP)

The Boot CFG page contains key information used to protect against unauthorized chip erase requests. CRP provides read protection for Boot CFG to prevent any accesses to the page. With a dual boot system there is a Boot CFG page in each panel's CFM for which BC1RP and BC2RP provide individual access control.

Trusted code can configure the read protect setting for Boot CFG pages. If trusted code does not configure the settings and lock them then normal code can do so.

When read protected, read of this page cause a bus error.

31.3.14.1.1. Local Lock Bits

The user may optionally select to prevent further writes to a CRP register bits by also setting associated local LOCK bits (in the same register) when writing the CRP value.

When a local lock bit is set (BC*n*RPLOCK = 1, *n* = 1 or 2), subsequent writes to that register (even if the unlock sequence is followed) have no effect, creating a “write once’ register. Local LOCK bits revert to the unlocked state at reset.

31.3.15. DMA

The FCR does not use the DMA module.

31.3.16. Interrupts

There are five interrupt flags in the INTFLAG register:

- FLTCAP – ECC Fault Capture. See Interrupts.
- CRCERR – CRC Error. See CRC Bits in INTFLAG.
- CRCDONE – CRC Calculation Done. See CRC Bits in INTFLAG.
- DERR – ECC Double Error Detected. See ECC Double Error Detected (INTFLAG.DERR).
- SERR – Single Error Detected and Corrected. See Single Error Corrected (SERR).

31.3.17. Events

The FWR does not use events or generate them.

31.3.18. Register Summary

For descriptions and definitions of both Register and bitfield properties, refer to Register Properties.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRLA	31:24								
		23:16					RDBUFWS[3:0]			
		15:8	AUTOWS	ADRWS			FWS[3:0]			
		7:0						ARB		
0x04 ... 0x07	Reserved									
0x08	INTENCLR	31:24								
		23:16								FLTCAP
		15:8							CRCERR	CRCDONE
		7:0							DERR	SERR
0x0C	INTENSET	31:24								
		23:16								FLTCAP
		15:8							CRCERR	CRCDONE
		7:0							DERR	SERR
0x10	INTFLAG	31:24								
		23:16								FLTCAP
		15:8							CRCERR	CRCDONE
		7:0							DERR	SERR
0x14 ... 0x17	Reserved									
0x18	DBGCTRL	31:24								
		23:16								
		15:8								
		7:0						DBGECC[1:0]		CRCRUN
0x1C	ECCCTRL	31:24								
		23:16								
		15:8	SECCNT[7:0]							
		7:0		ECCUNLCK	ECCCTRL[1:0]					
0x20	CRCCTRL	31:24	PERIOD[15:8]							
		23:16	PERIOD[7:0]							
		15:8	RIN	ROUT	AUTOR	PLEN32				
		7:0			RUNSTDBY				CRCEN	CRCRST
0x24	CRCPAUSE	31:24								
		23:16								
		15:8								
		7:0								PAUSE
0x28	CRCMADR	31:24					CRCMADR[27:24]			
		23:16	CRCMADR[23:16]							
		15:8	CRCMADR[15:8]							
		7:0	CRCMADR[7:0]							
0x2C	CRCMLEN	31:24								
		23:16	CRCMLEN[23:16]							
		15:8	CRCMLEN[15:8]							
		7:0	CRCMLEN[7:0]							
0x30	CRCIV	31:24	CRCIV[31:24]							
		23:16	CRCIV[23:16]							
		15:8	CRCIV[15:8]							
		7:0	CRCIV[7:0]							
0x34	CRCAC	31:24	CRCACC[31:24]							
		23:16	CRCACC[23:16]							
		15:8	CRCACC[15:8]							
		7:0	CRCACC[7:0]							
0x38	CRCPOLY	31:24	CRCPOLY[31:24]							
		23:16	CRCPOLY[23:16]							
		15:8	CRCPOLY[15:8]							
		7:0	CRCPOLY[7:0]							

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x3C	CRCFXOR	31:24	CRCFXOR[31:24]							
		23:16	CRCFXOR[23:16]							
		15:8	CRCFXOR[15:8]							
		7:0	CRCFXOR[7:0]							
0x40	CRCSUM	31:24	CRCSUM[31:24]							
		23:16	CRCSUM[23:16]							
		15:8	CRCSUM[15:8]							
		7:0	CRCSUM[7:0]							
0x44	FFLTCTRL	31:24								
		23:16								
		15:8		FLTMD[2:0]				CTLFLT[2:0]		
		7:0							FLTEN	FLTRST
0x48	FFLTPTR	31:24								FLT2PTR[8]
		23:16	FLT2PTR[7:0]							
		15:8								FLT1PTR[8]
		7:0	FLT1PTR[7:0]							
0x4C	FFLTADR	31:24					FLTADR[27:24]			
		23:16	FLTADR[23:16]							
		15:8	FLTADR[15:8]							
		7:0	FLTADR[7:0]							
0x50	FFLTCAP	31:24					FLTADR[27:24]			
		23:16	FLTADR[23:16]							
		15:8	FLTADR[15:8]							
		7:0	FLTADR[7:0]							
0x54	FFLTPAR	31:24	DEDOUT							SECOUT[8]
		23:16	SECOUT[7:0]							
		15:8	DEDIN							SECIN[8]
		7:0	SECIN[7:0]							
0x58	FFLTSYN	31:24	PERR3	PERR2	PERR1	PERR0		CTLSTAT[2:0]		
		23:16						CERR	DERR	SERR
		15:8	DEDSYN							SECSYN[8]
		7:0	SECSYN[7:0]							
0x5C	CRP	31:24						BC2RPLOCK		
		23:16						BC1RPLOCK		
		15:8						BC2RP		
		7:0						BC1RP		

31.3.18.1. Control A Register

Name: CTRLA
Offset: 0x0000
Reset: 0x00008000
Property: PAC Write Protection

Table 31-25. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					RDBUFWS[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
	AUTOWS	ADRWS			FWS[3:0]			
Access	R/W	R/W			R/W	R/W	R/W	R/W
Reset	1	0			0	0	0	0

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

Bits 19:16 – RDBUFWS[3:0] Data Returned from the Read Buffer match the Flash Wait States
When returning data from the AHB n read buffer, insert wait states to match ADRWS and FWS from Flash based on RDBUFWS[n].

Value	Description
0	Zero wait states for hits to AHB*n* read buffer.
1	ADRWS + FWS wait state for hits to AHB*n* read buffer, *n* = 0 to 2.

Bit 15 – AUTOWS Automatic Wait State Enable
Taws = Tacc + 5ns + 2 clocks

Value	Description
0	Use FWS: Total flash wait states are ADRWS + FWS
1	Use Automatic wait states: Total flash wait states are ADRWS + Taws

Bit 14 – ADRWS Address Wait State Enable
For Total flash wait states see AUTOWS.

Value	Description
0	Add 0 Address Wait States - allowing for higher performance at lower clock frequencies
1	Add 1 Address Wait State - allowing for higher clock frequencies

Bits 11:8 – FWS[3:0] Flash Access Time Defined in terms of AHB Clock Wait States

`1111`= Fifteen Wait States
`1110`= Fourteen Wait States
...
`0001`= One Wait State
`0000`= Zero Wait States

Note: This is not the wait states seen by the CPU. For Total Flash wait states see AUTOWS.

Bit 2 – ARB AHB Arbitration Scheme

Value	Description
0	Round Robin Arbitration.
1	Fixed priority AHB0 highest to AHB2 lowest

31.3.18.2. Interrupt Enable Clear Register

Name: INTENCLR
Offset: 0x0008
Reset: 0x00000000
Property: PAC Write-Protection

Table 31-26. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
								FLTCAP
Access								R/W
Reset								0

Bit	15	14	13	12	11	10	9	8
							CRCERR	CRCDONE
Access							R/W	R/W
Reset							0	0

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

Bit 16 – FLTCAP ECC Fault Capture Interrupt Disable
Writing a '0' to this bit has no effect.
Writing a 1 to this bit will disable the ECC Fault Capture as an interrupt request.
Reading this bit returns whether this interrupt is enabled (=1 ≥ enabled).

Bit 9 – CRCERR CRC Error Interrupt Disable
Writing a '0' to this bit has no effect.
Writing a 1 to this bit will disable the CRC Error as an interrupt request.
Reading this bit returns whether this interrupt is enabled (=1 ≥ enabled).

Bit 8 – CRCDONE CRC Calculation Done Interrupt Disable Bit
Writing a '0' to this bit has no effect.
Writing a 1 to this bit will disable the CRC Calculation Done as an interrupt request.
Reading this bit returns whether this interrupt is enabled (=1 ≥ enabled).

Bit 1 – DERR ECC Double Error Detected Interrupt Disable
Writing a '0' to this bit has no effect.
Writing a 1 to this bit will disable the ECC Double Error Detected as an interrupt request.
Reading this bit returns whether this interrupt is enabled (=1 ≥ enabled).

Bit 0 – SERR Single Error Corrected Interrupt Disable

Writing a '0' to this bit has no effect.
Writing a 1 to this bit will disable the Flash SEC as an interrupt request.
Reading this bit returns whether this interrupt is enabled (=1 ≥ enabled).

31.3.18.3. Interrupt Enable Set Register

Name: INTENSET
Offset: 0x000C
Reset: 0x00000000
Property: PAC Write-Protection

Table 31-27. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
								FLTCAP
Access								R/W
Reset								0

Bit	15	14	13	12	11	10	9	8
							CRCERR	CRCDONE
Access							R/W	R/W
Reset							0	0

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

Bit 16 – FLTCAP ECC Fault Capture Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a 1 to this bit will enable the ECC Fault Capture as an interrupt request.
Reading this bit returns whether this interrupt is enabled (=1 ≥ enabled).

Bit 9 – CRCERR CRC Error Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a 1 to this bit will enable the CRC Error as an interrupt request.
Reading this bit returns whether this interrupt is enabled (=1 ≥ enabled).

Bit 8 – CRCDONE CRC Calculation Done Interrupt Enable Bit
Writing a '0' to this bit has no effect.
Writing a 1 to this bit will enable the CRC Calculation Done as an interrupt request.
Reading this bit returns whether this interrupt is enabled (=1 ≥ enabled).

Bit 1 – DERR ECC Double Error Detected Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a 1 to this bit will enable the ECC Double Error Detected as an interrupt request.
Reading this bit returns whether this interrupt is enabled (=1 ≥ enabled).

Bit 0 – SERR Single Error Corrected Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a 1 to this bit will enable the Flash SEC as an interrupt request.
Reading this bit returns whether this interrupt is enabled (=1 ≥ enabled).

31.3.18.4. Interrupt Flag Register

Name: INTFLAG
Offset: 0x0010
Reset: 0x00000000
Property: PAC Write-Protection

Notes:
1. The interrupt flags in this register are set by hardware only.
2. Interrupt flags must be cleared and then read back to confirm the clear before exiting the ISR to avoid double interrupts.

Table 31-28. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
								FLTCAP
Access								R/W
Reset								0

Bit	15	14	13	12	11	10	9	8
							CRCERR	CRCDONE
Access							R/W	R/W
Reset							0	0

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

Bit 16 – FLTCAP ECC Fault Capture Flag
Writing a '0' to this bit has no effect.
Writing a 1 to this bit will clear the flag.

Value	Description
0	No Event.
1	An ECC Fault Capture, related to FFLTCTRL.FLTMD, has occurred. Write “1” to clear flag.

Bit 9 – CRCERR CRC Error Flag
Valid when CRCDONE = 1.
Writing a '0' to this bit has no effect.
Writing a 1 to this bit will clear the flag.

Value	Description
0	CRCACC Is Equal to the XOR of CRCSUM and CRCFXOR (No CRC Error)
1	CRCACC Is Not Equal to the XOR of CRCSUM and CRCFXOR (CRC Error has occurred). Write “1” to clear flag.

Bit 8 – CRCDONE CRC Calculation Done Flag
Writing a '0' to this bit has no effect.
Writing a 1 to this bit will clear the flag.

Value	Description
0	CRC calculation is not done.
1	CRC calculation is done. Write “1” to clear flag.

Bit 1 – DERR ECC Double Error Detected Flag
DED events are report in-band with returning read data and will be taken before a interrupt generated from this module.
Writing a '0' to this bit has no effect.
Writing a 1 to this bit will clear the flag.

Value	Description
0	No ECC Double Error Detected
1	ECC Double Error Detected. Write “1” to clear flag.

Bit 0 – SERR Single Error Corrected Flag
Writing a '0' to this bit has no effect.
Writing a 1 to this bit will clear the flag.

Value	Description
0	ECCCTRL.SECCNT Count not reached
1	ECCCTRL.SECCNT Count reached

31.3.18.5. Debug Control Register

Name: DBGCTRL
Offset: 0x0018
Reset: 0x00000000
Property: PAC Write Protection

Table 31-29. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

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

Bits 2:1 – DBGECC[1:0] Debug ECC Mode
ECC errors from debugger reads are:

Value	Description
x1	Not corrected, No Bus Error, INTLFAG is not updated, and FLT logic is not updated.
10	Corrected, Bus Error, INTFLAG is updated, and FLT logic operates as setup.
00	Corrected, No Bus ERR, INTFLAG is not updated, and FLT logic is not updated.

Bit 0 – CRCRUN CRC Debug Run

Value	Description
0	CRC Logic Halts in Debug Mode
1	CRC Logic Runs in Debug Mode

31.3.18.6. ECC Control Register

Name: ECCCTRL
Offset: 0x001C
Reset: 0x00000040
Property: PAC Write Protection

Table 31-30. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
	SECCNT[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
		ECCUNLCK	ECCCTRL[1:0]					
Access		R/W	R/W	R/W				
Reset		1	0	0				

Bits 15:8 – SECCNT[7:0] Flash SEC Count
SECCNT is the start value of an internal counter that decrements by 1 for each panel reporting an SEC event occurs (including ECC CTL[2:0] bit if in Dynamic ECC Mode). The SECCNT counter stops decrementing at zero. If an SEC error occurs when the SECCNT counter is zero, the INTFLAG.SERR flag bit is set.
Note: This field counts all SEC errors and is not limited to SEC errors on unique addresses.

Bit 6 – ECCUNLCK NVM ECC Mode Control Unlock
Note: This field can only be modified when ECCUNLCK=1.

The read value dictates the unlock state:

Value	Description
0	ECCUNLCK and ECCCTL[1:0] cannot be written
1	ECCUNLCK and ECCCTL[1:0] can be written

Bits 5:4 – ECCCTRL[1:0] NVM ECC Mode Control
Restricts one or more NVMOPs:

Value	Description
11	Dynamic Writes with No Error Check Reads
10	Dynamic Writes with SEC Reads but no DED/Parity Bus Error

Value	Description
01	Dynamic Writes with Dynamic Reads
00	ECC Writes with ECC Reads (NVMOP = Single Program Operation disabled)

31.3.18.7. CRC Control Register

Name: CRCCTRL
Offset: 0x0020
Reset: 0x00000000
Property: PAC Write Protection

Table 31-31. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	PERIOD[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
	PERIOD[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	RIN	ROUT	AUTOR	PLEN32				
Access	R/W	R/W	R/W	R/W				
Reset	0	0	0	0				

Bit	7	6	5	4	3	2	1	0
			RUNSTDBY				CRCEN	CRCRST
Access			R/W				R/W	R/W
Reset			0				0	0

Bits 31:16 – PERIOD[15:0] Read Period in Period Clock (PerCLK)counts
The number of PerCLK counts CRC logic waits between needing new data and reading that data from flash. (The PerCLK is clock at an 8 MHz fixed frequency. Non-zero PERIOD values are used to throttle back the bandwidth used for CRC calculations.)
0 = Read Data Immediately
All Other Values = Wait **PERIOD** PerCLK counts + 2 AHB Clocks (for sync) before starting

Bit 15 – RIN CRC Reflected Input
This option is sometimes referred to as Reflected Byte or Reflected Input.

Value	Description
0	The LFSR CRC is calculated Most Significant Bit first (Not Reflected)
1	The LFSR CRC is calculated Least Significant Bit first (Reflected)

Bit 14 – ROUT CRC Reflected Output
This option is sometimes referred to as Reflected Result or Reflected Output.

Value	Description
0	The CRCACC is Not Reflected
1	The CRCACC is Reflected (before the Final XOR)

Bit 13 – AUTOR CRC Auto Repeat

Value	Description
0	Perform CRC calculation once then set DONE and, if needed, set INTFLAG.CRCERR.
1	Continually Repeat CRC calculation; stop on error, set CRCDONE and CRCERR.

Bit 12 – PLEN32 Polynomial Length Select

Value	Description
0	Polynomial is16-bits
1	Polynomial is 32-bits

Bit 5 – RUNSTDBY CRC Run in Standby

Value	Description
0	CRC Stops in Standby
1	CRC Runs in Standby

Bit 1 – CRCEN Start CRC Calculation

Note: When CRCEN = 1 all other CRC* SFR bits are write protected, except CRCEN, CRCRST, and CRCPAUSE.PAUSE

Value	Description
0	CRC Stops in Standby
1	CRC Runs in Standby

Bit 0 – CRCRST CRC Reset

Value	Description
0	No Effect
1	Resets all CRC SFR (CRCCTRL and CRCPAUSE) bits.

31.3.18.8. CRC Pause Register

Name: CRCPAUSE
Offset: 0x0024
Reset: 0x00000000
Property: PAC Write Protection

Table 31-32. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

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

Bit 0 – PAUSE CRC Pause

Note: The CRC calculation continues until it needs more data, and then it pauses.

Prevent the CRC FSM from reading Flash memory so as to not interfere with CPU activity:

Value	Description
0	CRC Reads Flash as Required
1	Pause CRC Reads of Flash

31.3.18.9. CRC Message Address Register

Name: CRCMADR
Offset: 0x0028
Reset: 0x00000000
Property: PAC Write Protection

Table 31-33. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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

Bit	23	22	21	20	19	18	17	16
	CRCMADR[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
	CRCMADR[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
	CRCMADR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 27:0 – CRCMADR[27:0] Message Start Address in Flash
This is the system physical address of the first byte of the message.

31.3.18.10. CRC Message Length Register

Name: CRCMLEN
Offset: 0x002C
Reset: 0x00000000
Property: PAC Write Protection

Table 31-34. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	CRCMLEN[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
	CRCMLEN[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
	CRCMLEN[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 – CRCMLEN[23:0] Message Length in Bytes
Note: Only the number of bits needed to address the largest contiguous Flash region are implemented. (8 Mbyte of PFM = 24 bits for the length.)

31.3.18.11. CRC Initial Value Register

Name: CRCIV
Offset: 0x0030
Reset: 0x00000000
Property: PAC Write Protection

Table 31-35. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	CRCIV[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
	CRCIV[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
	CRCIV[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
	CRCIV[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 – CRCIV[31:0] CRC Initial Value

CRCIV is loaded into CRCACC before the start of each CRC calculation, including at the start of repeat loops.

31.3.18.12. CRC Accumulator Register

Name: CRCAC
Offset: 0x0034
Reset: 0x00000000
Property: PAC Write Protection

Table 31-36. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	CRCACC[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
	CRCACC[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
	CRCACC[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
	CRCACC[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – CRCACC[31:0] CRC Accumulator Result

31.3.18.13. CRC Polynomial Register

Name: CRCPOLY
Offset: 0x0038
Reset: 0x00000001
Property: PAC Write Protection

Table 31-37. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	CRCPOLY[31:24]							
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

Bit	23	22	21	20	19	18	17	16
	CRCPOLY[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
	CRCPOLY[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
	CRCPOLY[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 – CRCPOLY[31:0] CRC Polynomial Coefficients for LFSR

Note: CRCPOLY[0] is always 1 and CRCPOLY[32] is not implemented but implied to always be 1.

Value	Description
0	Disable the XOR input to the shift register at the associated bit position
1	Enable the XOR input to the shift register at the associated bit position

31.3.18.14. CRC Final XOR Register

Name: CRCFXOR
Offset: 0x003C
Reset: 0x00000000
Property: PAC Write Protection

Table 31-38. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	CRCFXOR[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
	CRCFXOR[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
	CRCFXOR[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
	CRCFXOR[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 – CRCFXOR[31:0] CRC Final XOR
CRCSUM is compared to the XOR of CRCACC and CRCFXOR.

31.3.18.15. CRC Checksum Register

Name: CRCSUM
Offset: 0x0040
Reset: 0x00000000
Property: PAC Write Protection

Table 31-39. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	CRCSUM[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
	CRCSUM[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
	CRCSUM[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
	CRCSUM[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 – CRCSUM[31:0] CRC Checksum
CRCSUM is compared to the XOR of CRCACC and CRCFXOR after CRCMLEN bytes have been computed.

31.3.18.16. Flash ECC Fault Control Register

Name: FFLTCTRL
Offset: 0x0044
Reset: 0x00000000
Property: PAC Write Protection

Table 31-40. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
			FLTMD[2:0]				CTLFLT[2:0]	
Access		R/W	R/W	R/W		R/W	R/W	R/W
Reset		0	0	0		0	0	0

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

Bits 14:12 – FLTMD[2:0] Fault Mode Control
Note: Write Protected when FLTEN = 1.

000 = Fault Injection Disabled
001 = Reserved
010 = Fault Capture Mode Enabled
Capture the address (in FLTADR) and Syndrome in (FLTSYN)
011 =Reserved
100 = Single Fault Injection (at bit selected by FLT1PTR) for Reads
101 = Double Fault Injection for Reads
110 = Single Fault Injection (at bit selected by FLT1PTR) for Writes
111 = Double Fault Injection for Writes

Bits 10:8 – CTLFLT[2:0] ECC/Parity Control Fault Bits
Note: Write Protected when FLTEN = 1.

If FLTMD = 1xx and FLTEN = 1:

Value	Description
0	No Fault Injected
1	Inject a Fault on to the associated ECC/Parity Control bits (CTL[n])

Bit 1 – FLTEN ECC Fault Enable Bit

Value	Description
0	ECC Fault Injection Disabled
1	ECC Fault Injection Enabled (module performs operation selected by FLT_MOD)

Bit 0 – FLTRST Fault Reset

Value	Description
0	No Effect
1	Resets all FLT SFR bits.

31.3.18.17. Flash ECC Fault Pointer Register

Name: FFLTPTR
Offset: 0x0048
Reset: 0x00000000
Property: PAC Write Protection

Table 31-41. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
								FLT2PTR[8]
Access								R/W
Reset								0

Bit	23	22	21	20	19	18	17	16
	FLT2PTR[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
								FLT1PTR[8]
Access								R/W
Reset								0

Bit	7	6	5	4	3	2	1	0
	FLT1PTR[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 24:16 – FLT2PTR[8:0] Fault 2 Injection Pointer
n = Inject fault at bit n. $0 \le n \le 265$.
For details of calculation vector bit order vs data bit order vs control bit order see the Flash ECC Vector table.
Note: For values of $n \ge 266$ the results will be undefined.

Bits 8:0 – FLT1PTR[8:0] Fault 1 Injection Pointer
n = Inject fault at bit n. $0 \le n \le 265$.
For details of calculation vector bit order vs data bit order vs control bit order see the Flash ECC Vector table.
Note: For values of $n \ge 266$ the results will be undefined.

31.3.18.18. Flash Fault Address Register

Name: FFLTADR
Offset: 0x004C
Reset: 0x00000000
Property: PAC Write Protection

Table 31-42. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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

Bit	23	22	21	20	19	18	17	16
	FLTADR[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
	FLTADR[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
	FLTADR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 27:0 – FLTADR[27:0] Fault Address
In Fault Injection Mode this is the System Physical Address at which to inject fault(s).
Note:
1. Address byte aligned but limited to read width of Flash (256-bits). Therefore, FLTADR[4:0] is read-only.

31.3.18.19. Flash Fault Capture Address Register

Name: FFLTCAP
Offset: 0x0050
Reset: 0x00000000
Property: PAC Write Protection

Table 31-43. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
					FLTADR[27:24]			
Access					R	R	R	R
Reset					0	0	0	0

Bit	23	22	21	20	19	18	17	16
	FLTADR[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
	FLTADR[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
	FLTADR[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 27:0 – FLTADR[27:0] Fault Address
In Fault Capture Mode this is the Flash Physical Address at which a fault was detected.

31.3.18.20. Flash Fault Capture Parity Register

Name: FFLTPAR
Offset: 0x0054
Reset: 0x00000000
Property: PAC Write Protection

Table 31-44. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	DEDOUT							SECOUT[8]
Access	R							R
Reset	0							0

Bit	23	22	21	20	19	18	17	16
	SECOUT[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
	DEDIN							SECIN[8]
Access	R							R
Reset	0							0

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

Bit 31 – DEDOUT Calculated Overall Parity used in Double Error Detection
For Writes this value is based on write data and the calculated SEC Parity bits.
For Reads this value is based on read data and the calculated SEC Parity bits.
Notes:
1. See the Flash ECC Vector table for calculation vector bit order vs data bit order vs control bit order.
2. “DED”, “Overall Parity”, and “Parity[0]” are used interchangeably.

Bits 24:16 – SECOUT[8:0] Calculated Single Error Parity bits
For Writes this value is based on write data.
For Reads this value is based on read data.
Notes:
1. See the Flash ECC Vector table for calculation vector bit order vs data bit order vs control bit order.
2. The number of active bits is dependent on the data width of the Flash panel.
3. The terms “SEC*[8:0]” and “Parity[9:1]” are used interchangeably.

Bit 15 – DEDIN Overall Parity from Flash
For Writes this value is always 0.

For Reads this value is the overall parity read from flash.

Notes:

1. See the Flash ECC Vector table for calculation vector bit order vs data bit order vs control bit order.
2. “DED”, “Overall Parity”, and “Parity[0]” are used interchangeably.

Bits 8:0 – SECIN[8:0] Single Error Parity bits from Flash

For Writes this value is always0.

For Reads this value is the Single Error Parity bits read from Flash.

Notes:

1. See the Flash ECC Vector table for calculation vector bit order vs data bit order vs control bit order.
2. The number of active bits is dependent on the data width of the Flash panel.
3. The terms “SEC*[8:0]” and “Parity[9:1]” are used interchangeably.

31.3.18.21. Flash ECC Fault Syndrome Register

Name: FFLTSYN
Offset: 0x0058
Reset: 0x00000000
Property: PAC Write Protection

Table 31-45. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	PERR3	PERR2	PERR1	PERR0		CTLSTAT[2:0]		
Access	R	R	R	R		R	R	R
Reset	0	0	0	0		0	0	0

Bit	23	22	21	20	19	18	17	16
						CERR	DERR	SERR
Access						R	R	R
Reset						0	0	0

Bit	15	14	13	12	11	10	9	8
	DEDSYN							SECSYN[8]
Access	R							R
Reset	0							0

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

Bits 28, 29, 30, 31 – PERR Per Word Parity Error Status
Note: Word size is defined by the Write Word size of the flash (64-bits).

Value	Description
0	No Parity Error Word n
1	Parity Error on Word n

Bits 26:24 – CTLSTAT[2:0] Parity vs ECC Control Status
Note: Panel is always programmed with 000 for ECC and 111 for parity.

000,001,010,100 = Calculation used ECC (i.e. programming used quad write)
011,101,110,111 = Calculation used Parity (i.e. programming used single write)

Bit 18 – CERR ECC Control bit Error

Value	Description
0	No Control bit Error (ECCSTAT either 111 or 000)
1	Single Control Bit Error

Bit 17 – DERR Double Error Detected
For Reads only when ECCSTAT = ECC

Value	Description
0	No Error
1	Double Error Detected

Bit 16 – SERR Single Error Corrected
For Reads only when ECCSTAT = ECC

Value	Description
0	No Error
1	Double Error Detected

Bit 15 – DEDSYN DED Syndrome
This is Overall Parity Calculated from Data and all Parity bits read from Flash.

Value	Description
0	Calculated Overall Parity Concurs with Read Overall Parity
1	Calculated Overall Parity Differs from Read Overall Parity

Bits 8:0 – SECSYN[8:0] Single Error Correction Syndrome
For Reads only when CTLSTAT = ECC or System bits ECCCTL[1:0]=ECC
This value is the bitwise XOR of SECIN and SECOUT.
If DEDSYN=1:
000000000 = No Data Error, but DED bit in Error
Non-Zero = SECSYN points to the bit position in the calculation vector that was corrected
If DEDSYN = 0:
000000000 = No Data Error or DED bit Error
Non-Zero = Double Error Detected.
Note: The number of active bits is dependent on the data width of the Flash panel.

31.3.18.22. CFM Page Read Protect Register

Name: CRP
Offset: 0x005C
Reset: 0x00000000
Property: PAC Write Protection

Table 31-46. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
						BC2RPLOCK		
Access						R/W		
Reset						0		

Bit	23	22	21	20	19	18	17	16
						BC1RPLOCK		
Access						R/W		
Reset						0		

Bit	15	14	13	12	11	10	9	8
						BC2RP		
Access						R/W		
Reset						0		

Bit	7	6	5	4	3	2	1	0
						BC1RP		
Access						R/W		
Reset						0		

Bit 26 – BC2RPLOCK Boot Configuration Page 2 Read Protection Lock Bit
Writing a 0 has no effect.
Writing a 1 enables the lock.
Notes:
1. This field can only be modified when BC2RPLOCK=0.
   This field can be set at the same time as writing to BC2RP.
   Once set, BC2RPLOCK can only be cleared by a reset.
2. The Boot ROM updates this field based on configuration settings.

Value	Description
0	BC2RP & BC2RPLOCK registers are NOT Locked and can be modified
1	BC2RP & BC2RPLOCK registers are Locked and cannot be modified

Bit 18 – BC1RPLOCK Boot Configuration Page 1 Read Protection Lock Bit
Writing a 0 has no effect.
Writing a 1 enables the lock.

Notes:
1. This field can only be modified when BC1RPLOCK=0.
   This field can be set at the same time as writing to BC1RP.
   Once set, BC1RPLOCK can only be cleared by a reset.
2. The Boot ROM updates this field based on configuration settings.

Value	Description
0	BC1RP & BC1RPLOCK registers are NOT Locked and can be modified
1	BC1RP & BC1RPLOCK registers are Locked and cannot be modified

Bit 10 – BC2RP Boot Configuration Page 2 Read Protect Bit

Notes:
1. This field can only be modified when BC2RPLOCK=0.
2. The Boot ROM updates this field based on configuration settings.

Value	Description
0	Read Protection for Boot Configuration Page 2 is Disabled
1	Read Protection for Boot Configuration Page 2 is Enabled

Bit 2 – BC1RP Boot Configuration Page 1 Read Protect Bit

Notes:
1. This field can only be modified when BC1RPLOCK=0.
2. The Boot ROM updates this field based on configuration settings.

Value	Description
0	Read Protection for Boot Configuration Page 1 is Disabled
1	Read Protection for Boot Configuration Page 1 is Enabled

32. Ethernet Media Access Controller (ETH)

32.1. Overview

The Ethernet Media Access Controller (ETH) module implements a 10/100 Mbps Ethernet MAC, compatible with the IEEE 802.3 standard. The ETH can operate in either half or full duplex mode at all supported speeds.

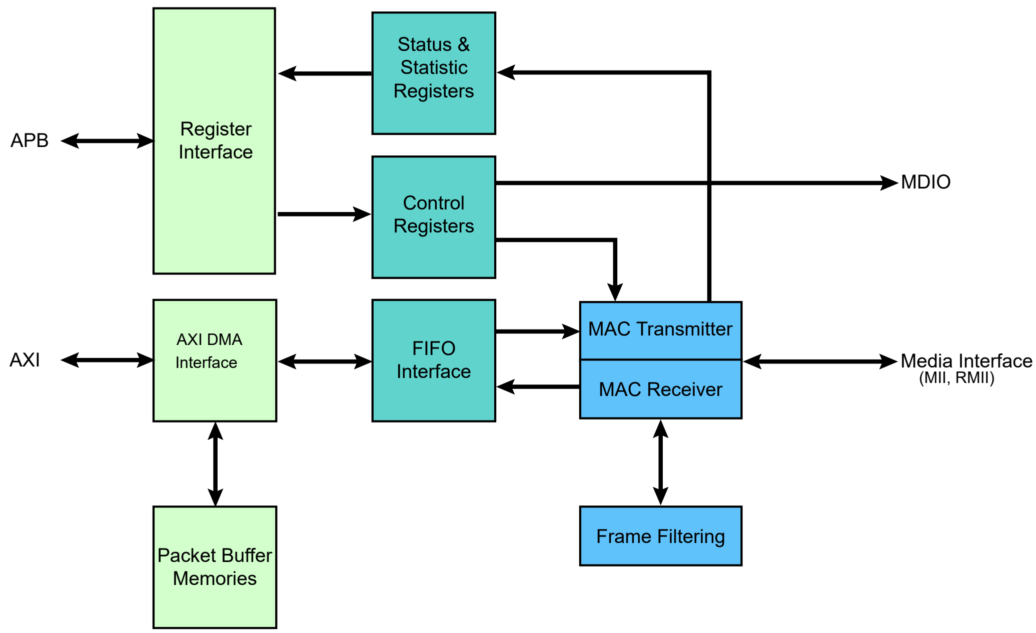
32.2. Features

The following are key features of the ETH module:

- Compatible with IEEE Standard 802.3
- 10 and 100 Mbps operation
- Full and half duplex operation at all supported speeds of operation
- Statistics Counter Registers for RMON/MIB
- MII/RMII interfaces to the physical layer
- Integrated physical coding
- Direct memory access (DMA) interface to external memory
- Support for 5 priority queues in DMA
- 8-KByte transmit RAM and 4-KByte receive RAM (refer to Queue Size for queue-specific sizes)
- Programmable burst length and endianism for DMA
- Interrupt generation to signal receive and transmit completion, errors or other events
- Automatic pad and cyclic redundancy check (CRC) generation on transmitted frames
- Automatic discard of frames received with errors
- Receive and transmit IP, TCP and UDP checksum offload. Both IPv4 and IPv6 packet types supported.
- Address checking logic for four specific 48-bit addresses, four type IDs, promiscuous mode, hash matching of unicast and multicast destination addresses and Wake-on-LAN
- Management Data Input/Output (MDIO) interface for physical layer management
- Support for jumbo frames up to 16383 Bytes
- Full duplex flow control with recognition of incoming pause frames and hardware generation of transmitted pause frames
- Half duplex flow control by forcing collisions on incoming frames
- Support for 802.1Q VLAN tagging with recognition of incoming VLAN and priority tagged frames
- Support for 802.1Qbb priority-based flow control
- Programmable Inter Packet Gap (IPG) Stretch
- Recognition of IEEE 1588 PTP frames
- IEEE 1588 time stamp unit (TSU)
- Support for 802.1AS timing and synchronization
- Support for 802.3az Energy Efficient Ethernet

32.3. Block Diagram

Figure 32-1. Block Diagram



32.4. Signal Interface

The ETH Controller module includes the following signal interfaces:

- MII and RMII to an external PHY
- MDIO interface for external PHY management
- Client APB interface for accessing ETH registers
- Host AXI interface for memory access
- TSUCOMP signal for TSU timer count value comparison

Table 32-1. Ethernet MAC Connections in Different Modes

Signal Name	Function	MII	RMII
ETH_TX_CLK(1)	Transmit Clock or Reference Clock	TXCK	REFCK
ETH_TXEN	Transmit Enable	TXEN	TXEN
ETH_TX[7:0]	Transmit Data	TXD[3:0]	TXD[1:0]
ETH_TXER	Transmit Coding Error	TXER	Not Used
ETH_RXCK	Receive Clock	RXCK	Not Used
ETH_RXDV	Receive Data Valid	RXDV	CRSDV
ETH_RX[7:0]	Receive Data	RXD[3:0]	RXD[1:0]
ETH_RXER	Receive Error	RXER	RXER
ETH_CRS	Carrier Sense and Data Valid	CRS	Not Used
ETH_COL	Collision Detect	COL	Not Used
ETH_MDC	Management Data Clock	MDC	MDC
ETH_MDIO	Management Data Input/Output	MDIO	MDIO

Note:

1. Input only. ETH_TX_CLK must be provided with a 25 MHz/50 MHz clock for MII/RMII interfaces.

32.5. Peripheral Dependencies

Peripheral Name	Base Address	NVIC IRQ Index: Source	MCLK AHBx/APBx Clock Enable Mask Bit	GCLK Peripheral Channel Clock Name:Register	PAC Peripheral Identifier (PAC.WRCTRL.PERIDx)	EVSYS Generator (EVSYS.CHANNELn.EVGENx)	Power Domain
ETH	0x4502_2000	145 : Q_0	AHB : MCLK.CLKMSK0[15] APB : MCLK.CLKMSK3[15]	CLK_ETH_TX : GCLK.PCHCTRL[41] CLK_ETH_TSU: GCLK.PCHCTRL[42]	53	124: TSU_CMP	VDDCORE_SW

32.6. Functional Description

32.6.1. Media Access Controller

The Transmit Block of the Media Access Controller (MAC) takes data from FIFO, adds preamble, checks and adds padding and frame check sequence (FCS). Both half duplex and full duplex Ethernet modes of operation are supported.

When operating in half duplex mode, the MAC Transmit Block generates data according to the Carrier Sense Multiple Access with Collision Detect (CSMA/CD) protocol. The start of transmission is deferred if Carrier Sense (CRS) is active. If Collision (COL) is detected during transmission, a jam sequence is asserted and the transmission is retried after a random back off. The CRS and COL signals have no effect in full duplex mode.

The Receive Block of the MAC checks for valid preamble, FCS, alignment and length, and presents received frames to the MAC address checking block and FIFO. Software can configure the ETH to receive jumbo frames of up to 10240 Bytes. It can optionally strip CRC (Cyclic Redundancy Check) from the received frame before transferring it to FIFO.

The Address Checker recognizes four specific 48-bit addresses, can recognize four different types of ID values, and contains a 64-bit Hash register for matching multicast and unicast addresses as required. It can recognize the broadcast address *all-'1'* (0xFFFFFFFFFFFF) and copy all frames. The MAC can also reject all frames that are not VLAN tagged, and can recognize Wake on LAN events.

The MAC Receive Block supports offloading of IP, TCP and UDP checksum calculations (both IPv4 and IPv6 packet types supported), and can automatically discard bad checksum frames.

32.6.2. IEEE 1588 Time Stamp Unit

The IEEE 1588 time stamp unit (TSU) is implemented as a 94-bit timer.

- The 48 upper bits [93:46] of the timer count seconds and are accessible in the ETH 1588 Timer Seconds High Register” (TSH) and ETH 1588 Timer Seconds Low Register (TSL)
- The 30 lower bits [45:16] of the timer count nanoseconds and are accessible in the ETH 1588 Timer Nanoseconds Register (TN)
- The lowest 16 bits [15:0] of the timer count sub-nanoseconds register (TISUBN)

The 46 lower bits roll over when they have counted to 1s. The timer increments by a programmable period (to approximately 15.2 femtosecond resolution) with each CLK_ETH_TSU period and can also be adjusted in 1ns resolution (incremented or decremented) through APB register accesses.

32.6.3. AXI Direct Memory Access Interface

The ETH DMA controller is connected to the MAC FIFO interface and provides a scatter-gather type capability for packet data storage.

The DMA implements packet buffering where dual-port memories are used to buffer multiple frames.

32.6.3.1. Packet Buffer DMA

- Easier to guarantee maximum line rate due to the ability to store multiple frames in the packet buffer, where the number of frames is limited by the amount of packet buffer memory and Ethernet frame size
- Full store and forward, or partial store and forward programmable options (partial store will cater for shorter latency requirements)
- Support for Transmit TCP/IP checksum offload
- Support for priority queuing
- When a collision on the line occurs during transmission, the packet will be automatically replayed directly from the packet buffer memory rather than having to re-fetch through the AXI (full store and forward ONLY)
- Received erroneous packets are automatically dropped before any of the packet is presented to the AXI (full store and forward ONLY), therefore reducing AXI activity
- Supports manual RX packet flush capabilities
- Optional RX packet flush when there is lack of AXI resource

32.6.3.2. Partial Store and Forward Using Packet Buffer DMA

The DMA uses SRAM-based packet buffers, and can be programmed into a low latency mode, known as Partial Store and Forward. This mode allows for a reduced latency as the full packet is not buffered before forwarding.
Note: This option is only available when the device is configured for full duplex operation.

This feature is enabled via the programmable TX and RX Partial Store and Forward registers (TPSF and RPSF). When the transmit Partial Store and Forward mode is activated, the transmitter will only begin to forward the packet to the MAC when there is enough packet data stored in the packet buffer. Likewise, when the receive Partial Store and Forward mode is activated, the receiver will only begin to forward the packet to the AXI when enough packet data is stored in the packet buffer. The amount of packet data required to activate the forwarding process is programmable via watermark registers. These registers are located at the same address as the partial store and forward enable bits.
Note: The minimum operational value for the TX partial store and forward watermark is 20. There is no operational limit for the RX partial store and forward watermark.

Enabling Partial Store and Forward is a useful means to reduce latency, but there are performance implications. The ETH DMA uses separate transmit and receive lists of buffer descriptors, with each descriptor describing a buffer area in memory. This allows Ethernet packets to be broken up and scattered around the AXI memory space.

32.6.3.3. Receive AXI Buffers

Received frames, optionally including FCS, are written to receive AXI buffers stored in memory. The receive buffer depth is programmable in the range of 64 Bytes to 16 KBytes through the DMA Configuration register (DCFGR), with the default being 128 Bytes.

The start location for each receive AXI buffer is stored in memory in a list of receive buffer descriptors at an address location pointed to by the receive buffer queue pointer. The base address for the receive buffer queue pointer is configured in software using the Receive Buffer Queue Base Address register (RBQB).

Each list entry consists of two words. The first is the address of the receive AXI buffer and the second the receive status.

If the length of a receive frame exceeds the AXI buffer length, the status word for the used buffer is written with zeroes except for the “Start of Frame” bit, which is always set for the first buffer in a frame.

Bit zero of the address field is written to 1 to show that the buffer has been used. The receive buffer manager then reads the location of the next receive AXI buffer and fills that with the next part of the received frame data. AXI buffers are filled until the frame is complete and the final buffer descriptor status word contains the complete frame status. See the following table for details of the receive buffer descriptor list.

Table 32-2. Receive Buffer Descriptor Entry

Bit	Function
Word 0	
31:2	Address of beginning of buffer
1	Wrap—marks last descriptor in receive buffer descriptor list.
0	Ownership—needs to be zero for the ETH to write data to the receive buffer. The ETH sets this to one once it has successfully written a frame to memory. Software has to clear this bit before the buffer can be used again.
Word 1	
31	Global all ones broadcast address detected
30	Multicast hash match
29	Unicast hash match
28	–
27	Specific Address Register match found, bit 25 and bit 26 indicate which Specific Address Register causes the match.
26:25	Specific Address Register match. Encoded as follows: 00: Specific Address Register 1 match 01: Specific Address Register 2 match 10: Specific Address Register 3 match 11: Specific Address Register 4 match If more than one specific address is matched only one is indicated with priority 4 down to 1.
24	This bit has a different meaning depending on whether RX checksum offloading is enabled. With RX checksum offloading disabled: (bit 24 clear in Network Configuration Register) Type ID register match found, bit 22 and bit 23 indicate which type ID register causes the match. With RX checksum offloading enabled: (bit 24 set in Network Configuration Register) 0: The frame was not SNAP encoded and/or had a VLAN tag with the Canonical Format Indicator (CFI) bit set. 1: The frame was SNAP encoded and had either no VLAN tag or a VLAN tag with the CFI bit not set.
23:22	This bit has a different meaning depending on whether RX checksum offloading is enabled. With RX checksum offloading disabled: (bit 24 clear in Network Configuration) Type ID register match. Encoded as follows: 00: Type ID register 1 match 01: Type ID register 2 match 10: Type ID register 3 match 11: Type ID register 4 match If more than one Type ID is matched only one is indicated with priority 4 down to 1. With RX checksum offloading enabled: (bit 24 set in Network Configuration Register) 00: Neither the IP header checksum nor the TCP/UDP checksum was checked. 01: The IP header checksum was checked and was correct. Neither the TCP nor UDP checksum was checked. 10: Both the IP header and TCP checksum were checked and were correct. 11: Both the IP header and UDP checksum were checked and were correct.

Table 32-2. Receive Buffer Descriptor Entry (continued)

Bit	Function
21	VLAN tag detected—type ID of 0x8100. For packets incorporating the stacked VLAN processing feature, this bit will be set if the second VLAN tag has a type ID of 0x8100
20	Priority tag detected—type ID of 0x8100 and null VLAN identifier. For packets incorporating the stacked VLAN processing feature, this bit will be set if the second VLAN tag has a type ID of 0x8100 and a null VLAN identifier.
19:17	VLAN priority—only valid if bit 21 is set.
16	Canonical format indicator (CFI) bit (only valid if bit 21 is set).
15	End of frame—when set the buffer contains the end of a frame. If end of frame is not set, then the only valid status bit is start of frame (bit 14).
14	Start of frame—when set the buffer contains the start of a frame. If both bits 15 and 14 are set, the buffer contains a whole frame.
13	This bit has a different meaning depending on whether jumbo frames and ignore FCS modes are enabled. If neither mode is enabled this bit will be zero. With jumbo frame mode enabled: (bit 3 set in Network Configuration Register) Additional bit for length of frame (bit[13]), that is concatenated with bits[12:0] With ignore FCS mode enabled and jumbo frames disabled: (bit 26 set in Network Configuration Register and bit 3 clear in Network Configuration Register) This indicates per frame FCS status as follows: 0: Frame had good FCS 1: Frame had bad FCS, but was copied to memory as ignore FCS enabled.
12:0	These bits represent the length of the received frame which may or may not include FCS depending on whether FCS discard mode is enabled. With FCS discard mode disabled: (bit 17 clear in Network Configuration Register) Least significant 12 bits for length of frame including FCS. If jumbo frames are enabled, these 12 bits are concatenated with bit[13] of the descriptor above. With FCS discard mode enabled: (bit 17 set in Network Configuration Register) Least significant 12 bits for length of frame excluding FCS. If jumbo frames are enabled, these 12 bits are concatenated with bit[13] of the descriptor above.

Each receive AXI buffer start location is a word address. The start of the first AXI buffer in a frame can be offset by up to three Bytes, depending on the value written to bits 14 and 15 of the Network Configuration register (NCFGR). If the start location of the AXI buffer is offset, the available length of the first AXI buffer is reduced by the corresponding number of Bytes.

To receive frames, the AXI buffer descriptors must be initialized by writing an appropriate address to bits 31:2 in the first word of each list entry. Bit 0 must be written with zero. Bit 1 is the wrap bit and indicates the last entry in the buffer descriptor list.

The start location of the receive buffer descriptor list must be written with the receive buffer queue base address before reception is enabled (receive enable in the Network Control register NCR). Once reception is enabled, any writes to the Receive Buffer Queue Base Address register (RBQB) are ignored. When read, it will return the current pointer position in the descriptor list, though this is only valid and stable when receive is disabled.

If the filter block indicates that a frame should be copied to memory, the receive data DMA operation starts writing data into the receive buffer. If an error occurs, the buffer is recovered.

An internal counter within the ETH represents the receive buffer queue pointer and it is not visible through the CPU interface. The receive buffer queue pointer increments by two words after each buffer has been used. It re-initializes to the receive buffer queue base address if any descriptor has its wrap bit set.

As receive AXI buffers are used, the receive AXI buffer manager sets bit zero of the first word of the descriptor to logic one indicating the AXI buffer has been used.

Software should search through the “used” bits in the AXI buffer descriptors to find out how many frames have been received, checking the start of frame and end of frame bits.

When the DMA is configured in the packet buffer Partial Store And Forward mode, received frames are written out to the AXI buffers as soon as enough frame data exists in the packet buffer. For both cases, this may mean several full AXI buffers are used before some error conditions can be detected. If a receive error is detected the receive buffer currently being written will be recovered. Previous buffers will not be recovered. As an example, when receiving frames with cyclic redundancy check (CRC) errors or excessive length, it is possible that a frame fragment might be stored in a sequence of AXI receive buffers. Software can detect this by looking for start of frame bit set in a buffer following a buffer with no end of frame bit set.

To function properly, a 10/100 Ethernet system should have no excessive length frames or frames greater than 128 Bytes with CRC errors. Collision fragments will be less than 128 Bytes long, therefore it will be a rare occurrence to find a frame fragment in a receive AXI buffer, when using the default value of 128 Bytes for the receive buffers size.

When in packet buffer full store and forward mode, only good received frames are written out of the DMA, so no fragments will exist in the AXI buffers due to MAC receiver errors. There is still the possibility of fragments due to DMA errors, for example used bit read on the second buffer of a multi-buffer frame.

If bit zero of the receive buffer descriptor is already set when the receive buffer manager reads the location of the receive AXI buffer, the buffer has been already used and cannot be used again until software has processed the frame and cleared bit zero. In this case, the “buffer not available” bit in the receive status register is set and an interrupt triggered. The receive resource error statistics register is also incremented.

When the DMA is configured in the packet buffer full store and forward mode, the user can optionally select whether received frames should be automatically discarded when no AXI buffer resource is available. This feature is selected via the DMA Discard Receive Packets bit in the DMA Configuration register (DCFGR.DDRP). By default, the received frames are not automatically discarded. If this feature is off, then received packets will remain to be stored in the SRAM-based packet buffer until AXI buffer resource next becomes available. This may lead to an eventual packet buffer overflow if packets continue to be received when bit zero (used bit) of the receive buffer descriptor remains set.

Note: After a used bit has been read, the receive buffer manager will re-read the location of the receive buffer descriptor every time a new packet is received. When the DMA is not configured in the packet buffer full store and forward mode and a used bit is read, the frame currently being received will be automatically discarded.

When the DMA is configured in the packet buffer full store and forward mode, a receive overrun condition occurs when the receive SRAM-based packet buffer is full, or because HRESP was not OK. In all other modes, a receive overrun condition occurs when either the AXI bus was not granted quickly enough, or because HRESP was not OK, or because a new frame has been detected by the receive block, but the status update or write back for the previous frame has not yet finished. For a receive overrun condition, the receive overrun interrupt is asserted and the buffer currently being written is recovered. The next frame that is received whose address is recognized reuses the buffer.

In any packet buffer mode, writing a '1' to the Flush Next Package bit in the NCR register (NCR.FNP) will force a packet from the external SRAM-based receive packet buffer to be flushed. This feature is only acted upon when the RX DMA is not currently writing packet data out to AXI, i.e., it is in an IDLE state. If the RX DMA is active, NCR.FNP=1 is ignored.

32.6.3.4. Transmit AXI Buffers

Frames to transmit are stored in one or more transmit AXI buffers. Transmit frames can be between 1 and 16384 Bytes long, so it is possible to transmit frames longer than the maximum length specified in the IEEE 802.3 standard. It should be noted that zero length AXI buffers are allowed and that the maximum number of buffers permitted for each transmit frame is 128.

The start location for each transmit AXI buffer is stored in memory in a list of transmit buffer descriptors at a location pointed to by the transmit buffer queue pointer. The base address for this queue pointer is set in software using the Transmit Buffer Queue Base Address register. Each list entry consists of two words. The first is the Byte address of the transmit buffer and the second containing the transmit control and status. For the packet buffer DMA, the start location for each AXI buffer is a Byte address, the bottom bits of the address being used to offset the start of the data from the data-word boundary (i.e., bits 2,1 and 0 are used to offset the address for 64-bit data paths).

Frames can be transmitted with or without automatic Cyclic Redundancy Checksum (CRC) generation. If CRC is automatically generated, pad will also be automatically generated to take frames to a minimum length of 64 Bytes. When CRC is not automatically generated (as defined in word 1 of the transmit buffer descriptor), the frame is assumed to be at least 64 Bytes long and pad is not generated.

An entry in the transmit buffer descriptor list is described in this table:

Table 32-3. Transmit Buffer Descriptor Entry

Bit	Function
Word 0	
31:0	Byte address of buffer
Word 1	
31	Used—must be zero for the ETH to read data to the transmit buffer. The ETH sets this to one for the first buffer of a frame once it has been successfully transmitted. Software must clear this bit before the buffer can be used again.
30	Wrap—marks last descriptor in transmit buffer descriptor list. This can be set for any buffer within the frame.
29	Retry limit exceeded, transmit error detected
28	Transmit underrun—occurs when the start of packet data has been written into the FIFO and either HRESP is not OK, the transmit data could not be fetched in time, or when buffers are exhausted.
27	Transmit frame corruption due to AXI error—set if an error occurs while midway through reading transmit frame from the AXI, including HRESP errors and buffers exhausted mid frame (if the buffers run out during transmission of a frame then transmission stops, FCS shall be bad and TXER asserted). Also set if single frame is too large for configured packet buffer memory size.
26	Late collision, transmit error detected.
25:23	Reserved
22:20	Transmit IP/TCP/UDP checksum generation offload errors: 000: No Error. 001: The Packet was identified as a VLAN type, but the header was not fully complete, or had an error in it. 010: The Packet was identified as a SNAP type, but the header was not fully complete, or had an error in it. 011: The Packet was not of an IP type, or the IP packet was invalidly short, or the IP was not of type IPv4/IPv6. 100: The Packet was not identified as VLAN, SNAP or IP. 101: Non supported packet fragmentation occurred. For IPv4 packets, the IP checksum was generated and inserted. 110: Packet type detected was not TCP or UDP. TCP/UDP checksum was therefore not generated. For IPv4 packets, the IP checksum was generated and inserted. 111: A premature end of packet was detected and the TCP/UDP checksum could not be generated.
19:17	Reserved
16	No CRC to be appended by MAC. When set, this implies that the data in the buffers already contains a valid CRC, therefore no CRC or padding is to be appended to the current frame by the MAC. This control bit must be set for the first buffer in a frame and will be ignored for the subsequent buffers of a frame. Note that this bit must be clear when using the transmit IP/TCP/UDP checksum generation offload, otherwise checksum generation and substitution will not occur.
15	Last buffer, when set this bit will indicate the last buffer in the current frame has been reached.
14	Reserved
13:0	Length of buffer

To transmit frames, the buffer descriptors must be initialized by writing an appropriate Byte address to bits [31:0] of the first word of each descriptor list entry.

The second word of the transmit buffer descriptor is initialized with control information that indicates the length of the frame, whether or not the MAC is to append CRC and whether the buffer is the last buffer in the frame.

After transmission the status bits are written back to the second word of the first buffer along with the used bit. Bit 31 is the used bit which must be zero when the control word is read if transmission is to take place. It is written to '1' once the frame has been transmitted. Bits[29:20] indicate various transmit error conditions. Bit 30 is the wrap bit which can be set for any buffer within a frame. If no wrap bit is encountered the queue pointer continues to increment.

The Transmit Buffer Queue Base Address register can only be updated while transmission is disabled or halted; otherwise any attempted write will be ignored. When transmission is halted the transmit buffer queue pointer will maintain its value. Therefore when transmission is restarted the next descriptor read from the queue will be from immediately after the last successfully transmitted frame. As long as transmit is disabled by writing a '0' to the Transmit Enable bit in the Network Control register (NCR.TXEN), the transmit buffer queue pointer resets to point to the address indicated by the Transmit Buffer Queue Base Address register (TBQB).
Note: Disabling receive does not have the same effect on the receive buffer queue pointer.

Once the transmit queue is initialized, transmit is activated by writing a '1' to the Start Transmission bit of the Network Control register (NCR.TSTART). Transmit is halted when a buffer descriptor with its used bit set is read, a transmit error occurs, or by writing to the Transmit Halt bit of the Network Control register (NCR.THALT). Transmission is suspended if a pause frame is received while the Transmit Pause Frame bit is '1' in the Network Configuration register (NCR.TXPF). Rewriting the Start bit (NCR.TSTART) while transmission is active is allowed. This is implemented by the Transmit Go variable which is readable in the Transmit Status register (TSR.TXGO). The TXGO variable is reset when:

- Transmit is disabled
- A buffer descriptor with its ownership bit set is read
- Bit 10, THALT, of the Network Control register is written
- There is a transmit error such as too many retries, or a transmit underrun

To set TXGO, write a '1' to NCR.TSTART. Transmit halt does not take effect until any ongoing transmit finishes.

The DMA transmission will automatically restart from the first buffer of the frame.

If the DMA is configured for packet buffer Partial Store and Forward mode and a collision occurs during transmission of a multi-buffer frame, transmission will automatically restart from the first buffer of the frame. For packet buffer mode, the entire contents of the frame are read into the transmit packet buffer memory, so the retry attempt will be replayed directly from the packet buffer memory rather than having to re-fetch through the AXI.

If a used bit is read midway through transmission of a multi-buffer frame, this is treated as a transmit error. Transmission stops, TXER is asserted and the FCS will be bad.

If transmission stops due to a transmit error or a used bit being read, transmission restarts from the first buffer descriptor of the frame being transmitted when the transmit start bit is rewritten.

32.6.3.5. DMA Bursting on the AXI

The DMA will always use SINGLE, or INCR type AXI accesses for buffer management operations. When performing data transfers, the AXI burst length is selected by the Fixed Burst Length for DMA Data Operations bit field in the DMA Configuration register (DCFGR.FBLDO) so that either SINGLE or fixed length incrementing bursts (INCR4, INCR8 or INCR16) are used where possible:

When there is enough space and enough data to be transferred, the programmed fixed length bursts will be used. If there is not enough data or space available, for example when at the beginning or the end of a buffer, SINGLE type accesses are used. Also SINGLE type accesses are used at 1024 Byte boundaries, so that the 1 KByte boundaries are not burst over as per AXI requirements.

The DMA will not terminate a fixed length burst early, unless an error condition occurs on the AXI or if receive or transmit are disabled in the Network Control register (NCR).

32.6.3.6. DMA Packet Buffer

The DMA uses packet buffers for both transmit and receive paths. This mode allows multiple packets to be buffered in both transmit and receive directions. This allows the DMA to withstand far greater access latencies on the AXI and make more efficient use of the AXI bandwidth. There are two modes of operation—Full Store and Forward and Partial Store and Forward.

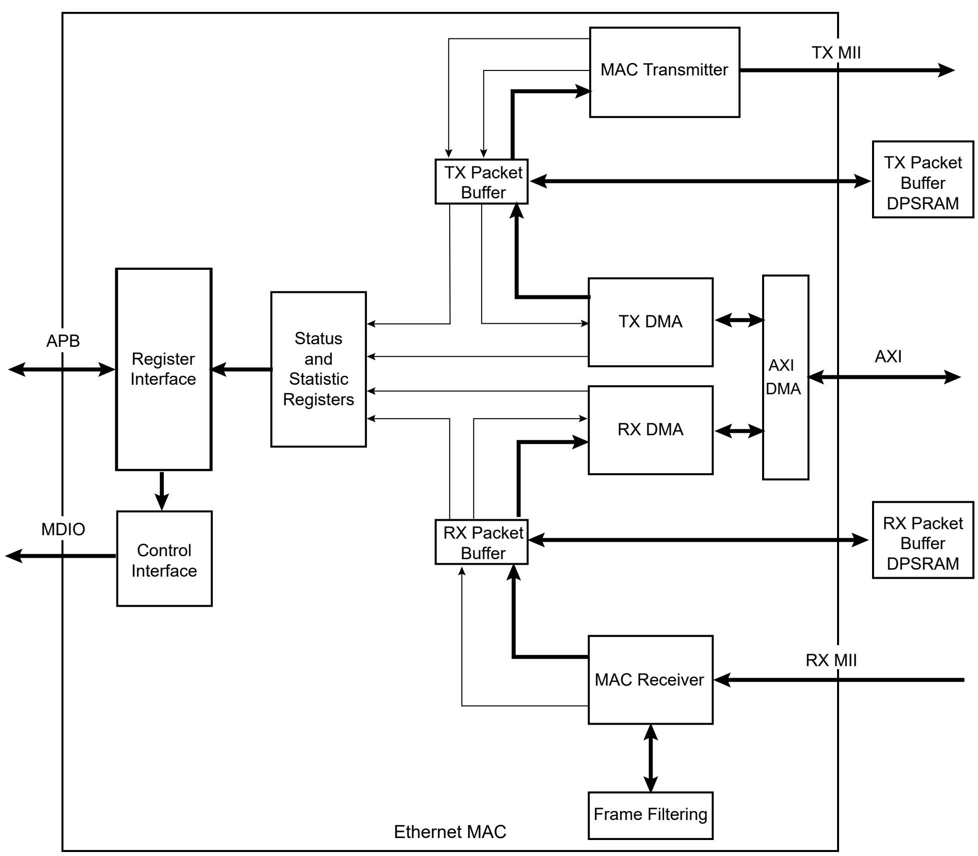
As described above, the DMA can be programmed into a low latency mode, known as Partial Store and Forward. For further details of this mode, see the related Links.

When the DMA is in full store and forward mode, full packets are buffered which provides the possibility to:

- Discard packets with error on the receive path before they are partially written out of the DMA, therefore saving AXI bus bandwidth and driver processing overhead,
- Retry collided transmit frames from the buffer, therefore saving AXI bus bandwidth,
- Implement transmit IP/TCP/UDP checksum generation offload.

With the packet buffers included, the structure of the ETH data paths is shown in this image:

Figure 32-2. Data Paths with Packet Buffers Included



32.6.3.7. Transmit Packet Buffer

The transmitter packet buffer will continue attempting to fetch frame data from the AXI system memory until the packet buffer itself is full, at which point it will attempt to maintain its full level.

To accommodate the status and statistics associated with each frame, three words per packet (or two if the ETH is configured in 64-bit data path mode) are reserved at the end of the packet data. If the packet is bad and requires to be dropped, the status and statistics are the only information held on that packet. Storing the status in the DPRAM is required in order to decouple the DMA interface of the buffer from the MAC interface, to update the MAC status/statistics and to generate interrupts in the order in which the packets that they represent were fetched from the AXI memory.

If any errors occur on the AXI while reading the transmit frame, the fetching of packet data from AXI memory is halted. The MAC transmitter will continue to fetch packet data, thereby emptying the packet buffer and allowing any good (non-erroneous) frames to be transmitted successfully. Once these have been fully transmitted, the status/statistics for the erroneous frame will be updated and software will be informed via an interrupt that an AXI error occurred. This way, the error is reported in the correct packet order.

The transmit packet buffer will only attempt to read more frame data from the AXI when space is available in the packet buffer memory. If space is not available it must wait until the a packet fetched by the MAC completes transmission and is subsequently removed from the packet buffer memory.
Note: If full store and forward mode is active and if a single frame is fetched that is too large for the packet buffer memory, the frame is flushed and the DMA halted with an error status. This is because a complete frame must be written into the packet buffer before transmission can begin, and therefore the minimum packet buffer memory size should be chosen to satisfy the maximum frame to be transmitted in the application.

In full store and forward mode, once the complete transmit frame is written into the packet buffer memory, a trigger is sent across to the MAC transmitter, which will then begin reading the frame from the packet buffer memory. Since the whole frame is present and stable in the packet buffer memory an underflow of the transmitter is not possible. The frame is kept in the packet buffer until notification is received from the MAC that the frame data has either been successfully transmitted or can no longer be retransmitted (too many retries in half duplex mode). When this notification is received the frame is flushed from memory to make room for a new frame to be fetched from AXI system memory.

In Partial Store and Forward mode, a trigger is sent across to the MAC transmitter as soon as sufficient packet data is available, which will then begin fetching the frame from the packet buffer memory. If, after this point, the MAC transmitter is able to fetch data from the packet buffer faster than the AXI DMA can fill it, an underflow of the transmitter is possible. In this case, the transmission is terminated early, and the packet buffer is completely flushed. Transmission can only be restarted by writing a '1' to the Transmit Start bit in the Network Control register (NCR.TSTART).

In half duplex mode, the frame is kept in the packet buffer until notification is received from the MAC that the frame data has either been successfully transmitted or can no longer be retransmitted (too many retries in half duplex mode). When this notification is received the frame is flushed from memory to make room for a new frame to be fetched from AXI system memory.

In full duplex mode, the frame is removed from the packet buffer on the fly.

Other than underflow, the only MAC related errors that can occur are due to collisions during half duplex transmissions. When a collision occurs the frame still exists in the packet buffer memory so can be retried directly from there. After sixteen failed transmit attempts, the frame will be flushed from the packet buffer.

32.6.3.8. Receive Packet Buffer

The receive packet buffer stores frames from the MAC receiver along with their status and statistics. Frames with errors are flushed from the packet buffer memory, while good frames are pushed onto the DMA AXI interface.

The receiver packet buffer monitors the FIFO write interface from the MAC receiver and translates the FIFO pushes into packet buffer writes. At the end of the received frame the status and statistics are buffered so that the information can be used when the frame is read out. When programmed in full store and forward mode and the frame has an error, the frame data is immediately flushed from the packet buffer memory allowing subsequent frames to utilize the freed up space. The status and statistics for bad frames are still used to update the ETH registers.

To accommodate the status and statistics associated with each frame, three words per packet (or two if configured in 64-bit datapath mode) are reserved at the end of the packet data. If the packet is bad and requires to be dropped, the status and statistics are the only information held on that packet.

The receiver packet buffer will also detect a full condition so that an overflow condition can be detected. If this occurs, subsequent packets are dropped and an RX overflow interrupt is raised.

For full store and forward, the DMA only begins packet fetches once the status and statistics for a frame are available. If the frame has a bad status due to a frame error, the status and statistics are passed on to the ETH registers. If the frame has a good status, the information is used to read the frame from the packet buffer memory and burst onto the AXI using the DMA buffer management protocol. Once the last frame data has been transferred to the packet buffer, the status and statistics are updated to the ETH registers.

If Partial Store and Forward mode is active, the DMA will begin fetching the packet data before the status is available. As soon as the status becomes available, the DMA will fetch this information as soon as possible before continuing to fetch the remainder of the frame. Once the last frame data has been transferred to the packet buffer, the status and statistics are updated to the ETH registers.

32.6.3.9. Priority Queuing in the DMA

The DMA by default uses a single transmit and receive queue. This means the list of transmit/receive buffer descriptors point to data buffers associated with a single transmit/receive data stream. The ETH can select up to 6 priority queues. Each queue has an independent list of buffer descriptors pointing to separate data streams.

The table below gives the DPRAM size associated with each queue.

Table 32-4. Queue Size

Queue Number	Queue Size
5 (highest priority)	1 KB
4	2 KB
3	2 KB
2	1 byte
1	1 byte
0 (lowest priority)	2 KB

In the transmit direction, higher priority queues are always serviced before lower priority queues, with Q0 as lowest priority and Q5 as highest priority. This strict priority scheme requires the user to ensure that high priority traffic is constrained so that lower priority traffic will have required bandwidth. The ETH DMA will determine the next queue to service by initiating a sequence of buffer descriptor reads interrogating the ownership bits of each. The buffer descriptor corresponding to the highest priority queue is read first.

As an example, if the ownership bit of this descriptor is set, the DMA will progress by reading the 2nd highest priority queue’s descriptor. If that ownership bit read of this lower priority queue is set as well, the DMA will read the 3rd highest priority queue’s descriptor. If all the descriptors return an ownership bit set, a resource error has occurred, so an interrupt is generated and transmission is automatically halted. Transmission can only be restarted by writing a '1' to the Transmission Start bit in the Network Control register (NCR.TSTART). The ETH DMA will need to identify the highest available queue to transmit from when the TSTART bit is written and the TX is in a halted state, or when the last word of any packet has been fetched from external AXI memory.

The ETH transmit DMA maximizes the effectiveness of priority queuing by ensuring that high priority traffic be transmitted as early as possible after being fetched from AXI. High priority traffic fetched from AXI will be pushed to the MAC layer, depending on traffic shaping being enabled and the associated credit value for that queue, before any lower priority traffic that may pre-exist in the transmit SRAM-based packet buffer. This is achieved by separating the transmit SRAM-based packet buffer into regions, one region per queue. The size of each region determines the amount of SRAM space allocated per queue.

For each queue, there is an associated Transmit Buffer Queue Base Address register (TBQB). For the lowest priority queue (or the only queue when only one queue is selected), the Transmit Buffer Queue Base Address is located at address 0x101C. For all other queues, the Transmit Buffer Queue Base Address registers are located at sequential addresses starting at address 0x1440.

In the receive direction each packet is written to AXI data buffers in the order that it is received. For each queue, there is an independent set of receive AXI buffers for each queue. There is therefore a separate Receive Buffer Queue Base Address register for each queue (RBPQB). For the lowest priority queue (or the only queue when only one queue is selected), the Receive Buffer Queue Base Address is located at address 0x1018. For all other queues, the Receive Buffer Queue Base Address registers are located at sequential addresses starting at address 0x1480. Every received packet will pass through a programmable screening algorithm which will allocate a particular queue to that frame. The user interface to the screeners is through two types of programmable registers:

- Screening Type 1 registers: The module features 4 Screening Type 1 registers. Screening Type 1 registers hold values to match against specific IP and UDP fields of the received frames. The fields matched against are DS (Differentiated Services field of IPv4 frames), TC (Traffic class field of IPv6 frames) and/or the UDP destination port.
- Screening Type 2 registers: The module features 8 Screening Type 2 registers SCRT2. Screening Type 2 registers operate independently of Screening Type 1 registers and offer additional match capabilities. Screening Type 2 allows a screen to be configured that is the combination of all or any of the following comparisons:
  - An enable bit VLAN priority, VLANE. A VLAN priority match will be performed if the VLAN priority enable is set. The extracted priority field in the VLAN header is compared against VLANP in the SCRT2 itself.
  - An enable bit EtherType, ETHE. The EtherType field I2ETH inside the SCRT2 maps to one of 4 EtherType match registers, SCRT2ET. The extracted EtherType is compared against SCRT2ET designated by this EtherType field.
  - An enable bit Compare A, COMPAE. This bit is associated with a Screening Type 2 Compare Word 0/1 register x, SCRT2CMP/1.
  - An enable bit Compare B, COMPBE. This bit is associated with a Screening Type 2 Compare Word 0/1 register x, SCRT2CMP/1.
  - An enable bit Compare C, COMPCE. This bit is associated with a Screening Type 2 Compare Word 0/1 register x, SCRT2CMP/1.

Each screener type has an enable bit, a match pattern and a queue number. If a received frame matches on an enabled screening register, then the frame will be tagged with the queue value in the associated screening register, and forwarded onto the DMA and subsequently into the external memory associated with that queue. If two screeners are matched then the one which resides at the lowest register address will take priority so care must be taken on the selection of the screener location.

When the priority queuing feature is enabled, the number of interrupt outputs from the ETH core is increased to match the number of supported queues. The number of Interrupt Status registers is increased by the same number. Only DMA related events are reported using the individual interrupt outputs, as the ETH can relate these events to specific queues. All other events generated within the ETH are reported in the interrupt associated with the lowest priority queue. For the lowest priority queue (or the only queue when only 1 queue is selected), the Interrupt Status register is located at address 0x1024. For all other queues, the Interrupt Status register is located at sequential addresses starting at address 0x1400.

Note: The address matching is the first level of filtering. If there is a match, the screeners are the next level of filtering for routing the data to the appropriate queue. See MAC Filtering Block for more details.

The additional screening done by the functions Compare A, B, and C each have an enable bit and compare register field. COMPA, COMPB and COMPC in SCRT2 are pointers to a configured offset (OFFSVAL), value (COMPVAL), and mask (MASKVAL). If enabled, the compare is true if the data at the offset into the frame, ANDed with MASKVAL, is equal to the value of COMPVAL ANDed with MASKVAL. A 16-bit word comparison is done. The byte at the offset number of bytes from the index start is compared to bits 7:0 of the configured COMPVAL and MASKVAL. The byte at the offset number of bytes + 1 from the index start is compared to bits 15:8 of the configured COMPVAL and MASKVAL.

The offset value in bytes, OFFSVAL, ranges from 0 to 127 bytes from either the start of the frame, the byte after the EtherType field, the byte after the IP header (IPv4 or IPv6) or the byte after the TCP/UDP header. Note the logic to decode the IP header or the TCP/UDP header is reused from the TCP/UDP/IP checksum offload logic and therefore has the same restrictions on use (the main limitation is that IP fragmentation is not supported). Refer to the Checksum Offload for IP, TCP and UDP section of this documentation for further details.

Compare A, B, and C use a common set of 24 SCRT2CMP/1 registers, therefore all COMPA, COMPB and COMPC fields in the registers SCRT2 point to a single pool of 24 SCRT2CMP/1 registers.

Note that Compare A, B and C together allow matching against an arbitrary 48 bits of data and so can be used to match against a MAC address.

All enabled comparisons are ANDed together to form the overall type 2 screening match.

32.6.4. MAC Transmit Block

The MAC transmitter can operate in either half duplex or full duplex mode and transmits frames in accordance with the Ethernet IEEE 802.3 standard. In half duplex mode, the CSMA/CD protocol of the IEEE 802.3 specification is followed.

A small input buffer receives data through the FIFO interface which, depending on the DMA bus width control bits in the Network Configuration register, will extract data in 32-bit or 64-bit form. All subsequent processing prior to the final output is performed in bytes.

Transmit data can be output using the MII interface.

Frame assembly starts by adding preamble and the start frame delimiter. Data is taken from the transmit FIFO interface a word at a time. In 10/100 mode, transmit data to the PHY is nibble wide and least significant nibble first using TXD[3:0] with TXD[7:4] tied to logic 0.

If necessary, padding is added to take the frame length to 60 bytes. CRC is calculated using an order 32-bit polynomial. This is inverted and appended to the end of the frame taking the frame length to a minimum of 64 bytes. If the no CRC bit is set in the second word of the last buffer descriptor of a transmit frame, neither pad nor CRC are appended. The no CRC bit can also be set through the FIFO interface.

In full duplex mode (at all data rates), frames are transmitted immediately. Back to back frames are transmitted at least 96 bit times apart to guarantee the interframe gap.

In half duplex mode, the transmitter checks carrier sense. If asserted, the transmitter waits for the signal to become inactive, and then starts transmission after the interframe gap of 96 bit times. If the collision signal is asserted during transmission, the transmitter will transmit a jam sequence of 32 bits taken from the data register and then retry transmission after the back off time has elapsed.

If the collision occurs during either the preamble or Start Frame Delimiter (SFD), then these fields will be completed prior to generation of the jam sequence.

The back off time is based on an XOR of the 10 least significant bits of the data coming from the transmit FIFO interface and a 10-bit pseudo random number generator. The number of bits used depends on the number of collisions seen. After the first collision 1 bit is used, then the second 2 bits and so on up to the maximum of 10 bits. All 10 bits are used above ten collisions. An error will be indicated and no further attempts will be made if 16 consecutive attempts cause collision. This operation is compliant with the description in Clause 4.2.3.2.5 of the IEEE 802.3 standard which refers to the truncated binary exponential back off algorithm.

In 10/100 mode, both collisions and late collisions are treated identically, and back off and retry will be performed up to 16 times. This condition is reported in the transmit buffer descriptor word 1 (late collision, bit 26) and also in the Transmit Status register (late collision, bit 7). An interrupt can also be generated (if enabled) when this exception occurs, and bit 5 in the Interrupt Status register will be set.

In all modes of operation, if the transmit DMA underruns, a bad CRC is automatically appended using the same mechanism as jam insertion and the TXER signal is asserted. For a properly configured system this should never happen and also it is impossible if configured to use the DMA with packet buffers, as the complete frame is buffered in local packet buffer memory.

By setting when bit 28 is set in the Network Configuration register, the Inter Packet Gap (IPG) may be stretched beyond 96 bits depending on the length of the previously transmitted frame and the value written to the IPG Stretch register (IPGS). The least significant 8 bits of the IPG Stretch register multiply the previous frame length (including preamble). The next significant 8 bits (+1 so as not to get a divide by zero) divide the frame length to generate the IPG. IPG stretch only works in full duplex mode and when bit 28 is set in the Network Configuration register. The IPG Stretch register cannot be used to shrink the IPG below 96 bits.

If the back pressure bit is set in the Network Control register, the transmit block transmits 64 bits of data, which can consist of 16 nibbles of 1011 or in bit rate mode 64 1s, whenever it sees an incoming frame to force a collision. This provides a way of implementing flow control in half duplex mode.

32.6.5. MAC Receive Block

All processing within the MAC receive block is implemented using a 16-bit data path. The MAC receive block checks for valid preamble, FCS, alignment and length, presents received frames to the FIFO interface and stores the frame destination address for use by the address checking block.

If, during the frame reception, the frame is found to be too long, a bad frame indication is sent to the FIFO interface. The receiver logic ceases to send data to memory as soon as this condition occurs.

At end of frame reception the receive block indicates to the DMA block whether the frame is good or bad. The DMA block will recover the current receive buffer if the frame was bad.

Ethernet frames are normally stored in DMA memory complete with the FCS. Setting the FCS remove bit in the network configuration (bit 17) causes frames to be stored without their corresponding FCS. The reported frame length field is reduced by four bytes to reflect this operation.

The receive block signals to the register block to increment the alignment, CRC (FCS), short frame, long frame, jabber or receive symbol errors when any of these exception conditions occur.

If bit 26 is set in the network configuration, CRC errors will be ignored and CRC errored frames will not be discarded, though the Frame Check Sequence Errors statistic register will still be incremented. Additionally, if not enabled for jumbo frames mode, then bit[13] of the receiver descriptor word 1 will be updated to indicate the FCS validity for the particular frame. This is useful for applications such as EtherCAT whereby individual frames with FCS errors must be identified.

Received frames can be checked for length field error by setting the length field error frame discard bit of the Network Configuration register (bit-16). When this bit is set, the receiver compares a frame's measured length with the length field (bytes 13 and 14) extracted from the frame. The frame is discarded if the measured length is shorter. This checking procedure is for received frames between 64 bytes and 1518 bytes in length.

Each discarded frame is counted in the 10-bit length field error statistics register. Frames where the length field is greater than or equal to 0x0600 hex will not be checked.

32.6.6. Checksum Offload for IP, TCP and UDP

The ETH can be programmed to perform IP, TCP and UDP checksum offloading in both receive and transmit directions, which is enabled by setting bit 24 in the Network Configuration register (NCFGR.RXCOEN =1) for receive and bit 11 in the DMA Configuration register (DCFGR.TXCOEN =1) for transmit.

IPv4 packets contain a 16-bit checksum field, which is the 16-bit 1’s complement of the 1’s complement sum of all 16-bit words in the header. TCP and UDP packets contain a 16-bit checksum field, which is the 16-bit 1’s complement of the 1’s complement sum of all 16-bit words in the header, the data and a conceptual IP pseudo header.

To calculate these checksums in software requires each byte of the packet to be processed. For TCP and UDP this can use a large amount of processing power. Offloading the checksum calculation to hardware can result in significant performance improvements.

For IP, TCP or UDP checksum offload to be useful, the operating system containing the protocol stack must be aware that this offload is available so that it can make use of the fact that the hardware can either generate or verify the checksum.

32.6.6.1. Receiver Checksum Offload

When receive checksum offloading is enabled in the ETH Network Configuration Register (NCFGR.RXCOEN), the IPv4 header checksum is checked as per RFC 791, where the packet meets the following criteria:

- If present, the VLAN header must be four octets long and the CFI bit must not be set
- Encapsulation must be RFC 894 Ethernet Type Encoding or RFC 1042 SNAP Encoding
- IPv4 packet
- IP header is of a valid length

The ETH also checks the TCP checksum as per RFC 793, or the UDP checksum as per RFC 768, if the following criteria are met:

- IPv4 or IPv6 packet
- Good IP header checksum (if IPv4)
- No IP fragmentation
- TCP or UDP packet

When an IP, TCP or UDP frame is received, the receive buffer descriptor gives an indication if the ETH was able to verify the checksums. There is also an indication if the frame had SNAP encapsulation. These indication bits will replace the type ID match indication bits when the receive checksum offload is enabled. For details of these indication bits refer to “Receive Buffer Description Entry”.

If any of the checksums are verified as incorrect by the ETH, the packet is discarded and the appropriate statistics counter incremented.

32.6.6.2. Transmitter Checksum Offload

The transmitter checksum offload is only available if the full store and forward mode is enabled. This is because the complete frame to be transmitted must be read into the packet buffer memory before the checksum can be calculated and written back into the headers at the beginning of the frame.

Transmitter checksum offload is enabled by setting bit [11] in the DMA Configuration register (DCFGR.TXCOEN =1). When enabled, it will monitor the frame as it is written into the transmitter packet buffer memory to automatically detect the protocol of the frame. Protocol support is identical to the receiver checksum offload.

For transmit checksum generation and substitution to occur, the protocol of the frame must be recognized and the frame must be provided without the FCS field, by making sure that bit [16] of the transmit descriptor word 1 is clear. If the frame data already had the FCS field, this would be corrupted by the substitution of the new checksum fields.

If these conditions are met, the transmit checksum offload engine will calculate the IP, TCP and UDP checksums as appropriate. Once the full packet is completely written into packet buffer memory, the checksums will be valid and the relevant DPRAM locations will be updated for the new checksum fields as per standard IP/TCP and UDP packet structures.

If the transmitter checksum engine is prevented from generating the relevant checksums, bits [22:20] of the transmitter DMA writeback status will be updated to identify the reason for the error. Note that the frame will still be transmitted but without the checksum substitution, as typically the reason that the substitution did not occur was that the protocol was not recognized.

32.6.7. MAC Filtering Block

The filter block determines which frames should be written to the FIFO interface and on to the DMA.

Whether a frame is passed depends on what is enabled in the Network Configuration register, the state of the external matching pins, the contents of the specific address, type and Hash registers and the frame's destination address and type field.

If bit 25 of the Network Configuration register (NCFGR.EFRHD) is not set, a frame will not be copied to memory if the ETH is transmitting in half duplex mode at the time a destination address is received.

Ethernet frames are transmitted a byte at a time, least significant bit first. The first six bytes (48 bits) of an Ethernet frame make up the destination address. The first bit of the destination address, which is the LSB of the first byte of the frame, is the group or individual bit. This is one for multicast addresses and zero for unicast. The all ones address is the broadcast address and a special case of multicast.

The ETH supports recognition of four specific addresses. Each specific address requires two registers, Specific Address register Bottom and Specific Address register Top. Specific Address register Bottom stores the first four bytes of the destination address and Specific Address register Top contains the last two bytes. The addresses stored can be specific, group, local or universal.

The destination address of received frames is compared against the data stored in the Specific Address registers once they have been activated. The addresses are deactivated at reset or when their corresponding Specific Address register Bottom is written. They are activated when Specific Address register Top is written. If a receive frame address matches an active address, the frame is written to the FIFO interface and on to DMA memory.

Frames may be filtered using the type ID field for matching. Four type ID registers exist in the register address space and each can be enabled for matching by writing a one to the MSB (bit 31) of the respective register. When a frame is received, the matching is implemented as an OR function of the various types of match.

The contents of each type ID register (when enabled) are compared against the length/type ID of the frame being received (e.g., bytes 13 and 14 in non-VLAN and non-SNAP encapsulated frames) and copied to memory if a match is found. The encoded type ID match bits (Word 0, Bit 22 and Bit 23) in the receive buffer descriptor status are set indicating which type ID register generated the match, if the receive checksum offload is disabled.

The reset state of the type ID registers is zero, therefore each is initially disabled.

The following example illustrates the use of the address and type ID match registers for a MAC address of 21:43:65:87:A9:CB:

Preamble	55
SFD	D5
DA (Octet 0 - LSB)	21
DA (Octet 1)	43
DA (Octet 2)	65
DA (Octet 3)	87
DA (Octet 4)	A9
DA (Octet 5 - MSB)	CB
SA (LSB)	00 (see **Note**)
SA	00(see **Note**)
SA	00(see **Note**)
SA	00(see **Note**)
SA	00(see **Note**)
SA (MSB)	00(see **Note**)
Type ID (MSB)	43
Type ID (LSB)	21

Note: Contains the address of the transmitting device.

The previous sequence shows the beginning of an Ethernet frame. Byte order of transmission is from top to bottom, as shown. For a successful match to specific address 1, the following address matching registers must be set up:

Specific Address 1 Bottom register (SAB1) (Address 0x088) 0x87654321

Specific Address 1 Top register (SAT1) (Address 0x08C) 0x0000CBA9

For a successful match to the type ID, the following Type ID Match 1 register must be set up:

Type ID Match 1 register (TIDM1) (Address 0x0A8) 0x80004321

32.6.8. Broadcast Address

Frames with the broadcast address of 0xFFFFFFFFFFFF are stored to memory only if the 'no broadcast' bit in the Network Configuration register is set to zero.

32.6.9. Hash Addressing

The hash address register is 64 bits long and takes up two locations in the memory map. The least significant bits are stored in Hash Register Bottom and the most significant bits in Hash Register Top.

The unicast hash enable and the multicast hash enable bits in the Network Configuration register enable the reception of hash matched frames. The destination address is reduced to a 6-bit index into the 64-bit Hash register using the following hash function: The hash function is an XOR of every sixth bit of the destination address.

hash_index[05] = da[05] ^ da[11] ^ da[17] ^ da[23] ^ da[29] ^ da[35] ^ da[41] ^ da[47]

hash_index[04] = da[04] ^ da[10] ^ da[16] ^ da[22] ^ da[28] ^ da[34] ^ da[40] ^ da[46]

hash_index[03] = da[03] ^ da[09] ^ da[15] ^ da[21] ^ da[27] ^ da[33] ^ da[39] ^ da[45] hash_index[02] = da[02] ^ da[08] ^ da[14] ^ da[20] ^ da[26] ^ da[32] ^ da[38] ^ da[44]

hash_index[01] = da[01] ^ da[07] ^ da[13] ^ da[19] ^ da[25] ^ da[31] ^ da[37] ^ da[43]

hash_index[00] = da[00] ^ da[06] ^ da[12] ^ da[18] ^ da[24] ^ da[30] ^ da[36] ^ da[42]

da[0] represents the least significant bit of the first byte received, that is, the multicast/unicast indicator, and da[47] represents the most significant bit of the last byte received.

If the hash index points to a bit that is set in the Hash register then the frame will be matched according to whether the frame is multicast or unicast.

A multicast match will be signaled if the multicast hash enable bit is set, da[0] is logic 1 and the hash index points to a bit set in the Hash register.

A unicast match will be signaled if the unicast hash enable bit is set, da[0] is logic 0 and the hash index points to a bit set in the Hash register.

To receive all multicast frames, the Hash register should be set with all ones and the multicast hash enable bit should be set in the Network Configuration register.

32.6.10. Copy all Frames (Promiscuous Mode)

If the Copy All Frames bit is set in the Network Configuration register then all frames (except those that are too long, too short, have FCS errors or have RXER asserted during reception) will be copied to memory. Frames with FCS errors will be copied if bit 26 is set in the Network Configuration register.

32.6.11. Disable Copy of Pause Frames

Pause frames can be prevented from being written to memory by setting the disable copying of pause frames control bit 23 in the Network Configuration register. When set, pause frames are not copied to memory regardless of the Copy All Frames bit, whether a hash match is found, a type ID match is identified or if a destination address match is found.

32.6.12. VLAN Support

The following table describes an Ethernet encoded 802.1Q VLAN tag.

Table 32-5. 802.1Q VLAN Tag

TPID (Tag Protocol Identifier) 16 bits	TCI (Tag Control Information) 16 bits
0x8100	First 3 bits priority, then CFI bit, last 12 bits VID

The VLAN tag is inserted at the 13th byte of the frame adding an extra four bytes to the frame. To support these extra four bytes, the ETH can accept frame lengths up to 1536 bytes by setting bit 8 in the Network Configuration register.

If the VID (VLAN identifier) is null (0x000) this indicates a priority-tagged frame.

The following bits in the receive buffer descriptor status word give information about VLAN tagged frames:

- Bit 21 set if receive frame is VLAN tagged (i.e., type ID of 0x8100)
- Bit 20 set if receive frame is priority tagged (i.e., type ID of 0x8100 and null VID) (If bit 20 is set, bit 21 will be set also)
- Bit 19, 18 and 17 set to priority if bit 21 is set
- Bit 16 set to CFI if bit 21 is set

The ETH can be configured to reject all frames except VLAN tagged frames by setting the discard non-VLAN frames bit in the Network Configuration register.

32.6.13. Wake on LAN Support

The receive block supports Wake on LAN by detecting the following events on incoming receive frames:

- Magic packet
- Address Resolution Protocol (ARP) request to the device IP address
- Specific address 1 filter match
- Multicast hash filter match

These events can be individually enabled through bits [19:16] of the Wake on LAN register. Also, for Wake on LAN detection to occur, receive enable must be set in the Network Control register, however a receive buffer does not have to be available.

In case of an ARP request, specific address 1 or multicast filter events will occur even if the frame is errored. For magic packet events, the frame must be correctly formed and error free.

A magic packet event is detected if all of the following are true:

- Magic packet events are enabled through bit 16 of the Wake on LAN register
- The frame's destination address matches specific address 1
- The frame is correctly formed with no errors
- The frame contains at least 6 bytes of 0xFF for synchronization
- There are 16 repetitions of the contents of Specific Address 1 register immediately following the synchronization

An ARP request event is detected if all of the following are true:

- ARP request events are enabled through bit 17 of the Wake on LAN register
- Broadcasts are allowed by bit 5 in the Network Configuration register
- The frame has a broadcast destination address (bytes 1 to 6)
- The frame has a type ID field of 0x0806 (bytes 13 and 14)
- The frame has an ARP operation field of 0x0001 (bytes 21 and 22)
- The least significant 16 bits of the frame's ARP target protocol address (bytes 41 and 42) match the value programmed in bits[15:0] of the Wake on LAN register

The decoding of the ARP fields adjusts automatically if a VLAN tag is detected within the frame. The reserved value of 0x0000 for the Wake on LAN target address value will not cause an ARP request event, even if matched by the frame.

A specific address 1 filter match event will occur if all of the following are true:

- Specific address 1 events are enabled through bit 18 of the Wake on LAN register
- The frame's destination address matches the value programmed in the Specific Address 1 registers

A multicast filter match event will occur if all of the following are true:

- Multicast hash events are enabled through bit 19 of the Wake on LAN register
- Multicast hash filtering is enabled through bit 6 of the Network Configuration register
- The frame destination address matches against the multicast hash filter
- The frame destination address is not a broadcast

32.6.14. IEEE 1588 Support

IEEE 1588 is a standard for precision time synchronization in local area networks. It works with the exchange of special Precision Time Protocol (PTP) frames. The PTP messages can be transported over IEEE 802.3/Ethernet, over Internet Protocol Version 4 or over Internet Protocol Version 6 as described in the annex of IEEE P1588.D2.1.

The ETH indicates the message time-stamp point (asserted on the start packet delimiter and de-asserted at end of frame) for all frames and the passage of PTP event frames (asserted when a PTP event frame is detected and de-asserted at end of frame).

IEEE 802.1AS is a subset of IEEE 1588. One difference is that IEEE 802.1AS uses the Ethernet multicast address 0180C200000E for sync frame recognition whereas IEEE 1588 does not. ETH is designed to recognize sync frames with both IEEE 802.1AS and IEEE 1588 addresses and so can support both 1588 and 802.1AS frame recognition simultaneously.

Synchronization between host and client clocks is a two stage process.

First, the offset between the host and client clocks is corrected by the host sending a sync frame to the client with a follow up frame containing the exact time the sync frame was sent. Hardware assist modules at the host and client side detect exactly when the sync frame was sent by the host and received by the client. The client then corrects its clock to match the host clock.

Second, the transmission delay between the host and client is corrected. The client sends a delay request frame to the host which sends a delay response frame in reply. Hardware assist modules at the host and client side detect exactly when the delay request frame was sent by the client and received by the host. The client will now have enough information to adjust its clock to account for delay. For example, if the client was assuming zero delay, the actual delay will be half the difference between the transmit and receive time of the delay request frame (assuming equal transmit and receive times) because the client clock will be lagging the host clock by the delay time already.

The time-stamp is taken when the message time-stamp point passes the clock time-stamp point. This can generate an interrupt if enabled (IER). However, MAC Filtering configuration is needed to actually 'copy' the message to memory. For Ethernet, the message time-stamp point is the SFD and the clock time-stamp point is the MII interface. (The IEEE 1588 specification refers to sync and delay_req messages as event messages as these require time-stamping. These events are captured in the registers TSSx, EFTx, and EFRx, respectively. Follow up, delay response and management messages do not require time-stamping and are referred to as general messages.)

1588 version 2 defines two additional PTP event messages. These are the peer delay request (Pdelay_Req) and peer delay response (Pdelay_Resp) messages. These events are captured in the registers PEFTx and PEFRx, respectively. These messages are used to calculate the delay on a link. Nodes at both ends of a link send both types of frames (regardless of whether they contain a host or client clock). The Pdelay_Resp message contains the time at which a Pdelay_Req was received and is itself an event message. The time at which a Pdelay_Resp message is received is returned in a Pdelay_Resp_Follow_Up message.

1588 version 2 introduces transparent clocks of which there are two kinds, peer-to-peer (P2P) and end-to-end (E2E). Transparent clocks measure the transit time of event messages through a bridge and amend a correction field within the message to allow for the transit time. P2P transparent clocks additionally correct for the delay in the receive path of the link using the information gathered from the peer delay frames. With P2P transparent clocks delay_req messages are not used to measure link delay. This simplifies the protocol and makes larger systems more stable.

The ETH recognizes four different encapsulations for PTP event messages:

1. 1588 version 1 (UDP/IPv4 multicast)
2. 1588 version 2 (UDP/IPv4 multicast)
3. 1588 version 2 (UDP/IPv6 multicast)
4. 1588 version 2 (Ethernet multicast)

Table 32-6. Example of Sync Frame in 1588 Version 1 Format

Frame Segment	Value
Preamble/SFD	55555555555555D5
DA (Octets 0–5)	—
SA (Octets 6–11)	—
Type (Octets 12–13)	0800
IP stuff (Octets 14–22)	—
UDP (Octet 23)	11
IP stuff (Octets 24–29)	—
IP DA (Octets 30–32)	E00001
IP DA (Octet 33)	81 or 82 or 83 or 84
Source IP port (Octets 34–35)	—
Dest IP port (Octets 36–37)	013F
Other stuff (Octets 38–42)	—
Version PTP (Octet 43)	01
Other stuff (Octets 44–73)	—
Control (Octet 74)	00
Other stuff (Octets 75–168)	—

Table 32-7. Example of Delay Request Frame in 1588 Version 1 Format

Frame Segment	Value
Preamble/SFD	55555555555555D5
DA (Octets 0–5)	—
SA (Octets 6–11)	—
Type (Octets 12–13)	0800
IP stuff (Octets 14–22)	—
UDP (Octet 23)	11
IP stuff (Octets 24–29)	—
IP DA (Octets 30–32)	E00001
IP DA (Octet 33)	81 or 82 or 83 or 84
Source IP port (Octets 34–35)	—
Dest IP port (Octets 36–37)	013F
Other stuff (Octets 38–42)	—
Version PTP (Octet 43)	01
Other stuff (Octets 44–73)	—
Control (Octet 74)	01
Other stuff (Octets 75–168)	—

For 1588 version 1 messages, sync and delay request frames are indicated by the ETH if the frame type field indicates TCP/IP, UDP protocol is indicated, the destination IP address is 224.0.1.129/130/131 or 132, the destination UDP port is 319 and the control field is correct.

The control field is 0x00 for sync frames and 0x01 for delay request frames.

For 1588 version 2 messages, the type of frame is determined by looking at the message type field in the first byte of the PTP frame. Whether a frame is version 1 or version 2 can be determined by looking at the version PTP field in the second byte of both version 1 and version 2 PTP frames.

In version 2 messages sync frames have a message type value of 0x0, delay_req have 0x1, Pdelay_Req have 0x2 and Pdelay_Resp have 0x3.

Table 32-8. Example of Sync Frame in 1588 Version 2 (UDP/IPv4) Format

Frame Segment	Value
Preamble/SFD	55555555555555D5
DA (Octets 0–5)	—
SA (Octets 6–11)	—
Type (Octets 12–13)	0800
IP stuff (Octets 14–22)	—
UDP (Octet 23)	11
IP stuff (Octets 24–29)	—
IP DA (Octets 30–33)	E0000181
Source IP port (Octets 34–35)	—
Dest IP port (Octets 36–37)	013F
Other stuff (Octets 38–41)	—
Message type (Octet 42)	00
Version PTP (Octet 43)	02

Table 32-9. Example of Pdelay_Req Frame in 1588 Version 2 (UDP/IPv4) Format

Frame Segment	Value
Preamble/SFD	55555555555555D5
DA (Octets 0–5)	—
SA (Octets 6–11)	—
Type (Octets 12–13)	0800
IP stuff (Octets 14–22)	—
UDP (Octet 23)	11
IP stuff (Octets 24–29)	—
IP DA (Octets 30–33)	E000006B
Source IP port (Octets 34–35)	—
Dest IP port (Octets 36–37)	013F
Other stuff (Octets 38–41)	—
Message type (Octet 42)	02
Version PTP (Octet 43)	02

Table 32-10. Example of Sync Frame in 1588 Version 2 (UDP/IPv6) Format

Frame Segment	Value
Preamble/SFD	55555555555555D5
DA (Octets 0–5)	—
SA (Octets 6–11)	—
Type (Octets 12–13)	86dd
IP stuff (Octets 14–19)	—
UDP (Octet 20)	11
IP stuff (Octets 21–37)	—
IP DA (Octets 38–53)	FF0X00000000018
Source IP port (Octets 54–55)	—
Dest IP port (Octets 56–57)	013F
Other stuff (Octets 58–61)	—
Message type (Octet 62)	00
Other stuff (Octets 63–93)	—
Version PTP (Octet 94)	02

Table 32-11. Example of Pdelay_Resp Frame in 1588 Version 2 (UDP/IPv6) Format

Frame Segment	Value
Preamble/SFD	55555555555555D5
DA (Octets 0–5)	—
SA (Octets 6–11)	—
Type (Octets 12–13)	86dd
IP stuff (Octets 14–19)	—
UDP (Octet 20)	11
IP stuff (Octets 21–37)	—
IP DA (Octets 38–53)	FF0200000000006B
Source IP port (Octets 54–55)	—
Dest IP port (Octets 56–57)	013F
Other stuff (Octets 58–61)	—
Message type (Octet 62)	03
Other stuff (Octets 63–93)	—
Version PTP (Octet 94)	02

For the multicast address 011B19000000 sync and delay request frames are recognized depending on the message type field, 00 for sync and 01 for delay request.

Table 32-12. Example of Sync Frame in 1588 Version 2 (Ethernet Multicast) Format

Frame Segment	Value
Preamble/SFD	55555555555555D5
DA (Octets 0–5)	011B19000000
SA (Octets 6–11)	—
Type (Octets 12–13)	88F7
Message type (Octet 14)	00
Version PTP (Octet 15)	02

Pdelay request frames need a special multicast address so they can pass through ports blocked by the spanning tree protocol. For the multicast address 0180C200000E sync, Pdelay_Req and Pdelay_Resp frames are recognized depending on the message type field, 00 for sync, 02 for pdelay request and 03 for pdelay response.

Table 32-13. Example of Pdelay_Req Frame in 1588 Version 2 (Ethernet Multicast) Format

Frame Segment	Value
Preamble/SFD	55555555555555D5
DA (Octets 0–5)	0180C200000E
SA (Octets 6–11)	—
Type (Octets 12–13)	88F7
Message type (Octet 14)	00
Version PTP (Octet 15)	02

32.6.15. Time Stamp Unit

Overview

The TSU consists of a timer and registers to capture the time at which PTP event frames cross the message timestamp point. An interrupt is issued when a capture register is updated.

The 1588 time stamp unit (TSU) is implemented as a 94-bit timer.

- The 48 upper bits [93:46] of the timer count seconds and are accessible in the ETH 1588 Timer Seconds High Register" (TSH) and ETH 1588 Timer Seconds Low Register (TSL)
- The 30 lower bits [45:16] of the timer count nanoseconds and are accessible in the ETH 1588 Timer Nanoseconds Register (TN)
- The lowest 16 bits [15:0] of the timer count sub-nanoseconds

The 46 lower bits roll over when they have counted to 1s. An interrupt is generated when the seconds increment. The timer increments by a programmable period (to approximately 15.2fs resolution) with each MCK period. The timer value can be read, written and adjusted with 1ns resolution (incremented or decremented) through the APB interface.

Timer Adjustment

The amount by which the timer increments each clock cycle is controlled by the Timer Increment register (TI). Bits [7:0] are the default increment value in nanoseconds. Additional 16 bits of sub-nanosecond resolution are available using the Timer Increment Sub-Nanoseconds register (TISUBN). If the rest of the register is written with zero, the timer increments by the value in [7:0], plus the value of the TISUBN for each clock cycle.

The TISUBN allows a resolution of approximately 15fs.

Bits [15:8] of the increment register are the alternative increment value in nanoseconds, and bits [23:16] are the number of increments after which the alternative increment value is used. If [23:16] are zero the alternative increment value will never be used.

> Taking the example of 10.2MHz, there are 102 cycles every 10µs or 51 cycles every 5µs. So a timer with a 10.2MHz clock source is constructed by incrementing by 98ns for fifty cycles and then incrementing by 100ns (98ns × 50 + 100ns = 5000ns). This is programmed by writing the value 0x00326462 to the Timer Increment register (TI).

> In a second example, a 49.8 MHz clock source requires 20ns for 248 cycles, followed by an increment of 40ns (20ns × 248 + 40ns = 5000ns). This is programmed by writing the value 0x00F82814 to the TI register.

The Number of Increments bit field in the TI register is 8 bit in size, so frequencies up to 50 MHz are supported with 200 kHz resolution.

Without the alternative increment field the period of the clock would be limited to an integer number of nanoseconds, resulting in supported clock frequencies of 8, 10, 20, 25, 40, 50, 100, 125, 200 and 250 MHz.

There are eight additional 80-bit registers that capture the time at which PTP event frames are transmitted and received. An interrupt is issued when these registers are updated. The TSU timer count value can be compared to a programmable comparison value. For the comparison, the 48 bits of the seconds value and the upper 22 bits of the nanoseconds value are used. A signal (TSUCOMP) is output from the core to indicate when the TSU timer count value is equal to the comparison value stored in the TSU timer comparison value registers (NSC, SCL, and SCH). An interrupt can also be generated (if enabled) when the TSU timer count value and comparison value are equal, mapped to bit 29 of the interrupt status register.

32.6.16. MAC 802.3 Pause Frame Support

Note: Refer to the Clause 31, and Annex 31A and 31B of the IEEE standard 802.3 for a full description of MAC 802.3 pause operation.

The following table shows the start of a MAC 802.3 pause frame.

Table 32-14. Start of an 802.3 Pause Frame

Address		Type (MAC Control Frame)	Pause	
Destination	Source		Opcode	Time
0x0180C2000001	6 bytes	0x8808	0x0001	2 bytes

The ETH supports both hardware controlled pause of the transmitter, upon reception of a pause frame, and hardware generated pause frame transmission.

32.6.16.1. 802.3 Pause Frame Reception

Bit 13 of the Network Configuration register is the pause enable control for reception. If this bit is set, transmission will pause if a non zero pause quantum frame is received.

If a valid pause frame is received then the Pause Time register is updated with the new frame's pause time, regardless of whether a previous pause frame is active or not. An interrupt (either bit 12 or bit 13 of the Interrupt Status register) is triggered when a pause frame is received, but only if the interrupt has been enabled (bit 12 and bit 13 of the Interrupt Mask register). Pause frames received with non zero quantum are indicated through the interrupt bit 12 of the Interrupt Status register. Pause frames received with zero quantum are indicated on bit 13 of the Interrupt Status register.

Once the Pause Time register is loaded and the frame currently being transmitted has been sent, no new frames are transmitted until the pause time reaches zero. The loading of a new pause time, and therefore the pausing of transmission, only occurs when the ETH is configured for full duplex operation. If the ETH is configured for half duplex there will be no transmission pause, but the pause frame received interrupt will still be triggered. A valid pause frame is defined as having a destination address that matches either the address stored in Specific Address register 1 or if it matches the reserved address of 0x0180C2000001. It must also have the MAC control frame type ID of 0x8808 and have the pause opcode of 0x0001.

Pause frames that have frame check sequence (FCS) or other errors will be treated as invalid and will be discarded. 802.3 Pause frames that are received after Priority-based Flow Control (PFC) has been negotiated will also be discarded. Valid pause frames received will increment the pause frames received statistic register.

The pause time register decrements every 512 bit times once transmission has stopped. For test purposes, the retry test bit can be set (bit 12 in the Network Configuration register) which causes the Pause Time register to decrement every TXCK cycle once transmission has stopped.

The interrupt (bit 13 in the Interrupt Status register) is asserted whenever the Pause Time register decrements to zero (assuming it has been enabled by bit 13 in the Interrupt Mask register). This interrupt is also set when a zero quantum pause frame is received.

32.6.16.2. 802.3 Pause Frame Transmission

Automatic transmission of pause frames is supported through the transmit pause frame bits of the Network Control register. If either bit 11 (NCR.TXPF) or bit 12 (NCR.TXZQPF) of the Network Control register is written with logic 1, an 802.3 pause frame will be transmitted, providing full duplex is selected in the Network Configuration register and the transmit block is enabled in the Network Control register.

Pause frame transmission will happen immediately if transmit is inactive or if transmit is active between the current frame and the next frame due to be transmitted.

Transmitted pause frames comprise the following:

- A destination address of 01-80-C2-00-00-01
- A source address taken from Specific Address register 1
- A type ID of 88-08 (MAC control frame)
- A pause opcode of 00-01

- A pause quantum register
- Fill of 00 to take the frame to minimum frame length
- Valid FCS

The pause quantum used in the generated frame will depend on the trigger source for the frame as follows:

- If bit 11 (NCR.TXPF) is written with a '1', the pause quantum will be taken from the Transmit Pause Quantum register. The Transmit Pause Quantum register resets to a value of 0xFFFF giving maximum pause quantum as default.
- If bit 12 (NCR.TXZQPF) is written with a '1', the pause quantum will be zero.

After transmission, a pause frame transmitted interrupt will be generated (bit 14 (ISR.PFTR) of the Interrupt Status register) and the only statistics register that will be incremented will be the Pause Frames Transmitted register.

Pause frames can also be transmitted by the MAC using normal frame transmission methods.

32.6.17. MAC PFC Priority-based Pause Frame Support

Note: Refer to the 802.1Qbb standard for a full description of priority-based pause operation.

The following table shows the start of a Priority-based Flow Control (PFC) pause frame.

Table 32-15. Start of a PFC Pause Frame

Address		Type (Mac Control Frame)	Pause Opcode	Priority Enable Vector	Pause Time
Destination	Source				
0x0180C2000001	6 bytes	0x8808	0x1001	2 bytes	8 × 2 bytes

The ETH supports PFC priority-based pause transmission and reception. Before PFC pause frames can be received, bit 16 of the Network Control register must be set (NCR.ENPBPR =1).

32.6.17.1. PFC Pause Frame Reception

The ability to receive and decode priority-based pause frames is enabled by setting bit 16 (NCR.ENPBPR) of the Network Control register. When this bit is set, the ETH will match either classic 802.3 pause frames or PFC priority-based pause frames. Once a priority-based pause frame has been received and matched, then from that moment on the ETH will only match on priority-based pause frames (this is an 802.1Qbb requirement, known as PFC negotiation). Once priority-based pause has been negotiated, any received 802.3x format pause frames will not be acted upon.

If a valid priority-based pause frame is received then the ETH will decode the frame and determine which, if any, of the eight priorities require to be paused. Up to eight Pause Time registers are then updated with the eight pause times extracted from the frame regardless of whether a previous pause operation is active or not. An interrupt (either bit 12 or bit 13 of the Interrupt Status register) is triggered when a pause frame is received, but only if the interrupt has been enabled (bit 12 and bit 13 of the Interrupt Mask register). Pause frames received with non zero quantum are indicated through the interrupt bit 12 of the Interrupt Status register. Pause frames received with zero quantum are indicated on bit 13 of the Interrupt Status register. The loading of a new pause time only occurs when the ETH is configured for full duplex operation. If the ETH is configured for half duplex, the pause time counters will not be loaded, but the pause frame received interrupt will still be triggered. A valid pause frame is defined as having a destination address that matches either the address stored in Specific Address register 1 or if it matches the reserved address of 0x0180C2000001. It must also have the MAC control frame type ID of 0x8808 and have the pause opcode of 0x0101.

Pause frames that have frame check sequence (FCS) or other errors will be treated as invalid and will be discarded. Valid pause frames received will increment the Pause Frames Received Statistic register.

The Pause Time registers decrement every 512 bit times immediately following the PFC frame reception. For test purposes, the retry test bit can be set (bit 12 in the Network Configuration register) which causes the Pause Time register to decrement every RXCK cycle once transmission has stopped.

The interrupt (bit 13 in the Interrupt Status register) is asserted whenever the Pause Time register decrements to zero (assuming it has been enabled by bit 13 in the Interrupt Mask register). This interrupt is also set when a zero quantum pause frame is received.

32.6.17.2. PFC Pause Frame Transmission

Automatic transmission of pause frames is supported through the transmit priority-based pause frame bit of the Network Control register. If bit 17 of the Network Control register is written with logic 1, a PFC pause frame will be transmitted providing full duplex is selected in the Network Configuration register and the transmit block is enabled in the Network Control register. When bit 17 of the Network Control register is set, the fields of the priority-based pause frame will be built using the values stored in the Transmit PFC Pause register.

Pause frame transmission will happen immediately if transmit is inactive or if transmit is active between the current frame and the next frame due to be transmitted.

Transmitted pause frames comprise the following:

- A destination address of 01-80-C2-00-00-01
- A source address taken from Specific Address register 1
- A type ID of 88-08 (MAC control frame)
- A pause opcode of 01-01
- A priority enable vector taken from Transmit PFC Pause register
- 8 pause quantum registers
- Fill of 00 to take the frame to minimum frame length
- Valid FCS

The pause quantum registers used in the generated frame will depend on the trigger source for the frame as follows:

- If bit 17 of the Network Control register is written with a one, then the priority enable vector of the priority-based pause frame will be set equal to the value stored in the Transmit PFC Pause register [7:0]. For each entry equal to zero in the Transmit PFC Pause register [15:8], the pause quantum field of the pause frame associated with that entry will be taken from the transmit pause quantum register. For each entry equal to one in the Transmit PFC Pause register [15:8], the pause quantum associated with that entry will be zero.
- The Transmit Pause Quantum register resets to a value of 0xFFFF giving maximum pause quantum as default.

After transmission, a pause frame transmitted interrupt will be generated (bit 14 of the Interrupt Status register) and the only statistics register that will be incremented will be the Pause Frames Transmitted register.

PFC Pause frames can also be transmitted by the ETH using normal frame transmission methods.

32.6.18. Energy Efficient Ethernet Support

Features

- Energy Efficient Ethernet according to IEEE 802.3az
- A system's transmit path can enter a low power mode if there is nothing to transmit.
- A PHY can detect whether its link partner's transmit path is in low power mode, and configure its own receive path to enter low power mode.

- Link remains up during lower power mode and no frames are dropped.
- Asymmetric, one direction can be in low power mode while the other is transmitting normally.
- LPI (Low Power Idle) signaling is used to control entry and exit to and from low power modes.
  Note: LPI signaling can only take place if both sides have indicated support for it through auto-negotiation.

Operation

- Low power control is done at the MII (reconciliation sublayer).
- As an architectural convenience in writing the 802.3az it is assumed that transmission is deferred by asserting carrier sense - in practice it will not be done this way. This system will know when it has nothing to transmit and only enter low power mode when it is not transmitting.
- LPI should not be requested unless the link has been up for at least one second.
- LPI is signaled on the MII transmit path by asserting 0x01 on TXD with TX_EN low and TX_ER high.
- A PHY on seeing LPI requested on the MII will send the sleep signal before going quiet. After going quiet it will periodically emit refresh signals.
- The sleep, quiet and refresh periods are defined in 802.3az, Table 78-2.
- LPI mode ends by transmitting normal idle for the wake time. There is a default time for this but it can be adjusted in software using the Link Layer Discovery Protocol (LLDP) described in 802.3az, Clause 79.
- LPI is indicated at the receive side when sleep and refresh signaling has been detected.

32.6.19. LPI Operation in the EMAC

It is best to use firmware to control LPI. LPI operation happens at the system level. Firmware gives maximum control and flexibility of operation. LPI operation is straightforward and firmware should be capable of responding within the required timeframes.

Autonegotiation:

1. Indicate EEE capability using next page autonegotiation.

For the transmit path:

1. If the link has been up for 1 second and there is nothing being transmitted, write to the TXLPIEN bit in the Network Control register.
2. Wake up by clearing the TXLPIEN bit in the Network Control register.

For the receive path:

1. Enable RXLPISBC bit in IER. The bit RXLPIS is set in Network Status Register triggering an interrupt.
2. Wait for an interrupt to indicate that LPI has been received.
3. Disable relevant parts of the receive path if desired.
4. The RXLPIS bit in Network Status Register gets cleared to indicate that regular idle has been received. This triggers an interrupt.
5. Re-enable the receive path.

32.6.20. PHY Interface

Different PHY interfaces are supported by the Ethernet MAC:

- MII
- RMII

The MII interface is provided for 10/100 operation and uses TXD[3:0] and RXD[3:0]. The RMII interface is provided for 10/100 operation and uses TXD[1:0] and RXD[1:0].

32.6.21. Jumbo Frames

The jumbo frames enable bit in the Network Configuration register allows the ETH, in its default configuration, to receive jumbo frames up to 10240 bytes in size. This operation does not form part of the IEEE 802.3 specification and is normally disabled. When jumbo frames are enabled, frames received with a frame size greater than 10240 bytes are discarded.

32.7. Programming Interface

32.7.1. Initialization

32.7.1.1. Configuration

Initialization of the ETH configuration (i.e., loop back mode, frequency ratios) must be done while the transmit and receive circuits are disabled. See the description of the Network Control register and Network Configuration register earlier in this document.

To change loop back mode, the following sequence of operations must be followed:

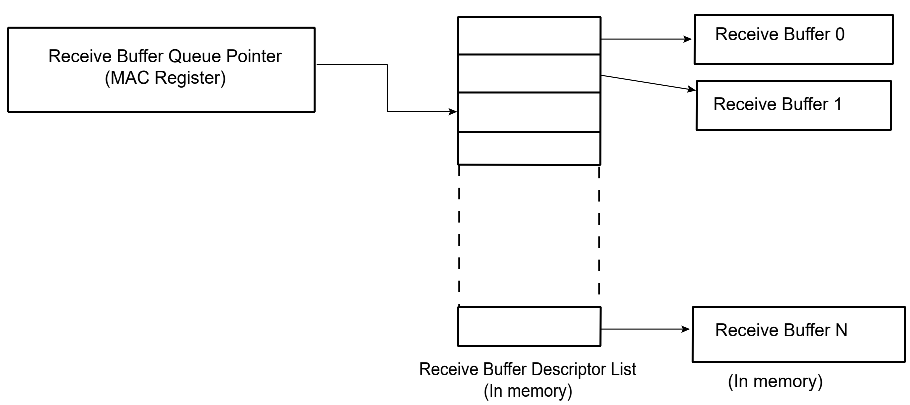
1. Write to Network Control register to disable transmit and receive circuits.
2. Write to Network Control register to change loop back mode.
3. Write to Network Control register to re-enable transmit or receive circuits.
   Note: These writes to the Network Control register cannot be combined in any way.

32.7.1.2. Receive Buffer List

Receive data is written to areas of data (i.e., buffers) in system memory. These buffers are listed in another data structure that also resides in main memory. This data structure (receive buffer queue) is a sequence of descriptor entries as defined in the table “Receive Buffer Description Entry”.

The Receive Buffer Queue Pointer register points to this data structure.

Figure 32-3. Receive Buffer List



To create the list of buffers:

1. Allocate a number (N) of buffers of X bytes in system memory, where X is the DMA buffer length programmed in the DMA Configuration register.
2. Allocate an area 8N bytes for the receive buffer descriptor list in system memory and create N entries in this list. Mark all entries in this list as owned by ETH, i.e., bit 0 of word 0 set to 0.
3. Mark the last descriptor in the queue with the wrap bit (bit 1 in word 0 set to 1).
4. Write address of receive buffer descriptor list and control information to ETH register receive buffer queue pointer
5. The receive circuits can then be enabled by writing to the address recognition registers and the Network Control register.

Note: The queue pointers must be initialized and point to USED descriptors for all queues including those not intended for use.

32.7.1.3. Transmit Buffer List

Transmit data is read from areas of data (the buffers) in system memory. These buffers are listed in another data structure that also resides in main memory. This data structure (Transmit Buffer Queue) is a sequence of descriptor entries as defined in the table “Transmit Buffer Description Entry”.

The Transmit Buffer Queue Pointer register points to this data structure.

To create this list of buffers:

1. Allocate a number (N) of buffers of between 1 and 2047 bytes of data to be transmitted in system memory. Up to 128 buffers per frame are allowed.
2. Allocate an area 8N bytes for the transmit buffer descriptor list in system memory and create N entries in this list. Mark all entries in this list as owned by ETH, i.e., bit 31 of word 1 set to 0.
3. Mark the last descriptor in the queue with the wrap bit (bit 30 in word 1 set to 1).
4. Write address of transmit buffer descriptor list and control information to ETH register transmit buffer queue pointer.
5. The transmit circuits can then be enabled by writing to the Network Control register.

Note: The queue pointers must be initialized and point to USED descriptors for all queues including those not intended for use.

32.7.1.4. Address Matching

The ETH Hash register pair and the four Specific Address register pairs must be written with the required values. Each register pair comprises of a bottom register and top register, with the bottom register being written first. The address matching is disabled for a particular register pair after the bottom register has been written and re-enabled when the top register is written. Each register pair may be written at any time, regardless of whether the receive circuits are enabled or disabled.

As an example, to set Specific Address register 1 to recognize destination address 21:43:65:87:A9:CB, the following values are written to Specific Address register 1 bottom and Specific Address register 1 top:

- Specific Address register 1 bottom bits 31:0 (0x98): 0x8765_4321
- Specific Address register 1 top bits 31:0 (0x9C): 0x0000_CBA9
  Note: The address matching is the first level of filtering. If there is a match, the screeners are the next level of filtering for routing the data to the appropriate queue. See “Priority Queueing in the DMA” for more details.

32.7.1.5. PHY Maintenance

The PHY Maintenance register is implemented as a shift register. Writing to the register starts a shift operation which is signalled as complete when bit two is set in the Network Status register (about 2000 MCK cycles later when bits 18:16 are set to 010 in the Network Configuration register). An interrupt is generated as this bit is set.

During this time, the MSB of the register is output on the MDIO pin and the LSB updated from the MDIO pin with each Management Data Clock (MDC) cycle. This causes the transmission of a PHY management frame on MDIO. See section 22.2.4.5 of the IEEE 802.3 standard.

Reading during the shift operation will return the current contents of the shift register. At the end of the management operation the bits will have shifted back to their original locations. For a read operation the data bits are updated with data read from the PHY. It is important to write the correct values to the register to ensure a valid PHY management frame is produced.

The Management Data Clock (MDC) should not toggle faster than 2.5 MHz (minimum period of 400 ns), as defined by the IEEE 802.3 standard. MDC is generated by dividing down MCK. Three bits in the Network Configuration register determine by how much MCK should be divided to produce MDC.

32.7.1.6. Interrupts

There are multiple interrupt conditions that are detected within the ETH. The conditions are ORed to make multiple interrupts. Each queue has its own interrupt vector. There are 6 interrupt lines connected to the NVIC correspondence to each queue. On receipt of the interrupt signal, the CPU enters the interrupt handler. Refer to the NVIC chapter to find more information about ETH Queue Interrupts.

At reset all interrupts are disabled. To enable an interrupt, write to Interrupt Enable register with the pertinent interrupt bit set to 1. To disable an interrupt, write to Interrupt Disable register with the pertinent interrupt bit set to 1. To check whether an interrupt is enabled or disabled, read Interrupt Mask register. If the bit is set to 1, the interrupt is disabled.

32.7.1.7. Transmitting Frames

The procedure to set up a frame for transmission is the following:

1. Enable transmit in the Network Control register.
2. Allocate an area of system memory for transmit data. This does not have to be contiguous, varying byte lengths can be used if they conclude on byte borders.
3. Set-up the transmit buffer list by writing buffer addresses to word zero of the transmit buffer descriptor entries and control and length to word one.
4. Write data for transmission into the buffers pointed to by the descriptors.
5. Write the address of the first buffer descriptor to transmit buffer descriptor queue pointer.
6. Enable appropriate interrupts.
7. Write to the transmit start bit (TSTART) in the Network Control register.

32.7.1.8. Receiving Frames

When a frame is received and the receive circuits are enabled, the ETH checks the address and, in the following cases, the frame is written to system memory:

- If it matches one of the four Specific Address registers
- If it matches one of the four type ID registers
- If it matches the hash address function
- If it is a broadcast address (0xFFFFFFFFFFFF) and broadcasts are allowed
- If the ETH is configured to “copy all frames”

The register receive buffer queue pointer points to the next entry in the receive buffer descriptor list and the ETH uses this as the address in system memory to write the frame to.

Once the frame has been completely and successfully received and written to system memory, the ETH then updates the receive buffer descriptor entry (see Receive Buffer Description Entry) with the reason for the address match and marks the area as being owned by software. Once this is complete, a receive complete interrupt is set. Software is then responsible for copying the data to the application area and releasing the buffer (by writing the ownership bit back to 0).

If the ETH is unable to write the data at a rate to match the incoming frame, then a receive overrun interrupt is set. If there is no receive buffer available, i.e., the next buffer is still owned by software, a receive buffer not available interrupt is set. If the frame is not successfully received, a statistics register is incremented and the frame is discarded without informing software.

32.7.2. Statistics Registers

Statistics registers are described in the user interface beginning with ETH Octets Transmitted Low Register and ending with ETH UDP Checksum Errors Register.

The statistics register block begins at 0x1100 and runs to 0x11B0, and comprises the registers listed below.

Octets Transmitted Low Register	Broadcast Frames Received Register
Octets Transmitted High Register	Multicast Frames Received Register
Frames Transmitted Register	Pause Frames Received Register
Broadcast Frames Transmitted Register	64 Byte Frames Received Register
Multicast Frames Transmitted Register	65 to 127 Byte Frames Received Register
Pause Frames Transmitted Register	128 to 255 Byte Frames Received Register
64 Byte Frames Transmitted Register	256 to 511 Byte Frames Received Register
65 to 127 Byte Frames Transmitted Register	512 to 1023 Byte Frames Received Register
128 to 255 Byte Frames Transmitted Register	1024 to 1518 Byte Frames Received Register
256 to 511 Byte Frames Transmitted Register	1519 to Maximum Byte Frames Received Register
512 to 1023 Byte Frames Transmitted Register	Undersize Frames Received Register
1024 to 1518 Byte Frames Transmitted Register	Oversize Frames Received Register
Greater Than 1518 Byte Frames Transmitted Register	Jabbers Received Register
Transmit Underruns Register	Frame Check Sequence Errors Register
Single Collision Frames Register	Length Field Frame Errors Register
Multiple Collision Frames Register	Receive Symbol Errors Register
Excessive Collisions Register	Alignment Errors Register
Late Collisions Register	Receive Resource Errors Register
Deferred Transmission Frames Register	Receive Overrun Register
Carrier Sense Errors Register	IP Header Checksum Errors Register
Octets Received Low Register	TCP Checksum Errors Register
Octets Received High Register	UDP Checksum Errors Register
Frames Received Register	

These registers reset to zero on a read and stick at all ones when they count to their maximum value. They should be read frequently enough to prevent loss of data.

The receive statistics registers are only incremented when the receive enable bit (RXEN) is set in the Network Control register.

Once a statistics register has been read, it is automatically cleared. When reading the Octets Transmitted and Octets Received registers, bits 31:0 should be read prior to bits 47:32 to ensure reliable operation.

32.8. Register Summary

For descriptions and definitions of both Register and bitfield properties, refer to Register Properties.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRLA	31:24								
		23:16								
		15:8								
		7:0		RUNSTDBY					ENABLE	
0x04	CTRLB	31:24								
		23:16								
		15:8								
		7:0	TSUINC[1:0]		TSUMS			TSUCLKREQ		
0x08 ... 0x0B	Reserved									
0x0C	EVCTRL	31:24								
		23:16								
		15:8								
		7:0								CMPEO
0x10 ... 0x1F	Reserved									
0x20	SYNCB	31:24								
		23:16								
		15:8								
		7:0							ENABLE	SWRST
0x24 ... 0x0FFF	Reserved									
0x1000	NCR	31:24								
		23:16					LPI	FNP	TXPBPF	ENPBPR
		15:8	SRTSM			TXZQPF	TXPF	THALT	TSTART	BP
		7:0	WESTAT	INCSTAT	CLRSTAT	MPE	TXEN	RXEN	LBL	
0x1004	NCFGR	31:24		IRXER	RXBP	IPGSEN		IRXFCS	EFRHD	RXCOEN
		23:16	DCPF	DBW[1:0]		CLK[2:0]			RFCS	LFERD
		15:8	RXBUFO[1:0]		PEN	RTY				MAXFS
		7:0	UNIHEN	MTIHEN	NBC	CAF	JFRAME	DNVLAN	FD	SPD
0x1008	NSR	31:24								
		23:16								
		15:8								
		7:0						IDLE	MDIO	
0x100C	UR	31:24								
		23:16								
		15:8								
		7:0								MII
0x1010	DCFGR	31:24								DDRP
		23:16	DRBS[7:0]							
		15:8					TXCOEN	TXPBMS	RXBMS[1:0]	
		7:0	ESPA	ESMA		FBLDO[4:0]				
0x1014	TSR	31:24								
		23:16								
		15:8								HRESP
		7:0		UND	TXCOMP	TFC	TXGO	RLE	COL	UBR
0x1018	RBQB	31:24	ADDR[29:22]							
		23:16	ADDR[21:14]							
		15:8	ADDR[13:6]							
		7:0	ADDR[5:0]							
0x101C	TBQB	31:24	ADDR[29:22]							
		23:16	ADDR[21:14]							
		15:8	ADDR[13:6]							
		7:0	ADDR[5:0]							

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x101C	TBFT127	31:24	NFTX[31:24]							
		23:16	NFTX[23:16]							
		15:8	NFTX[15:8]							
		7:0	NFTX[7:0]							
0x1020	RSR	31:24								
		23:16								
		15:8								
		7:0					HNO	RXOVR	REC	BNA
0x1024	ISR	31:24			TSUTIMCMP	WOL	RXLPISBC	SRI	PDRSFT	PDRQFT
		23:16	PDRSFR	PDRQFR	SFT	DRQFT	SFR	DRQFR		
		15:8		PFTR	PTZ	PFNZ	HRESP	ROVR		
		7:0	TCOMP	TFC	RLEX	TUR	TXUBR	RXUBR	RCOMP	MFS
0x1028	IER	31:24			TSUTIMCMP	WOL	RXLPISBC	SRI	PDRSFT	PDRQFT
		23:16	PDRSFR	PDRQFR	SFT	DRQFT	SFR	DRQFR		
		15:8	EXINT	PFTR	PTZ	PFNZ	HRESP	ROVR		
		7:0	TCOMP	TFC	RLEX	TUR	TXUBR	RXUBR	RCOMP	MFS
0x102C	IDR	31:24			TSUTIMCMP	WOL	RXLPISBC	SRI	PDRSFT	PDRQFT
		23:16	PDRSFR	PDRQFR	SFT	DRQFT	SFR	DRQFR		
		15:8	EXINT	PFTR	PTZ	PFNZ	HRESP	ROVR		
		7:0	TCOMP	TFC	RLEX	TUR	TXUBR	RXUBR	RCOMP	MFS
0x1030	IMR	31:24			TSUTIMCMP	WOL	RXLPISBC	SRI	PDRSFT	PDRQFT
		23:16	PDRSFR	PDRQFR	SFT	DRQFT	SFR	DRQFR		
		15:8	EXINT	PFTR	PTZ	PFNZ	HRESP	ROVR		
		7:0	TCOMP	TFC	RLEX	TUR	TXUBR	RXUBR	RCOMP	MFS
0x1034	MAN	31:24	WZO	CLTTO	OP[1:0]		PHYA[4:1]			
		23:16	PHYA[0]	REGA[4:0]					WTN[1:0]	
		15:8	DATA[15:8]							
		7:0	DATA[7:0]							
0x1038	RPQ	31:24								
		23:16								
		15:8	RPQ[15:8]							
		7:0	RPQ[7:0]							
0x103C	TPQ	31:24								
		23:16								
		15:8	TPQ[15:8]							
		7:0	TPQ[7:0]							
0x1040	TPSF	31:24	ENTXP							
		23:16								
		15:8					TPB1ADR[11:8]			
		7:0	TPB1ADR[7:0]							
0x1044	RPSF	31:24	ENRXP							
		23:16								
		15:8					RPB1ADR[11:8]			
		7:0	RPB1ADR[7:0]							
0x1048	RJFML	31:24								
		23:16								
		15:8			FML[13:8]					
		7:0	FML[7:0]							
0x104C ... 0x107F	Reserved									
0x1080	HRB	31:24	ADDR[31:24]							
		23:16	ADDR[23:16]							
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0x1084	HRT	31:24	ADDR[31:24]							
		23:16	ADDR[23:16]							
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x1088	SAB1	31:24	ADDR[31:24]							
		23:16	ADDR[23:16]							
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0x108C ... 0x108F	Reserved									
0x1090	SAB2	31:24	ADDR[31:24]							
		23:16	ADDR[23:16]							
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0x1094 ... 0x1097	Reserved									
0x1098	SAB3	31:24	ADDR[31:24]							
		23:16	ADDR[23:16]							
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0x109C ... 0x109F	Reserved									
0x10A0	SAB4	31:24	ADDR[31:24]							
		23:16	ADDR[23:16]							
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0x10A4 ... 0x10A7	Reserved									
0x10A8	TIDM1	31:24	ENIDn							
		23:16								
		15:8	TID[15:8]							
		7:0	TID[7:0]							
0x10AC	TIDM2	31:24	ENIDn							
		23:16								
		15:8	TID[15:8]							
		7:0	TID[7:0]							
0x10B0	TIDM3	31:24	ENIDn							
		23:16								
		15:8	TID[15:8]							
		7:0	TID[7:0]							
0x10B4	TIDM4	31:24	ENIDn							
		23:16								
		15:8	TID[15:8]							
		7:0	TID[7:0]							
0x10B8	WOL	31:24								
		23:16					MTI	SA1	ARP	MAG
		15:8	IP[15:8]							
		7:0	IP[7:0]							
0x10BC	IPGS	31:24								
		23:16								
		15:8	FL[15:8]							
		7:0	FL[7:0]							
0x10C0	SVLAN	31:24	ESVLAN							
		23:16								
		15:8	VLAN_TYPE[15:8]							
		7:0	VLAN_TYPE[7:0]							
0x10C4	TPFCP	31:24								
		23:16								
		15:8	PQ[7:0]							
		7:0	PEV[7:0]							

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x10C8	SAMB1	31:24	ADDR[31:24]							
		23:16	ADDR[23:16]							
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0x10CC	SAMT1	31:24								
		23:16								
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0x10D0 ... 0x10DB	Reserved									
0x10DC	NSC	31:24								
		23:16			NANOSEC[21:16]					
		15:8	NANOSEC[15:8]							
		7:0	NANOSEC[7:0]							
0x10E0	SCL	31:24	SEC[31:24]							
		23:16	SEC[23:16]							
		15:8	SEC[15:8]							
		7:0	SEC[7:0]							
0x10E4	SCH	31:24								
		23:16								
		15:8	SEC[15:8]							
		7:0	SEC[7:0]							
0x10E8	EFTSH	31:24								
		23:16								
		15:8	RUD[15:8]							
		7:0	RUD[7:0]							
0x10EC	EFRSH	31:24								
		23:16								
		15:8	RUD[15:8]							
		7:0	RUD[7:0]							
0x10F0	PEFTSH	31:24								
		23:16								
		15:8	RUD[15:8]							
		7:0	RUD[7:0]							
0x10F4	PEFRSH	31:24								
		23:16								
		15:8	RUD[15:8]							
		7:0	RUD[7:0]							
0x10F8 ... 0x10FF	Reserved									
0x1100	OTLO	31:24	TXO[31:24]							
		23:16	TXO[23:16]							
		15:8	TXO[15:8]							
		7:0	TXO[7:0]							
0x1104	OTHI	31:24								
		23:16								
		15:8	TXO[15:8]							
		7:0	TXO[7:0]							
0x1108	FT	31:24	FTX[31:24]							
		23:16	FTX[23:16]							
		15:8	FTX[15:8]							
		7:0	FTX[7:0]							
0x110C	BCFT	31:24	BFTX[31:24]							
		23:16	BFTX[23:16]							
		15:8	BFTX[15:8]							
		7:0	BFTX[7:0]							

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x1110	MFT	31:24	MFTX[31:24]							
		23:16	MFTX[23:16]							
		15:8	MFTX[15:8]							
		7:0	MFTX[7:0]							
0x1114	PFT	31:24								
		23:16								
		15:8	PFTX[15:8]							
		7:0	PFTX[7:0]							
0x1118	BFT64	31:24	NFTX[31:24]							
		23:16	NFTX[23:16]							
		15:8	NFTX[15:8]							
		7:0	NFTX[7:0]							
0x111C ... 0x111F	Reserved									
0x1120	TBFT255	31:24	NFTX[31:24]							
		23:16	NFTX[23:16]							
		15:8	NFTX[15:8]							
		7:0	NFTX[7:0]							
0x1124	TBFT511	31:24	NFTX[31:24]							
		23:16	NFTX[23:16]							
		15:8	NFTX[15:8]							
		7:0	NFTX[7:0]							
0x1128	TBFT1023	31:24	NFTX[31:24]							
		23:16	NFTX[23:16]							
		15:8	NFTX[15:8]							
		7:0	NFTX[7:0]							
0x112C	TBFT1518	31:24	NFTX[31:24]							
		23:16	NFTX[23:16]							
		15:8	NFTX[15:8]							
		7:0	NFTX[7:0]							
0x1130	GTBFT1518	31:24	NFTX[31:24]							
		23:16	NFTX[23:16]							
		15:8	NFTX[15:8]							
		7:0	NFTX[7:0]							
0x1134	TUR	31:24								
		23:16								
		15:8							TXUNR[9:8]	
		7:0	TXUNR[7:0]							
0x1138	SCF	31:24								
		23:16							SCOL[17:16]	
		15:8	SCOL[15:8]							
		7:0	SCOL[7:0]							
0x113C	MCF	31:24								
		23:16							MCOL[17:16]	
		15:8	MCOL[15:8]							
		7:0	MCOL[7:0]							
0x1140	EC	31:24								
		23:16								
		15:8							XCOL[9:8]	
		7:0	XCOL[7:0]							
0x1144	LC	31:24								
		23:16								
		15:8							LCOL[9:8]	
		7:0	LCOL[7:0]							
0x1148	DTF	31:24								
		23:16							DEFT[17:16]	
		15:8	DEFT[15:8]							
		7:0	DEFT[7:0]							

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x114C	CSE	31:24								
		23:16								
		15:8							CSR[9:8]	
		7:0	CSR[7:0]							
0x1150	ORLO	31:24	RXO[31:24]							
		23:16	RXO[23:16]							
		15:8	RXO[15:8]							
		7:0	RXO[7:0]							
0x1154	ORHI	31:24								
		23:16								
		15:8	RXO[15:8]							
		7:0	RXO[7:0]							
0x1158	FR	31:24	FRX[31:24]							
		23:16	FRX[23:16]							
		15:8	FRX[15:8]							
		7:0	FRX[7:0]							
0x115C	BCFR	31:24	BFRX[31:24]							
		23:16	BFRX[23:16]							
		15:8	BFRX[15:8]							
		7:0	BFRX[7:0]							
0x1160	MFR	31:24	MFRX[31:24]							
		23:16	MFRX[23:16]							
		15:8	MFRX[15:8]							
		7:0	MFRX[7:0]							
0x1164	PFR	31:24								
		23:16								
		15:8	PFRX[15:8]							
		7:0	PFRX[7:0]							
0x1168	BFR64	31:24	NFRX[31:24]							
		23:16	NFRX[23:16]							
		15:8	NFRX[15:8]							
		7:0	NFRX[7:0]							
0x116C	TBFR127	31:24	NFRX[31:24]							
		23:16	NFRX[23:16]							
		15:8	NFRX[15:8]							
		7:0	NFRX[7:0]							
0x1170	TBFR255	31:24	NFRX[31:24]							
		23:16	NFRX[23:16]							
		15:8	NFRX[15:8]							
		7:0	NFRX[7:0]							
0x1174	TBFR511	31:24	NFRX[31:24]							
		23:16	NFRX[23:16]							
		15:8	NFRX[15:8]							
		7:0	NFRX[7:0]							
0x1178	TBFR1023	31:24	NFRX[31:24]							
		23:16	NFRX[23:16]							
		15:8	NFRX[15:8]							
		7:0	NFRX[7:0]							
0x117C	TBFR1518	31:24	NFRX[31:24]							
		23:16	NFRX[23:16]							
		15:8	NFRX[15:8]							
		7:0	NFRX[7:0]							
0x1180	TMXBFR	31:24	NFRX[31:24]							
		23:16	NFRX[23:16]							
		15:8	NFRX[15:8]							
		7:0	NFRX[7:0]							
0x1184	UFR	31:24								
		23:16								
		15:8							UFRX[9:8]	
		7:0	UFRX[7:0]							

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x1188	OFR	31:24								
		23:16								
		15:8							OFRX[9:8]	
		7:0	OFRX[7:0]							
0x118C	JR	31:24								
		23:16								
		15:8							JRX[9:8]	
		7:0	JRX[7:0]							
0x1190	FCSE	31:24								
		23:16								
		15:8							FCKR[9:8]	
		7:0	FCKR[7:0]							
0x1194	LFFE	31:24								
		23:16								
		15:8							LFER[9:8]	
		7:0	LFER[7:0]							
0x1198	RSE	31:24								
		23:16								
		15:8							RXSE[9:8]	
		7:0	RXSE[7:0]							
0x119C	AE	31:24								
		23:16								
		15:8							AER[9:8]	
		7:0	AER[7:0]							
0x11A0	RRE	31:24								
		23:16							RXRER[17:16]	
		15:8	RXRER[15:8]							
		7:0	RXRER[7:0]							
0x11A4	ROE	31:24								
		23:16								
		15:8							RXOVR[9:8]	
		7:0	RXOVR[7:0]							
0x11A8	IHCE	31:24								
		23:16								
		15:8								
		7:0	HCKER[7:0]							
0x11AC	TCE	31:24								
		23:16								
		15:8								
		7:0	TCKER[7:0]							
0x11B0	UCE	31:24								
		23:16								
		15:8								
		7:0	UCKER[7:0]							
0x11B4 ... 0x11BB	Reserved									
0x11BC	TISUBN	31:24								
		23:16								
		15:8	LSBTIR[15:8]							
		7:0	LSBTIR[7:0]							
0x11C0	TSH	31:24								
		23:16								
		15:8	TCS[15:8]							
		7:0	TCS[7:0]							
0x11C4 ... 0x11C7	Reserved									

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x11C8	TSSSL	31:24	VTS[31:24]							
		23:16	VTS[23:16]							
		15:8	VTS[15:8]							
		7:0	VTS[7:0]							
0x11CC	TSSSN	31:24	VTN[31:24]							
		23:16	VTN[23:16]							
		15:8	VTN[15:8]							
		7:0	VTN[7:0]							
0x11D0	TSL	31:24	TCS[31:24]							
		23:16	TCS[23:16]							
		15:8	TCS[15:8]							
		7:0	TCS[7:0]							
0x11D4	TN	31:24			TNS[29:24]					
		23:16	TNS[23:16]							
		15:8	TNS[15:8]							
		7:0	TNS[7:0]							
0x11D8	TA	31:24	ADJ		ITDT[29:24]					
		23:16	ITDT[23:16]							
		15:8	ITDT[15:8]							
		7:0	ITDT[7:0]							
0x11DC	TI	31:24								
		23:16	NIT[7:0]							
		15:8	ACNS[7:0]							
		7:0	CNS[7:0]							
0x11E0	EFTSL	31:24	RUD[31:24]							
		23:16	RUD[23:16]							
		15:8	RUD[15:8]							
		7:0	RUD[7:0]							
0x11E4	EFTN	31:24			RUD[29:24]					
		23:16	RUD[23:16]							
		15:8	RUD[15:8]							
		7:0	RUD[7:0]							
0x11E8	EFRSL	31:24	RUD[31:24]							
		23:16	RUD[23:16]							
		15:8	RUD[15:8]							
		7:0	RUD[7:0]							
0x11EC	EFRN	31:24			RUD[29:24]					
		23:16	RUD[23:16]							
		15:8	RUD[15:8]							
		7:0	RUD[7:0]							
0x11F0	PEFTSL	31:24	RUD[31:24]							
		23:16	RUD[23:16]							
		15:8	RUD[15:8]							
		7:0	RUD[7:0]							
0x11F4	PEFTN	31:24			RUD[29:24]					
		23:16	RUD[23:16]							
		15:8	RUD[15:8]							
		7:0	RUD[7:0]							
0x11F8	PEFRSL	31:24	RUD[31:24]							
		23:16	RUD[23:16]							
		15:8	RUD[15:8]							
		7:0	RUD[7:0]							
0x11FC	PEFRN	31:24			RUD[29:24]					
		23:16	RUD[23:16]							
		15:8	RUD[15:8]							
		7:0	RUD[7:0]							
0x1200 ... 0x126F	Reserved									

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x1270	RLPITR	31:24								
		23:16								
		15:8	RLPITR[15:8]							
		7:0	RLPITR[7:0]							
0x1274	RLPITI	31:24								
		23:16	RLPITI[23:16]							
		15:8	RLPITI[15:8]							
		7:0	RLPITI[7:0]							
0x1278	TLPITR	31:24								
		23:16	TLPITR[23:16]							
		15:8	TLPITR[15:8]							
		7:0	TLPITR[7:0]							
0x127C	TLPITI	31:24								
		23:16	TLPITI[23:16]							
		15:8	TLPITI[15:8]							
		7:0	TLPITI[7:0]							
0x1280 ... 0x13FF	Reserved									
0x1400	ISRQ1	31:24								
		23:16								
		15:8					HRESP	ROVR		
		7:0	TCOMP	TFC	RLEX	TUR	TXUBR	RXUBR	RCOMP	
0x1404	ISRQ2	31:24								
		23:16								
		15:8					HRESP	ROVR		
		7:0	TCOMP	TFC	RLEX	TUR	TXUBR	RXUBR	RCOMP	
0x1408	ISRQ3	31:24								
		23:16								
		15:8					HRESP	ROVR		
		7:0	TCOMP	TFC	RLEX	TUR	TXUBR	RXUBR	RCOMP	
0x140C	ISRQ4	31:24								
		23:16								
		15:8					HRESP	ROVR		
		7:0	TCOMP	TFC	RLEX	TUR	TXUBR	RXUBR	RCOMP	
0x1410	ISRQ5	31:24								
		23:16								
		15:8					HRESP	ROVR		
		7:0	TCOMP	TFC	RLEX	TUR	TXUBR	RXUBR	RCOMP	
0x1414 ... 0x143F	Reserved									
0x1440	TBPQB1	31:24	TXBQBA[29:22]							
		23:16	TXBQBA[21:14]							
		15:8	TXBQBA[13:6]							
		7:0	TXBQBA[5:0]							
0x1444	TBPQB2	31:24	TXBQBA[29:22]							
		23:16	TXBQBA[21:14]							
		15:8	TXBQBA[13:6]							
		7:0	TXBQBA[5:0]							
0x1448	TBPQB3	31:24	TXBQBA[29:22]							
		23:16	TXBQBA[21:14]							
		15:8	TXBQBA[13:6]							
		7:0	TXBQBA[5:0]							
0x144C	TBPQB4	31:24	TXBQBA[29:22]							
		23:16	TXBQBA[21:14]							
		15:8	TXBQBA[13:6]							
		7:0	TXBQBA[5:0]							

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x1450	TBPQB5	31:24	TXBQBA[29:22]							
		23:16	TXBQBA[21:14]							
		15:8	TXBQBA[13:6]							
		7:0	TXBQBA[5:0]							
0x1454 ... 0x147F	Reserved									
0x1480	RBPQB1	31:24	RXBQBA[29:22]							
		23:16	RXBQBA[21:14]							
		15:8	RXBQBA[13:6]							
		7:0	RXBQBA[5:0]							
0x1484	RBPQB2	31:24	RXBQBA[29:22]							
		23:16	RXBQBA[21:14]							
		15:8	RXBQBA[13:6]							
		7:0	RXBQBA[5:0]							
0x1488	RBPQB3	31:24	RXBQBA[29:22]							
		23:16	RXBQBA[21:14]							
		15:8	RXBQBA[13:6]							
		7:0	RXBQBA[5:0]							
0x148C	RBPQB4	31:24	RXBQBA[29:22]							
		23:16	RXBQBA[21:14]							
		15:8	RXBQBA[13:6]							
		7:0	RXBQBA[5:0]							
0x1490	RBPQB5	31:24	RXBQBA[29:22]							
		23:16	RXBQBA[21:14]							
		15:8	RXBQBA[13:6]							
		7:0	RXBQBA[5:0]							
0x1494 ... 0x149F	Reserved									
0x14A0	RBQSZ1	31:24								
		23:16								
		15:8	RBS[15:8]							
		7:0	RBS[7:0]							
0x14A4	RBQSZ2	31:24								
		23:16								
		15:8	RBS[15:8]							
		7:0	RBS[7:0]							
0x14A8	RBQSZ3	31:24								
		23:16								
		15:8	RBS[15:8]							
		7:0	RBS[7:0]							
0x14AC	RBQSZ4	31:24								
		23:16								
		15:8	RBS[15:8]							
		7:0	RBS[7:0]							
0x14B0	RBQSZ5	31:24								
		23:16								
		15:8	RBS[15:8]							
		7:0	RBS[7:0]							
0x14B4 ... 0x14FF	Reserved									
0x1500	SCRT10	31:24			UDPE	DSTCE	UDPP[15:12]			
		23:16	UDPP[11:4]							
		15:8	UDPP[3:0]				DSTCM[7:4]			
		7:0	DSTCM[3:0]					QNMBR[2:0]		

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x1504	SCRT11	31:24			UDPE	DSTCE	UDPP[15:12]			
		23:16	UDPP[11:4]							
		15:8	UDPP[3:0]				DSTCM[7:4]			
		7:0	DSTCM[3:0]					QNMBR[2:0]		
0x1508	SCRT12	31:24			UDPE	DSTCE	UDPP[15:12]			
		23:16	UDPP[11:4]							
		15:8	UDPP[3:0]				DSTCM[7:4]			
		7:0	DSTCM[3:0]					QNMBR[2:0]		
0x150C	SCRT13	31:24			UDPE	DSTCE	UDPP[15:12]			
		23:16	UDPP[11:4]							
		15:8	UDPP[3:0]				DSTCM[7:4]			
		7:0	DSTCM[3:0]					QNMBR[2:0]		
0x1510 ... 0x153F	Reserved									
0x1540	SCRT20	31:24		COMPCE	COMPC[4:0]					COMPBE
		23:16	COMPB[4:0]					COMPAE	COMPA[4:3]	
		15:8	COMPA[2:0]			ETHE	I2ETH[2:0]			VLANE
		7:0		VLANP[2:0]				QNB[2:0]		
0x1544	SCRT21	31:24		COMPCE	COMPC[4:0]					COMPBE
		23:16	COMPB[4:0]					COMPAE	COMPA[4:3]	
		15:8	COMPA[2:0]			ETHE	I2ETH[2:0]			VLANE
		7:0		VLANP[2:0]				QNB[2:0]		
0x1548	SCRT22	31:24		COMPCE	COMPC[4:0]					COMPBE
		23:16	COMPB[4:0]					COMPAE	COMPA[4:3]	
		15:8	COMPA[2:0]			ETHE	I2ETH[2:0]			VLANE
		7:0		VLANP[2:0]				QNB[2:0]		
0x154C	SCRT23	31:24		COMPCE	COMPC[4:0]					COMPBE
		23:16	COMPB[4:0]					COMPAE	COMPA[4:3]	
		15:8	COMPA[2:0]			ETHE	I2ETH[2:0]			VLANE
		7:0		VLANP[2:0]				QNB[2:0]		
0x1550	SCRT24	31:24		COMPCE	COMPC[4:0]					COMPBE
		23:16	COMPB[4:0]					COMPAE	COMPA[4:3]	
		15:8	COMPA[2:0]			ETHE	I2ETH[2:0]			VLANE
		7:0		VLANP[2:0]				QNB[2:0]		
0x1554	SCRT25	31:24		COMPCE	COMPC[4:0]					COMPBE
		23:16	COMPB[4:0]					COMPAE	COMPA[4:3]	
		15:8	COMPA[2:0]			ETHE	I2ETH[2:0]			VLANE
		7:0		VLANP[2:0]				QNB[2:0]		
0x1558	SCRT26	31:24		COMPCE	COMPC[4:0]					COMPBE
		23:16	COMPB[4:0]					COMPAE	COMPA[4:3]	
		15:8	COMPA[2:0]			ETHE	I2ETH[2:0]			VLANE
		7:0		VLANP[2:0]				QNB[2:0]		
0x155C	SCRT27	31:24		COMPCE	COMPC[4:0]					COMPBE
		23:16	COMPB[4:0]					COMPAE	COMPA[4:3]	
		15:8	COMPA[2:0]			ETHE	I2ETH[2:0]			VLANE
		7:0		VLANP[2:0]				QNB[2:0]		
0x1560 ... 0x15FF	Reserved									
0x1600	IERQ1	31:24								
		23:16								
		15:8					HRESP	ROVR		
		7:0	TCOMP	TFC	RLEX	TUR	TXUBR	RXUBR	RCOMP	
0x1604	IERQ2	31:24								
		23:16								
		15:8					HRESP	ROVR		
		7:0	TCOMP	TFC	RLEX	TUR	TXUBR	RXUBR	RCOMP	

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x1608	IERQ3	31:24								
		23:16								
		15:8					HRESP	ROVR		
		7:0	TCOMP	TFC	RLEX	TUR	TXUBR	RXUBR	RCOMP	
0x160C	IERQ4	31:24								
		23:16								
		15:8					HRESP	ROVR		
		7:0	TCOMP	TFC	RLEX	TUR	TXUBR	RXUBR	RCOMP	
0x1610	IERQ5	31:24								
		23:16								
		15:8					HRESP	ROVR		
		7:0	TCOMP	TFC	RLEX	TUR	TXUBR	RXUBR	RCOMP	
0x1614 ... 0x161F	Reserved									
0x1620	IDRQ1	31:24								
		23:16								
		15:8					HRESP	ROVR		
		7:0	TCOMP	TFC	RLEX	TUR	TXUBR	RXUBR	RCOMP	
0x1624	IDRQ2	31:24								
		23:16								
		15:8					HRESP	ROVR		
		7:0	TCOMP	TFC	RLEX	TUR	TXUBR	RXUBR	RCOMP	
0x1628	IDRQ3	31:24								
		23:16								
		15:8					HRESP	ROVR		
		7:0	TCOMP	TFC	RLEX	TUR	TXUBR	RXUBR	RCOMP	
0x162C	IDRQ4	31:24								
		23:16								
		15:8					HRESP	ROVR		
		7:0	TCOMP	TFC	RLEX	TUR	TXUBR	RXUBR	RCOMP	
0x1630	IDRQ5	31:24								
		23:16								
		15:8					HRESP	ROVR		
		7:0	TCOMP	TFC	RLEX	TUR	TXUBR	RXUBR	RCOMP	
0x1634 ... 0x163F	Reserved									
0x1640	IMRQ1	31:24								
		23:16								
		15:8					HRESP	ROVR		
		7:0	TCOMP	AXI	RLEX	TUR	TXUBR	RXUBR	RCOMP	
0x1644	IMRQ2	31:24								
		23:16								
		15:8					HRESP	ROVR		
		7:0	TCOMP	AXI	RLEX	TUR	TXUBR	RXUBR	RCOMP	
0x1648	IMRQ3	31:24								
		23:16								
		15:8					HRESP	ROVR		
		7:0	TCOMP	AXI	RLEX	TUR	TXUBR	RXUBR	RCOMP	
0x164C	IMRQ4	31:24								
		23:16								
		15:8					HRESP	ROVR		
		7:0	TCOMP	AXI	RLEX	TUR	TXUBR	RXUBR	RCOMP	
0x1650	IMRQ5	31:24								
		23:16								
		15:8					HRESP	ROVR		
		7:0	TCOMP	AXI	RLEX	TUR	TXUBR	RXUBR	RCOMP	

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x1654 ... 0x16DF	Reserved									
0x16E0	SCRT2ET0	31:24								
		23:16								
		15:8	COMPVAL[15:8]							
		7:0	COMPVAL[7:0]							
0x16E4	SCRT2ET1	31:24								
		23:16								
		15:8	COMPVAL[15:8]							
		7:0	COMPVAL[7:0]							
0x16E8	SCRT2ET2	31:24								
		23:16								
		15:8	COMPVAL[15:8]							
		7:0	COMPVAL[7:0]							
0x16EC	SCRT2ET3	31:24								
		23:16								
		15:8	COMPVAL[15:8]							
		7:0	COMPVAL[7:0]							
0x16F0 ... 0x16FF	Reserved									
0x1700	SCRT2CMP00	31:24	COMPVAL[15:8]							
		23:16	COMPVAL[7:0]							
		15:8	MASK[15:8]							
		7:0	MASK[7:0]							
0x1704	SCRT2CMP10	31:24								
		23:16								
		15:8								OFFSSTRT[1]
		7:0	OFFSSTRT[0]	OFFSVAL[6:0]						
0x1708	SCRT2CMP01	31:24	COMPVAL[15:8]							
		23:16	COMPVAL[7:0]							
		15:8	MASK[15:8]							
		7:0	MASK[7:0]							
0x170C	SCRT2CMP11	31:24								
		23:16								
		15:8								OFFSSTRT[1]
		7:0	OFFSSTRT[0]	OFFSVAL[6:0]						
0x1710	SCRT2CMP02	31:24	COMPVAL[15:8]							
		23:16	COMPVAL[7:0]							
		15:8	MASK[15:8]							
		7:0	MASK[7:0]							
0x1714	SCRT2CMP12	31:24								
		23:16								
		15:8								OFFSSTRT[1]
		7:0	OFFSSTRT[0]	OFFSVAL[6:0]						
0x1718	SCRT2CMP03	31:24	COMPVAL[15:8]							
		23:16	COMPVAL[7:0]							
		15:8	MASK[15:8]							
		7:0	MASK[7:0]							
0x171C	SCRT2CMP13	31:24								
		23:16								
		15:8								OFFSSTRT[1]
		7:0	OFFSSTRT[0]	OFFSVAL[6:0]						
0x1720	SCRT2CMP04	31:24	COMPVAL[15:8]							
		23:16	COMPVAL[7:0]							
		15:8	MASK[15:8]							
		7:0	MASK[7:0]							

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x1724	SCRT2CMP14	31:24								
		23:16								
		15:8								OFFSSTRT[1]
		7:0	OFFSSTRT[0]	OFFSVAL[6:0]						
0x1728	SCRT2CMP05	31:24	COMPVAL[15:8]							
		23:16	COMPVAL[7:0]							
		15:8	MASK[15:8]							
		7:0	MASK[7:0]							
0x172C	SCRT2CMP15	31:24								
		23:16								
		15:8								OFFSSTRT[1]
		7:0	OFFSSTRT[0]	OFFSVAL[6:0]						
0x1730	SCRT2CMP06	31:24	COMPVAL[15:8]							
		23:16	COMPVAL[7:0]							
		15:8	MASK[15:8]							
		7:0	MASK[7:0]							
0x1734	SCRT2CMP16	31:24								
		23:16								
		15:8								OFFSSTRT[1]
		7:0	OFFSSTRT[0]	OFFSVAL[6:0]						
0x1738	SCRT2CMP07	31:24	COMPVAL[15:8]							
		23:16	COMPVAL[7:0]							
		15:8	MASK[15:8]							
		7:0	MASK[7:0]							
0x173C	SCRT2CMP17	31:24								
		23:16								
		15:8								OFFSSTRT[1]
		7:0	OFFSSTRT[0]	OFFSVAL[6:0]						
0x1740	SCRT2CMP08	31:24	COMPVAL[15:8]							
		23:16	COMPVAL[7:0]							
		15:8	MASK[15:8]							
		7:0	MASK[7:0]							
0x1744	SCRT2CMP18	31:24								
		23:16								
		15:8								OFFSSTRT[1]
		7:0	OFFSSTRT[0]	OFFSVAL[6:0]						
0x1748	SCRT2CMP09	31:24	COMPVAL[15:8]							
		23:16	COMPVAL[7:0]							
		15:8	MASK[15:8]							
		7:0	MASK[7:0]							
0x174C	SCRT2CMP19	31:24								
		23:16								
		15:8								OFFSSTRT[1]
		7:0	OFFSSTRT[0]	OFFSVAL[6:0]						
0x1750	SCRT2CMP010	31:24	COMPVAL[15:8]							
		23:16	COMPVAL[7:0]							
		15:8	MASK[15:8]							
		7:0	MASK[7:0]							
0x1754	SCRT2CMP110	31:24								
		23:16								
		15:8								OFFSSTRT[1]
		7:0	OFFSSTRT[0]	OFFSVAL[6:0]						
0x1758	SCRT2CMP011	31:24	COMPVAL[15:8]							
		23:16	COMPVAL[7:0]							
		15:8	MASK[15:8]							
		7:0	MASK[7:0]							
0x175C	SCRT2CMP111	31:24								
		23:16								
		15:8								OFFSSTRT[1]
		7:0	OFFSSTRT[0]	OFFSVAL[6:0]						

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x1760	SCRT2CMP012	31:24	COMPVAL[15:8]							
		23:16	COMPVAL[7:0]							
		15:8	MASK[15:8]							
		7:0	MASK[7:0]							
0x1764	SCRT2CMP112	31:24								
		23:16								
		15:8								OFFSSTRT[1]
		7:0	OFFSSTRT[0]	OFFSVAL[6:0]						
0x1768	SCRT2CMP013	31:24	COMPVAL[15:8]							
		23:16	COMPVAL[7:0]							
		15:8	MASK[15:8]							
		7:0	MASK[7:0]							
0x176C	SCRT2CMP113	31:24								
		23:16								
		15:8								OFFSSTRT[1]
		7:0	OFFSSTRT[0]	OFFSVAL[6:0]						
0x1770	SCRT2CMP014	31:24	COMPVAL[15:8]							
		23:16	COMPVAL[7:0]							
		15:8	MASK[15:8]							
		7:0	MASK[7:0]							
0x1774	SCRT2CMP114	31:24								
		23:16								
		15:8								OFFSSTRT[1]
		7:0	OFFSSTRT[0]	OFFSVAL[6:0]						
0x1778	SCRT2CMP015	31:24	COMPVAL[15:8]							
		23:16	COMPVAL[7:0]							
		15:8	MASK[15:8]							
		7:0	MASK[7:0]							
0x177C	SCRT2CMP115	31:24								
		23:16								
		15:8								OFFSSTRT[1]
		7:0	OFFSSTRT[0]	OFFSVAL[6:0]						
0x1780	SCRT2CMP016	31:24	COMPVAL[15:8]							
		23:16	COMPVAL[7:0]							
		15:8	MASK[15:8]							
		7:0	MASK[7:0]							
0x1784	SCRT2CMP116	31:24								
		23:16								
		15:8								OFFSSTRT[1]
		7:0	OFFSSTRT[0]	OFFSVAL[6:0]						
0x1788	SCRT2CMP017	31:24	COMPVAL[15:8]							
		23:16	COMPVAL[7:0]							
		15:8	MASK[15:8]							
		7:0	MASK[7:0]							
0x178C	SCRT2CMP117	31:24								
		23:16								
		15:8								OFFSSTRT[1]
		7:0	OFFSSTRT[0]	OFFSVAL[6:0]						
0x1790	SCRT2CMP018	31:24	COMPVAL[15:8]							
		23:16	COMPVAL[7:0]							
		15:8	MASK[15:8]							
		7:0	MASK[7:0]							
0x1794	SCRT2CMP118	31:24								
		23:16								
		15:8								OFFSSTRT[1]
		7:0	OFFSSTRT[0]	OFFSVAL[6:0]						
0x1798	SCRT2CMP019	31:24	COMPVAL[15:8]							
		23:16	COMPVAL[7:0]							
		15:8	MASK[15:8]							
		7:0	MASK[7:0]							

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x179C	SCRT2CMP119	31:24								
		23:16								
		15:8								OFFSSTRT[1]
		7:0	OFFSSTRT[0]	OFFSVAL[6:0]						
0x17A0	SCRT2CMP020	31:24	COMPVAL[15:8]							
		23:16	COMPVAL[7:0]							
		15:8	MASK[15:8]							
		7:0	MASK[7:0]							
0x17A4	SCRT2CMP120	31:24								
		23:16								
		15:8								OFFSSTRT[1]
		7:0	OFFSSTRT[0]	OFFSVAL[6:0]						
0x17A8	SCRT2CMP021	31:24	COMPVAL[15:8]							
		23:16	COMPVAL[7:0]							
		15:8	MASK[15:8]							
		7:0	MASK[7:0]							
0x17AC	SCRT2CMP121	31:24								
		23:16								
		15:8								OFFSSTRT[1]
		7:0	OFFSSTRT[0]	OFFSVAL[6:0]						
0x17B0	SCRT2CMP022	31:24	COMPVAL[15:8]							
		23:16	COMPVAL[7:0]							
		15:8	MASK[15:8]							
		7:0	MASK[7:0]							
0x17B4	SCRT2CMP122	31:24								
		23:16								
		15:8								OFFSSTRT[1]
		7:0	OFFSSTRT[0]	OFFSVAL[6:0]						
0x17B8	SCRT2CMP023	31:24	COMPVAL[15:8]							
		23:16	COMPVAL[7:0]							
		15:8	MASK[15:8]							
		7:0	MASK[7:0]							
0x17BC	SCRT2CMP123	31:24								
		23:16								
		15:8								OFFSSTRT[1]
		7:0	OFFSSTRT[0]	OFFSVAL[6:0]						
0x17C0 ... 0x01008B	Reserved									
0x01008C	SAT1	31:24								
		23:16								
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0x010090 ... 0x010093	Reserved									
0x010094	SAT2	31:24								
		23:16								
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0x010098 ... 0x01009B	Reserved									
0x01009C	SAT3	31:24								
		23:16								
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0x0100A0 ... 0x0100A3	Reserved									

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0100A4	SAT4	31:24								
		23:16								
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							

32.8.1. ETH Control A Register

Name: CTRLA
Offset: 0x0000
Reset: 0x00000000
Property: PAC Write Protected

Table 32-16. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

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 is used to keep the ETH running in standby mode.

Value	Description
0	The ETH module is disabled in Standby Sleep mode, clock requests are de-asserted after any pending bus transactions or requests are complete.
1	The ETH module continues to run in Standby Sleep mode.

Bit 1 – ENABLE ETH Clock Enable

Changing the state of this bit from ‘0’ to ‘1’ or ‘1’ to ‘0’ sets the SYNCBUSY.ENABLE bit to 1. The SYNCBUSY.ENABLE bit stays asserted until the module is either completely enabled or completely disabled.

Note: If the ETH is enabled the user should ensure the ETH finishes all tasks before writing this bit to ‘0’.

Value	Description
0	Disable module. System clock is only requested for bus transactions. GCLK is never requested, turn off module, disable clocks, disable interrupt event generation.
1	Enable module by allowing both the generic clock and system clock requests based on the incoming clock requests.

32.8.2. ETH Control B Register

Name: CTRLB
Offset: 0x0004
Reset: 0x000000C0
Property: PAC Write Protected, Enable Write-Protected

Table 32-17. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

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

Bits 7:6 – TSUINC[1:0] Timer Adjust Mode
An alternative way of controlling the way the timer increment register
2'b11 = timer register increments as normal
2'b10 = timer register increments by an additional nanosecond
2'b01 = timer increments by a nanosecond less.
2'b00 = uses TSUINC

Bit 5 – TSUMS Timer Adjust

Value	Description
0	The timer register increments as normal, but the timer value is copied to the sync strobe register
1	The "nanoseconds" timer register is cleared and the "seconds" timer register is incremented with each clock cycle.

Bit 2 – TSUCLKREQ TSU GCLK Request

Value	Description
0	no clock request.
1	GCLK_ETH_TSU clock request

32.8.3. Event Control Register

Name: EVCTRL
Offset: 0x000C
Reset: 0x00000000
Property: PAC Write Protected, Enable Write-Protected

Table 32-18. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

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

Bit 0 – CMPEO Enable the Output of the Time Stamp Compare Event

Value	Description
0	Not Enabled
1	Enabled

32.8.4. SYNCBUSY Register

Name: SYNCB
Offset: 0x0020
Reset: 0x00000000
Property: -

Table 32-19. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
							ENABLE	SWRST
Access							R/HS/HC	R/S/HC
Reset							0	0

Bit 1 – ENABLE Module Enable Synchronization Busy

Value	Description
0	Enable synchronization is not busy
1	Enable synchronization is busy

Bit 0 – SWRST Software Reset Busy bit

Synchronizing Busy bit for CTRLA.SWRST
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.

Notes:

1. When the CTRLA.SWRST is written, the user should poll SYNCB.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.

32.8.5. Network Control Register

Name: NCR
Offset: 0x1000
Reset: 0x00000000
Property: Read/Write

Table 32-20. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					LPI	FNP	TXPBPF	ENPBPR
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
	SRTSM			TXZQPF	TXPF	THALT	TSTART	BP
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
	WESTAT	INCSTAT	CLRSTAT	MPE	TXEN	RXEN	LBL	
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	0	0	0	0	0	0	0	

Bit 19 – LPI Low Power Idle Transmission Enable
When this bit is set, LPI (low power idle) is immediately transmitted. LPI is transmitted even if transmit enable bit NCR.TXEN is disabled.
Setting this bit also sends a pause signal to the transmit datapath.

Bit 18 – FNP Flush Next Packet
Writing a '1' to this bit will flush the next packet from the System RAM. Flushing the next packet will only take effect if the DMA is not currently writing a packet already stored in the DPRAM to memory.

Bit 17 – TXPBPF Transmit PFC Priority-based Pause Frame
Takes the values stored in the Transmit PFC Pause Register.

Bit 16 – ENPBPR Enable PFC Priority-based Pause Reception
Writing a '1' to this bit enables PFC Priority Based Pause Reception capabilities, enabling PFC negotiation and recognition of priority-based pause frames.

Value	Description
0	Normal operation
1	PFC Priority-based Pause frames are recognized.

Bit 15 – SRTSM Store Receive Time Stamp to Memory

Writing a '1' to this bit causes the CRC of every received frame to be replaced with the value of the nanoseconds field of the 1588 timer that was captured as the receive frame passed the message time stamp point.
Note that bit RFCS in register NCFGR may not be set to 1 when the timer should be captured.

Value	Description
0	Normal operation
1	All received frames' CRC is replaced with a time stamp.

Bit 12 – TXZQPF Transmit Zero Quantum Pause Frame

Writing a '1' to this bit causes a pause frame with zero quantum to be transmitted.
Writing a '0' to this bit has no effect.

Bit 11 – TXPF Transmit Pause Frame

Writing one to this bit causes a pause frame to be transmitted.
Writing a '0' to this bit has no effect.

Bit 10 – THALT Transmit Halt

Writing a '1' to this bit halts transmission as soon as any ongoing frame transmission ends.
Writing a '0' to this bit has no effect.

Bit 9 – TSTART Start Transmission

Writing a '1' to this bit starts transmission.
Writing a '0' to this bit has no effect.

Bit 8 – BP Back Pressure

In 10M or 100M half duplex mode, writing a '1' to this bit forces collisions on all received frames.

Value	Description
0	Frame collisions are not forced.
1	Frame collisions are forced in 10M and 100M half duplex mode.

Bit 7 – WESTAT Write Enable for Statistics Registers

Writing a '1' to this bit makes the statistics registers writable for functional test purposes.

Value	Description
0	Statistics Registers are write-protected.
1	Statistics Registers are write-enabled.

Bit 6 – INCSTAT Increment Statistics Registers

Writing a '1' to this bit increments all Statistics Registers by one for test purposes.
Writing a '0' to this bit has no effect.
This bit will always read '0'.

Bit 5 – CLRSTAT Clear Statistics Registers

Writing a '1' to this bit clears the Statistics Registers.
Writing a '0' to this bit has no effect.
This bit will always read '0'.

Bit 4 – MPE Management Port Enable

Writing a '1' to this bit enables the Management Port.
Writing a '0' to this bit disables the Management Port, and forces MDIO to high impedance state and MDC to low impedance.

Value	Description
0	Management Port is disabled.

Value	Description
1	Management Port is enabled.

Bit 3 – TXEN Transmit Enable

Writing a '1' to this bit enables the ETH transmitter to send data.
Writing a '0' to this bit stops transmission immediately, the transmit pipeline and control registers is cleared, and the Transmit Queue Pointer Register will be set to point to the start of the transmit descriptor list.

Note: The TXEN and RXEN bits should not be set (enabled) until all other NCR settings are completed.

Value	Description
0	Transmit is disabled.
1	Transmit is enabled.

Bit 2 – RXEN Receive Enable

Writing a '1' to this bit enables the ETH to receive data.
Writing a '0' to this bit stops frame reception immediately, and the receive pipeline is cleared. The Receive Queue Pointer Register is not affected.

Note: The TXEN and RXEN bits should not be set (enabled) until all other NCR settings are completed.

Value	Description
0	Receive is disabled.
1	Receive is enabled.

Bit 1 – LBL Loop Back Local

Writing '1' to this bit connects ETH_TX to ETH_RX, ETH_TXEN to ETH_RXDV, and forces full duplex mode.
ETH_RXCK and ETH_TXCK may malfunction as the ETH is switched into and out of internal loop back. It is important that receive and transmit circuits have already been disabled when making the switch into and out of internal loop back.

Value	Description
0	Loop back local is disabled.
1	Loop back local is enabled.

32.8.6. ETH Network Configuration Register

Name: NCFGR
Offset: 0x1004
Reset: 0x00080000
Property: Read/Write

Table 32-21. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
		IRXER	RXBP	IPGSEN		IRXFCS	EFRHD	RXCOEN
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
	DCPF	DBW[1:0]		CLK[2:0]			RFCS	LFERD
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	1	0	0	0

Bit	15	14	13	12	11	10	9	8
	RXBUFO[1:0]		PEN	RTY				MAXFS
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
	UNIHEN	MTIHEN	NBC	CAF	JFRAME	DNVLAN	FD	SPD
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 30 – IRXER Ignore IPG RXER
When this bit is written to '1', the Receive Error signal (ETH_RXER) has no effect on the ETH operation when Receive Data Valid signal (ETH_RXDV) is low.

Bit 29 – RXBP Receive Bad Preamble
When written to '1', frames with non-standard preamble are not rejected.

Bit 28 – IPGSEN IP Stretch Enable
Writing a '1' to this bit allows the transmit IPG to increase above 96 bit times, depending on the previous frame length using the IPG Stretch Register.

Bit 26 – IRXFCS Ignore RX FCS
For normal operation this bit must be written to zero.
When this bit is written to '1', frames with FCS/CRC errors will not be rejected. FCS error statistics will still be collected for frames with bad FCS, and FCS status will be recorded in the DMA descriptor of the frame.

Bit 25 – EFRHD Enable Frames Received in half-duplex
Writing a '1' to this bit enables frames to be received in half-duplex mode while transmitting.

Bit 24 – RXCOEN Receive Checksum Offload Enable
Writing a '1' to this bit enables the receive checksum engine, and frames with bad IP, TCP or UDP checksums are discarded.

Bit 23 – DCPF Disable Copy of Pause Frames
Writing a '1' to this bit prevents valid pause frames from being copied to memory. Pause frames are not copied regardless of the state of the Copy All Frames (CAF) bit, whether a hash match is found or whether a type ID match is identified.
If a destination address match is found, the pause frame will be copied to memory. Note that valid pause frames received will still increment pause statistics and pause the transmission of frames, as required.

Bits 22:21 – DBW[1:0] Data Bus Width
Should always be written to '1'.

Value	Name	Description
0	DBW32	32-bit data bus width
1	DBW64	64-bit data bus width

Bits 20:18 – CLK[2:0] MDC Clock Division
These bits must be set according to MCK speed, and determine the number MCK will be divided by to generate Management Data Clock (MDC). For conformance with the 802.3 specification, MDC must not exceed 2.5MHz.
Note: MDC is only active during MDIO read and write operations.

Value	Name	Description
0	MCK_8	MCK divided by 8 (MCK up to 20MHz)
1	MCK_16	MCK divided by 16 (MCK up to 40MHz)
2	MCK_32	MCK divided by 32 (MCK up to 80MHz)
3	MCK_48	MCK divided by 48 (MCK up to 120MHz)
4	MCK_64	MCK divided by 64 (MCK up to 160MHz)
5	MCK_96	MCK divided by 96 (MCK up to 240MHz)
6	MCK_128	MCK divided by 128 (MCK up to 320MHz)
7	MCK_224	MCK divided by 224 (MCK up to 560MHz)

Bit 17 – RFCS Remove FCS
Writing this bit to '1' will cause received frames to be written to memory without their frame check sequence (last 4 bytes). The indicated frame length will be reduced by four bytes in this mode.

Bit 16 – LFERD Length Field Error Frame Discard
Writing a '1' to this bit discards frames with a measured length shorter than the extracted length field (as indicated by bytes 13 and 14 in a non-VLAN tagged frame). This only applies to frames with a length field less than 0x0600.

Bits 15:14 – RXBUFO[1:0] Receive Buffer Offset
These bits determine the number of bytes by which the received data is offset from the start of the receive buffer.

Bit 13 – PEN Pause Enable
When written to '1', transmission will pause if a non-zero 802.3 classic pause frame is received and PFC has not been negotiated.

Bit 12 – RTY Retry Test
This bit must be written to '0' for normal operation.

When writing a '1' to this bit, the back-off between collisions will always be one slot time. This setting helps testing the too many retries condition. This setting is also useful for pause frame tests by reducing the pause counter's decrement time from "512 bit times" to "every RXCK cycle".

Bit 8 – MAXFS 1536 Maximum Frame Size
Writing a '1' to this bit increases the maximum accepted frame size to 1536 bytes in length. When written to '0', any frame above 1518 bytes in length is rejected.

Bit 7 – UNIHEN Unicast Hash Enable
When writing a '1' to this bit, unicast frames will be accepted when the 6-bit hash function of the destination address points to a bit that is set in the Hash Register.
Writing a '0' to this bit disables unicast hashing.

Bit 6 – MTIHEN Multicast Hash Enable
When writing a '1' to this bit, multicast frames will be accepted when the 6-bit hash function of the destination address points to a bit that is set in the Hash Register.
Writing a '0' to this bit disables multicast hashing.

Bit 5 – NBC No Broadcast
Writing a '1' to this bit will reject frames addressed to the broadcast address 0xFFFFFFFFFFFF (all '1').
Writing a '0' to this bit allows broadcasting to 0xFFFFFFFFFFFF.

Bit 4 – CAF Copy All Frames
When writing a '1' to this bit, all valid frames will be accepted.

Bit 3 – JFRAME Jumbo Frame Size
Writing a '1' to this bit enables jumbo frames of up to 10240 bytes to be accepted. The default length is 10240 bytes.

Bit 2 – DNVLAN Discard Non-VLAN Frames
Writing a '1' to this bit allows only VLAN-tagged frames to pass to the address matching logic.
Writing a '0' to this bit allows both VLAN_tagged and untagged frames to pass to the address matching logic.

Bit 1 – FD Full Duplex
Writing a '1' enables full duplex operation, so the transmit block ignores the state of collision and carrier sense and allows receive while transmitting.
Writing a '0' disables full duplex operation.

Bit 0 – SPD Speed
Writing a '1' selects 100Mbps operation.
Writing a '0' to this bit selects 10Mbps operation.

32.8.7. ETH Network Status Register

Name: NSR
Offset: 0x1008
Reset: 0x000001X0
Property: Read-only

Table 32-22. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

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

Bit 2 – IDLE PHY Management Logic Idle
The PHY management logic is idle (i.e., has completed).

Bit 1 – MDIO MDIO Input Status
Returns status of the MDIO pin.

32.8.8. ETH User Register

Name: UR
Offset: 0x100C
Reset: 0x00000000
Property: -

Table 32-23. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

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

Bit 0 – MII Reduced MII Mode

Value	Description
0	RMII mode is selected
1	MII mode is selected

32.8.9. ETH DMA Configuration Register

Name: DCFGR
Offset: 0x1010
Reset: 0x00020004
Property: Read/Write

Table 32-24. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
								DDRP
Access								R/W
Reset								0

Bit	23	22	21	20	19	18	17	16
	DRBS[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	1	0

Bit	15	14	13	12	11	10	9	8
					TXCOEN	TXPBMS	RXBMS[1:0]	
Access					R/W	R/W	R/W	R/W
Reset					0	1	1	1

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

Bit 24 – DDRP DMA Discard Receive Packets

A write to this bit is ignored if the DMA is not configured in the packet buffer full store and forward mode.

Value	Description
0	Received packets are stored in the SRAM based packet buffer until next AXI buffer resource becomes available.
1	Receive packets from the receiver packet buffer memory are automatically discarded when no AXI resource is available.

Bits 23:16 – DRBS[7:0] DMA Receive Buffer Size

These bits defined by these bits determines the size of buffer to use in main AXI system memory when writing received data.

The value is defined in multiples of 64 bytes. For example:

- 0x02: 128 bytes
- 0x18: 1536 bytes (1 × max length frame/buffer)
- 0xA0: 10240 bytes (1 × 10K jumbo frame/buffer)

WARNING Do not write 0x00 to this bit field.

Bit 11 – TXCOEN Transmitter Checksum Generation Offload Enable
Transmitter IP, TCP and UDP checksum generation offload enable.

Value	Description
0	Frame data is unaffected.
1	The transmitter checksum generation engine calculates and substitutes checksums for transmit frames.

Bit 10 – TXPBMS Transmitter Packet Buffer Memory Size Select
When written to zero, the amount of memory used for the transmit packet buffer is reduced by 50%. This reduces the amount of memory used by the ETH.
It is important to write this bit to '1' if the full configured physical memory is available. The value in parentheses represents the size that would result for the default maximum configured memory size of 4 KBytes.

Value	Description
0	Top address bits not used. (2KByte used.)
1	Full configured addressable space (4KBytes) used.

Bits 9:8 – RXBMS[1:0] Receiver Packet Buffer Memory Size Select
The default receive packet buffer size is FULL=8 Kbytes. The table below shows how to configure this memory to FULL, HALF, QUARTER or EIGHTH of the default size.

Value	Name	Description
0	EIGHTH	8/8 Kbyte Memory Size
1	QUARTER	8/4 Kbytes Memory Size
2	HALF	8/2 Kbytes Memory Size
3	FULL	8 Kbytes Memory Size

Bit 7 – ESPA Endian Swap Mode Enable for Packet Data Accesses

Value	Description
0	Little endian mode for AXI transfers selected.
1	Big endian mode for AXI transfers selected.

Bit 6 – ESMA Endian Swap Mode Enable for Management Descriptor Accesses

Value	Description
0	Little endian mode for AXI transfers selected.
1	Big endian mode for AXI transfers selected.

Bits 4:0 – FBLDO[4:0] Fixed Burst Length for DMA Data Operations
Selects the burst length to attempt to use on the AXI when transferring frame data. Not used for DMA management operations and only used where space and data size allow. Otherwise SINGLE type AXI transfers are used.
One-hot priority encoding enforced automatically on register writes as follows. ‘x’ represents don’t care.

Value	Name	Description
0	-	Reserved
1	SINGLE	00001: Always use SINGLE AXI bursts
2	-	Reserved
4	INCR4	001xx: Attempt to use INCR4 AXI bursts (Default)

Value	Name	Description
8	INCR8	01xxx: Attempt to use INCR8 AXI bursts
16	INCR16	1xxxx: Attempt to use INCR16 AXI bursts

32.8.10. ETH Transmit Status Register

Name: TSR
Offset: 0x1014
Reset: 0x00000000
Property: -

Table 32-25. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
								HRESP
Access								R/W
Reset								0

Bit	7	6	5	4	3	2	1	0
		UND	TXCOMP	TFC	TXGO	RLE	COL	UBR
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit 8 – HRESP HRESP Not OK

Set when the DMA block sees HRESP not OK.
This bit is cleared by writing a '1' to it.

Bit 6 – UND Transmit Underrun

This bit is set if the transmitter was forced to terminate a frame that it had already began transmitting due to further data being unavailable. This bit is set if a transmitter status write back has not completed when another status write back is attempted. When using the DMA interface configured for internal FIFO mode, this bit is also set when the transmit DMA has written the SOP data into the FIFO and either the AXI bus was not granted in time for further data, or because an AXI not OK response was returned, or because a used bit was read. When using the DMA interface configured for packet buffer mode, this bit will never be set.

Bit 5 – TXCOMP Transmit Complete

Set when a frame has been transmitted.
This bit is cleared by writing a '1' to it.

Bit 4 – TFC Transmit Frame Corruption Due to AXI Error
This bit is set when an error occurs during reading transmit frame from the AXI. Error causes include HRESP errors and buffers exhausted mid frame. (If the buffers run out during transmission of a frame then transmission stops, FCS shall be bad and TXER asserted).
In DMA packet buffer mode, this bit is also set if a single frame is too large for the configured packet buffer memory size.
This bit is cleared by writing a '1' to it.

Bit 3 – TXGO Transmit Go
This bit is '1' when transmit is active. When using the DMA interface this bit represents the TXGO variable as specified in the transmit buffer description.

Bit 2 – RLE Retry Limit Exceeded
This bit is cleared by writing a '1' to it.

Bit 1 – COL Collision Occurred
When operating in 10/100Mbps mode, this bit is set by the assertion of either a collision or a late collision.
This bit is cleared by writing a '1' to it.

Bit 0 – UBR Used Bit Read
This bit is set when a transmit buffer descriptor is read with its used bit set.
This bit is cleared by writing a '1' to it.

32.8.11. ETH Receive Buffer Queue Base Address Register

Name: RBQB
Offset: 0x1018
Reset: 0x00000000
Property: Read/Write

This register holds the start address of the receive buffer queue (receive buffers descriptor list). The receive buffer queue base address must be initialized before receive is enabled through bit 2 of the Network Control Register. Once reception is enabled, any write to the Receive Buffer Queue Base Address Register is ignored. Reading this register returns the location of the descriptor currently being accessed. This value increments as buffers are used. Software should not use this register for determining where to remove received frames from the queue as it constantly changes as new frames are received. Software should instead work its way through the buffer descriptor queue checking the “used” bits.

In terms of AMBA AXI operation, the descriptors are read from memory using a single 32-bit AXI access. The descriptors should be aligned at 32-bit boundaries and the descriptors are written to using two individual non sequential accesses.

Table 32-26. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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]							
Access	R/W	R/W	R/W	R/W	R/W	R/W		
Reset	0	0	0	0	0	0		

Bits 31:2 – ADDR[29:0] Receive Buffer Queue Base Address
Written with the address of the start of the receive queue.

32.8.12. ETH Transmit Buffer Queue Base Address Register

Name: TBQB
Offset: 0x101C
Reset: 0x00000000
Property: -

This register holds the start address of the transmit buffer queue (transmit buffers descriptor list). The Transmit Buffer Queue Base Address Register must be initialized before transmit is started through bit 9 of the Network Control Register. Once transmission has started, any write to the Transmit Buffer Queue Base Address Register is illegal and therefore ignored.

Note that due to clock boundary synchronization, it takes a maximum of four MCK cycles from the writing of the transmit start bit before the transmitter is active. Writing to the Transmit Buffer Queue Base Address Register during this time may produce unpredictable results.

Reading this register returns the location of the descriptor currently being accessed. Since the DMA handles two frames at once, this may not necessarily be pointing to the current frame being transmitted.

In terms of AMBA AHB operation, the descriptors are written to memory using a single 32-bit AHB access. The descriptors should be aligned at 32-bit boundaries and the descriptors are read from memory using two individual non sequential accesses.

Table 32-27. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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]							
Access	R/W	R/W	R/W	R/W	R/W	R/W		
Reset	0	0	0	0	0	0		

Bits 31:2 – ADDR[29:0] Transmit Buffer Queue Base Address
Written with the address of the start of the transmit queue.

32.8.13. ETH Receive Status Register

Name: RSR
Offset: 0x1020
Reset: 0x00000000
Property: -

This register, when read, provides receive status details. Once read, individual bits may be cleared by writing a '1' to them. It is not possible to set a bit to '1' by writing to this register.

Table 32-28. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
					HNO	RXOVR	REC	BNA
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit 3 – HNO HRESP Not OK
This bit is set when the DMA block sees HRESP not OK.
This bit is cleared by writing a '1' to it.

Bit 2 – RXOVR Receive Overrun
This bit is set if the receive status was not taken at the end of the frame. The buffer will be recovered if an overrun occurs.
This bit is cleared by writing a '1' to it.

Bit 1 – REC Frame Received
This bit is set to when one or more frames have been received and placed in memory.
This bit is cleared by writing a '1' to it.

Bit 0 – BNA Buffer Not Available
When this bit is set, an attempt was made to get a new buffer and the pointer indicated that it was owned by the processor. The DMA will re-read the pointer each time an end of frame is received until a valid pointer is found. This bit is set following each descriptor read attempt that fails, even if consecutive pointers are unsuccessful and software has in the mean time cleared the status flag.

This bit is cleared by writing a '1' to it.

32.8.14. ETH Interrupt Status Register

Name: ISR
Offset: 0x1024
Reset: 0x00000000
Property: Write to Clear

This register indicates the source of the interrupt. An interrupt source must be enabled in the mask register first so the corresponding bits of this register will be set and the ETH interrupt signal will be asserted in the system.

Note: Interrupt flags must be cleared and then read back to confirm the clear before exiting the ISR to avoid double interrupts.

Table 32-29. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
			TSUTIMCMP	WOL	RXLPISBC	SRI	PDRSFT	PDRQFT
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	PDRSFR	PDRQFR	SFT	DRQFT	SFR	DRQFR		
Access	R	R	R	R	R	R		
Reset	0	0	0	0	0	0		

Bit	15	14	13	12	11	10	9	8
		PFTR	PTZ	PFNZ	HRESP	ROVR		
Access		R	R	R	R	R		
Reset		0	0	0	0	0		

Bit	7	6	5	4	3	2	1	0
	TCOMP	TFC	RLEX	TUR	TXUBR	RXUBR	RCOMP	MFS
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 29 – TSUTIMCMP TSU Timer Comparison
Indicates when TSU timer count value is equal to programmed value.
Cleared on read.

Bit 28 – WOL Wake On LAN
WOL interrupt. Indicates a WOL message has been received.

Bit 27 – RXLPISBC Receive LPI indication Status Bit Change
Receive LPI indication status bit change.
Cleared on read.

Bit 26 – SRI TSU Seconds Register Increment
Indicates the register has incremented.
Cleared on read.

Bit 25 – PDRSFT PDelay Response Frame Transmitted
Indicates a PTP pdelay_resp frame has been transmitted.
Cleared on read.

Bit 24 – PDRQFT PDelay Request Frame Transmitted
Indicates a PTP pdelay_req frame has been transmitted.
Cleared on read.

Bit 23 – PDRSFR PDelay Response Frame Received
Indicates a PTP pdelay_resp frame has been received.
Cleared on read.

Bit 22 – PDRQFR PDelay Request Frame Received
Indicates a PTP pdelay_req frame has been received.
Cleared on read.

Bit 21 – SFT PTP Sync Frame Transmitted
Indicates a PTP sync frame has been transmitted.
Cleared on read.

Bit 20 – DRQFT PTP Delay Request Frame Transmitted
Indicates a PTP delay_req frame has been transmitted.
Cleared on read.

Bit 19 – SFR PTP Sync Frame Received
Indicates a PTP sync frame has been received.
Cleared on read.

Bit 18 – DRQFR PTP Delay Request Frame Received
Indicates a PTP delay_req frame has been received.
Cleared on read.

Bit 14 – PFTR Pause Frame Transmitted
Indicates a pause frame has been successfully transmitted after being initiated from the Network Control Register.
Cleared on read.

Bit 13 – PTZ Pause Time Zero
Set when either the Pause Time Register at address 0x1038 decrements to zero, or when a valid pause frame is received with a zero pause quantum field.
Cleared on read.

Bit 12 – PFNZ Pause Frame with Non-zero Pause Quantum Received
Indicates a valid pause has been received that has a non-zero pause quantum field.
Cleared on read.

Bit 11 – HRESP HRESP Not OK
Set when the DMA block sees HRESP not OK.
Cleared on read.

Bit 10 – ROVR Receive Overrun
Set when the receive overrun status bit is set.
Cleared on read.

Bit 7 – TCOMP Transmit Complete
Set when a frame has been transmitted.
Cleared on read.

Bit 6 – TFC Transmit Frame Corruption Due to AXI Error
Transmit frame corruption due to AXI error. Set if an error occurs during reading a transmit frame from the AXI, including HRESP errors and buffers exhausted mid frame.

Bit 5 – RLEX Retry Limit Exceeded
Retry Limit Exceeded Transmit error.
Cleared on read.

Bit 4 – TUR Transmit Underrun
This interrupt is set if the transmitter was forced to terminate an ongoing frame transmission due to further data being unavailable.
This interrupt is also set if a transmitter status write back has not completed when another status write back is attempted.
This interrupt is also set when the transmit DMA has written the SOP data into the FIFO and either the AXI bus was not granted in time for further data, or because an AXI not OK response was returned, or because the used bit was read.

Bit 3 – TXUBR TX Used Bit Read
Set when a transmit buffer descriptor is read with its used bit set.
Cleared on read.

Bit 2 – RXUBR RX Used Bit Read
Set when a receive buffer descriptor is read with its used bit set.
Cleared on read.

Bit 1 – RCOMP Receive Complete
A frame has been stored in memory.
Cleared on read.

Bit 0 – MFS Management Frame Sent
The PHY Maintenance Register has completed its operation.
Cleared on read.

32.8.15. ETH Interrupt Enable Register

Name: IER
Offset: 0x1028
Reset: –
Property: Write-only

This register is write-only and will always return zero.

The following values are valid for all listed bit names of this register:

0: No effect.

1: Enables the corresponding interrupt.

Table 32-30. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
			TSUTIMCMP	WOL	RXLPISBC	SRI	PDRSFT	PDRQFT
Access			W	W	R	W	W	W
Reset			–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	PDRSFR	PDRQFR	SFT	DRQFT	SFR	DRQFR		
Access	W	W	W	W	W	W		
Reset	–	–	–	–	–	–		

Bit	15	14	13	12	11	10	9	8
	EXINT	PFTR	PTZ	PFNZ	HRESP	ROVR		
Access	W	W	W	W	W	W		
Reset	–	–	–	–	–	–		

Bit	7	6	5	4	3	2	1	0
	TCOMP	TFC	RLEX	TUR	TXUBR	RXUBR	RCOMP	MFS
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit 29 – TSUTIMCMP TSU Timer Comparison

Bit 28 – WOL Wake On LAN

Bit 27 – RXLPISBC Receive LPI indication Status Bit Change
Receive LPI indication status bit change.
Cleared on read.

Bit 26 – SRI TSU Seconds Register Increment

Bit 25 – PDRSFT PDelay Response Frame Transmitted

Bit 24 – PDRQFT PDelay Request Frame Transmitted

Bit 23 – PDRSFR PDelay Response Frame Received

Bit 22 – PDRQFR PDelay Request Frame Received

Bit 21 – SFT PTP Sync Frame Transmitted

Bit 20 – DRQFT PTP Delay Request Frame Transmitted

Bit 19 – SFR PTP Sync Frame Received

Bit 18 – DRQFR PTP Delay Request Frame Received

Bit 15 – EXINT External Interrupt

Bit 14 – PFTR Pause Frame Transmitted

Bit 13 – PTZ Pause Time Zero

Bit 12 – PFNZ Pause Frame with Non-zero Pause Quantum Received

Bit 11 – HRESP HRESP Not OK

Bit 10 – ROVR Receive Overrun

Bit 7 – TCOMP Transmit Complete

Bit 6 – TFC Transmit Frame Corruption Due to AXI Error

Bit 5 – RLEX Retry Limit Exceeded or Late Collision

Bit 4 – TUR Transmit Underrun

Bit 3 – TXUBR TX Used Bit Read

Bit 2 – RXUBR RX Used Bit Read

Bit 1 – RCOMP Receive Complete

Bit 0 – MFS Management Frame Sent

32.8.16. ETH Interrupt Disable Register

Name: IDR
Offset: 0x102C
Reset: –
Property: Write-only

This register is write-only and will always return zero.

The following values are valid for all listed bit names of this register:

0: No effect.

1: Disables the corresponding interrupt.

Table 32-31. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
			TSUTIMCMP	WOL	RXLPISBC	SRI	PDRSFT	PDRQFT
Access			W	W	R	W	W	W
Reset			–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	PDRSFR	PDRQFR	SFT	DRQFT	SFR	DRQFR		
Access	W	W	W	W	W	W		
Reset	–	–	–	–	–	–		

Bit	15	14	13	12	11	10	9	8
	EXINT	PFTR	PTZ	PFNZ	HRESP	ROVR		
Access	W	W	W	W	W	W		
Reset	–	–	–	–	–	–		

Bit	7	6	5	4	3	2	1	0
	TCOMP	TFC	RLEX	TUR	TXUBR	RXUBR	RCOMP	MFS
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit 29 – TSUTIMCMP TSU Timer Comparison

Bit 28 – WOL Wake On LAN

Bit 27 – RXLPISBC Receive LPI indication Status Bit Change
Receive LPI indication status bit change.
Cleared on read.

Bit 26 – SRI TSU Seconds Register Increment

Bit 25 – PDRSFT PDelay Response Frame Transmitted

Bit 24 – PDRQFT PDelay Request Frame Transmitted

Bit 23 – PDRSFR PDelay Response Frame Received

Bit 22 – PDRQFR PDelay Request Frame Received

Bit 21 – SFT PTP Sync Frame Transmitted

Bit 20 – DRQFT PTP Delay Request Frame Transmitted

Bit 19 – SFR PTP Sync Frame Received

Bit 18 – DRQFR PTP Delay Request Frame Received

Bit 15 – EXINT External Interrupt

Bit 14 – PFTR Pause Frame Transmitted

Bit 13 – PTZ Pause Time Zero

Bit 12 – PFNZ Pause Frame with Non-zero Pause Quantum Received

Bit 11 – HRESP HRESP Not OK

Bit 10 – ROVR Receive Overrun

Bit 7 – TCOMP Transmit Complete

Bit 6 – TFC Transmit Frame Corruption Due to AXI Error

Bit 5 – RLEX Retry Limit Exceeded or Late Collision

Bit 4 – TUR Transmit Underrun

Bit 3 – TXUBR TX Used Bit Read

Bit 2 – RXUBR RX Used Bit Read

Bit 1 – RCOMP Receive Complete

Bit 0 – MFS Management Frame Sent

32.8.17. ETH Interrupt Mask Register

Name: IMR
Offset: 0x1030
Reset: 0x07FFFFFF
Property: Read/Write

This register is a read-only register indicating which interrupts are masked. All bits are set at Reset and can be reset individually by writing to the Interrupt Enable Register (IER), or set individually by writing to the Interrupt Disable Register (IDR).

For test purposes there is a write-only function to this register that allows the bits in the Interrupt Status Register to be set or cleared, regardless of the state of the mask register. A write to this register directly affects the state of the corresponding bit in the Interrupt Status Register, causing an interrupt to be generated if a 1 is written.

The following values are valid for all listed bit names of this register when read:

0: The corresponding interrupt is enabled.

1: The corresponding interrupt is not enabled.

Table 32-32. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
			TSUTIMCMP	WOL	RXLPISBC	SRI	PDRSFT	PDRQFT
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	1	1	1

Bit	23	22	21	20	19	18	17	16
	PDRSFR	PDRQFR	SFT	DRQFT	SFR	DRQFR		
Access	R/W	R/W	R/W	R/W	R/W	R/W		
Reset	1	1	1	1	1	1		

Bit	15	14	13	12	11	10	9	8
	EXINT	PFTR	PTZ	PFNZ	HRESP	ROVR		
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
	TCOMP	TFC	RLEX	TUR	TXUBR	RXUBR	RCOMP	MFS
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 29 – TSUTIMCMP TSU Timer Comparison

Bit 28 – WOL Wake On LAN

Bit 27 – RXLPISBC Receive LPI indication Status Bit Change
Receive LPI indication status bit change.
Cleared on read.

Bit 26 – SRI TSU Seconds Register Increment

Bit 25 – PDRSFT PDelay Response Frame Transmitted

Bit 24 – PDRQFT PDelay Request Frame Transmitted

Bit 23 – PDRSFR PDelay Response Frame Received

Bit 22 – PDRQFR PDelay Request Frame Received

Bit 21 – SFT PTP Sync Frame Transmitted

Bit 20 – DRQFT PTP Delay Request Frame Transmitted

Bit 19 – SFR PTP Sync Frame Received

Bit 18 – DRQFR PTP Delay Request Frame Received

Bit 15 – EXINT External Interrupt

Bit 14 – PFTR Pause Frame Transmitted

Bit 13 – PTZ Pause Time Zero

Bit 12 – PFNZ Pause Frame with Non-zero Pause Quantum Received

Bit 11 – HRESP HRESP Not OK

Bit 10 – ROVR Receive Overrun

Bit 7 – TCOMP Transmit Complete

Bit 6 – TFC Transmit Frame Corruption Due to AXI Error

Bit 5 – RLEX Retry Limit Exceeded

Bit 4 – TUR Transmit Underrun

Bit 3 – TXUBR TX Used Bit Read

Bit 2 – RXUBR RX Used Bit Read

Bit 1 – RCOMP Receive Complete

Bit 0 – MFS Management Frame Sent

32.8.18. ETH PHY Maintenance Register

Name: MAN
Offset: 0x1034
Reset: 0x00000000
Property: Read/Write

This register is a shift register. Writing to it starts a shift operation which is signaled completed when bit 2 is set in the Network Status Register (NSR). It takes about 2000 MCK cycles to complete, when MDC is set for MCK divide by 32 in the Network Configuration Register. An interrupt is generated upon completion.

During this time, the MSB of the register is output on the MDIO pin and the LSB updated from the MDIO pin with each MDC cycle. This causes transmission of a PHY management frame on MDIO. Refer also to section 22.2.4.5 of the IEEE 802.3 standard.

Reading during the shift operation returns the current contents of the shift register. At the end of management operation, the bits will have shifted back to their original locations. For a read operation, the data bits are updated with data read from the PHY. It is important to write the correct values to the register to ensure a valid PHY management frame is produced.

The MDIO interface can read IEEE 802.3 clause 45 PHYs, as well as clause 22 PHYs. To read clause 45 PHYs, bit 30 should be written with a '0' rather than a '1'. To write clause 45 PHYs, bits 31:28 should be written as 0x1:

PHY	Access	Bit Value			
		WZO	CLTTO	OP[1]	OP[0]
Clause 22	Read	0	1	1	0
	Write	0	1	0	1
Clause 45	Read	0	0	1	1
	Write	0	0	0	1
	Read + Address	0	0	1	0

For a description of MDC generation, see also the 'ETH Network Configuration Register' (NCR) description.

Table 32-33. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	WZO	CLTTO	OP[1:0]		PHYA[4:1]			
Access	R/W	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
	PHYA[0]	REGA[4:0]					WTN[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	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

Bit 31 – WZO Write ZERO
Must be written to '0'.

Value	Description
0	Mandatory
1	Reserved

Bit 30 – CLTTO Clause 22 Operation

Value	Description
0	Clause 45 operation
1	Clause 22 operation

Bits 29:28 – OP[1:0] Operation

Value	Description
01	Write
10	Read
Other	Reseved

Bits 27:23 – PHYA[4:0] PHY Address

Bits 22:18 – REGA[4:0] Register Address
Specifies the register in the PHY to access.

Bits 17:16 – WTN[1:0] Write Ten
Must be written to '10'.

Value	Description
10	Mandatory
Other	Reserved

Bits 15:0 – DATA[15:0] PHY Data
For a write operation, this field is written with the data to be written to the PHY.
After a read operation, this field contains the data read from the PHY.

32.8.19. ETH Receive Pause Quantum Register

Name: RPQ
Offset: 0x1038
Reset: 0x00000000
Property: -

Table 32-34. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
	RPQ[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
	RPQ[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – RPQ[15:0] Received Pause Quantum
Stores the current value of the Receive Pause Quantum Register which is decremented every 512 bit times.

32.8.20. ETH Transmit Pause Quantum Register

Name: TPQ
Offset: 0x103C
Reset: 0x0000FFFF
Property: -

Table 32-35. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
	TPQ[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
	TPQ[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 – TPQ[15:0] Transmit Pause Quantum
Written with the pause quantum value for pause frame transmission.

32.8.21. ETH TX Partial Store and Forward Register

Name: TPSF
Offset: 0x1040
Reset: 0x000003FF
Property: -

Table 32-36. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	ENTXP							
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
					TPB1ADR[11:8]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	1	1

Bit	7	6	5	4	3	2	1	0
	TPB1ADR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bit 31 – ENTXP Enable TX Partial Store and Forward Operation

Bits 11:0 – TPB1ADR[11:0] Transmit Partial Store and Forward Address
Watermark value.
This value must be ≥ 0x14.

32.8.22. ETH RX Partial Store and Forward Register

Name: RPSF
Offset: 0x1044
Reset: 0x000003FF
Property: -

Table 32-37. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	ENRXP							
Access	R							
Reset	0							

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
					RPB1ADR[11:8]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	1	1

Bit	7	6	5	4	3	2	1	0
	RPB1ADR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bit 31 – ENRXP Enable RX Partial Store and Forward Operation

Bits 11:0 – RPB1ADR[11:0] Receive Partial Store and Forward Address
Watermark value. Reset = 1.

32.8.23. ETH RX Jumbo Frame Max Length Register

Name: RJFML
Offset: 0x1048
Reset: 0x00002800
Property: Read/Write

Table 32-38. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
			FML[13:8]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			1	0	1	0	0	0

Bit	7	6	5	4	3	2	1	0
	FML[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 13:0 – FML[13:0] Frame Max Length
Rx jumbo frame maximum length.

32.8.24. ETH Hash Register Bottom

Name: HRB
Offset: 0x1080
Reset: 0x00000000
Property: Read/Write

The unicast hash enable (UNIHEN) and the multicast hash enable (MITIHEN) bits in the Network Configuration Register (NCFGR) enable the reception of hash matched frames.

Table 32-39. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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

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

Bit	15	14	13	12	11	10	9	8
	ADDR[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

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

Bits 31:0 – ADDR[31:0] Hash Address
The first 32 bits of the Hash Address Register.

32.8.25. ETH Hash Register Top

Name: HRT
Offset: 0x1084
Reset: 0x00000000
Property: Read/Write

The Unicast Hash Enable (UNIHEN) and the Multicast Hash Enable (MITIHEN) bits in the Network Configuration Register (NCFGR) enable the reception of hash matched frames.

Table 32-40. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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

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

Bit	15	14	13	12	11	10	9	8
	ADDR[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

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

Bits 31:0 – ADDR[31:0] Hash Address
Bits 63 to 32 of the Hash Address Register.

32.8.26. ETH Specific Address n Bottom Register

Name: SABx
Offset: 0x1088 + (x-1)*0x08 [x=1..4]
Reset: 0x00000000
Property: Read/Write

The addresses stored in the Specific Address Registers are deactivated at reset or when their corresponding Specific Address Register Bottom is written. They are activated when Specific Address Register Top is written.

Table 32-41. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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

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

Bit	15	14	13	12	11	10	9	8
	ADDR[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

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

Bits 31:0 – ADDR[31:0] Specific Address n
Least significant 32 bits of the destination address, that is, bits 31:0. Bit zero indicates whether the address is multicast or unicast and corresponds to the least significant bit of the first byte received.

32.8.27. ETH Specific Address n Top Register

Name: SATx
Offset: 0x01008C + (x-1)*0x08 [x=1..4]
Reset: 0x00000000
Property: Read/Write

The addresses stored in the Specific Address Registers are deactivated at reset or when their corresponding Specific Address Register Bottom is written. They are activated when Specific Address Register Top is written.

Table 32-42. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
	ADDR[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

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

Bits 15:0 – ADDR[15:0] Specific Address n
The most significant bits of the destination address, that is, bits 47:32.

32.8.28. ETH Type ID Match n Register

Name: TIDMx
Offset: 0x10A8 + (x-1)*0x04 [x=1..4]
Reset: 0x00000000
Property: Read/Write

Table 32-43. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	ENIDn							
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
	TID[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
	TID[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 31 – ENIDn Enable Copying of TID Matched Frames

Value	Description
0	TID n is not part of the comparison match.
1	TID n is processed for the comparison match.

Bits 15:0 – TID[15:0] Type ID Match n
For use in comparisons with received frames type ID/length frames.

32.8.29. ETH Wake on LAN Register

Name: WOL
Offset: 0x10B8
Reset: 0x00000000
Property: -

Table 32-44. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					MTI	SA1	ARP	MAG
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
	IP[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
	IP[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 19 – MTI Multicast Hash Event Enable

Value	Description
0	Wake on LAN multicast hash Event disabled
1	Wake on LAN multicast hash Event enabled

Bit 18 – SA1 Specific Address Register 1 Event Enable

Value	Description
0	Wake on Specific Address Register 1 Event disabled
1	Wake on Specific Address Register 1 Event enabled

Bit 17 – ARP ARP Request Event Enable

Value	Description
0	Wake on LAN ARP request Event disabled
1	Wake on LAN ARP request Event enabled

Bit 16 – MAG Magic Packet Event Enable

Value	Description
0	Wake on LAN magic packet Event disabled

Value	Description
1	Wake on LAN magic packet Event enabled

Bits 15:0 – IP[15:0] ARP Request IP Address

Wake on LAN ARP request IP address. Written to define the 16 least significant bits of the target IP address that is matched to generate a Wake on LAN event.

Value	Description
0x0000	No Event generated, even if matched by the received frame.
0x0001-0xFFFF	Wake on LAN Event generated for matching LSB of the target IP address.

32.8.30. ETH IPG Stretch Register

Name: IPGS
Offset: 0x10BC
Reset: 0x00000000
Property: -

Table 32-45. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
	FL[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
	FL[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 – FL[15:0] Frame Length

Bits FL[7:0] are multiplied with the previously transmitted frame length (including preamble), and divided by FL[15:8]+1 (adding 1 to prevent division by zero). $\mathrm{RESULT} = \frac{\mathrm{FL[7:0]}}{\mathrm{F[15+8]+1}}$

If RESULT > 96 and the IP Stretch Enable bit in the Network Configuration Register (NCFGR.IPGSEN) is written to '1', RESULT is used for the transmit inter-packet-gap.

32.8.31. ETH Stacked VLAN Register

Name: SVLAN
Offset: 0x10C0
Reset: 0x00000000
Property: -

Table 32-46. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	ESVLAN							
Access	-							
Reset	0							

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	VLAN_TYPE[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
	VLAN_TYPE[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 31 – ESVLAN Enable Stacked VLAN Processing Mode
0: Disable the stacked VLAN processing mode
1: Enable the stacked VLAN processing mode

Value	Description
0	Stacked VLAN Processing disabled
1	Stacked VLAN Processing enabled

Bits 15:0 – VLAN_TYPE[15:0] User Defined VLAN_TYPE Field
When Stacked VLAN is enabled (ESVLAN=1), the first VLAN tag in a received frame will only be accepted if the VLAN type field is equal to this user defined VLAN_TYPE, OR equal to the standard VLAN type (0x8100).
Note: The second VLAN tag of a Stacked VLAN packet will only be matched correctly if its VLAN_TYPE field equals 0x8100.

32.8.32. ETH Transmit PFC Pause Register

Name: TPFCP
Offset: 0x10C4
Reset: 0x00000000
Property: -

Table 32-47. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
	PQ[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
	PEV[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:8 – PQ[7:0] Pause Quantum

When the Remove FCS bit in the ETH Network Configuration register (NCFGR.RFCS) is written to '1', and one or more bits in this bit field are written to '0', the associated PFC pause frame's pause quantum field value is taken from the Transmit Pause Quantum register (TPQ).
For each entry equal to '1' in this bit field, the pause quantum associated with that entry will be zero.

Bits 7:0 – PEV[7:0] Priority Enable Vector

When the Remove FCS bit in the ETH Network Configuration register (NCFGR.RFCS) is written to '1', the priority enable vector of the PFC priority-based pause frame is set to the value stored in this bit field.

32.8.33. ETH Specific Address 1 Mask Bottom

Name: SAMB1
Offset: 0x10C8
Reset: 0x00000000
Property: -

Table 32-48. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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

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

Bit	15	14	13	12	11	10	9	8
	ADDR[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

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

Bits 31:0 – ADDR[31:0] Specific Address 1 Mask
Setting a bit to '1' masks the corresponding bit in the Specific Address 1 Bottom register (SAB1).

32.8.34. ETH Specific Address Mask 1 Top

Name: SAMT1
Offset: 0x10CC
Reset: 0x00000000
Property: -

Table 32-49. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
	ADDR[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

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

Bits 15:0 – ADDR[15:0] Specific Address 1 Mask
Setting a bit to '1' masks the corresponding bit in the Specific Address 1 register SAT1.

32.8.35. ETH 1588 Timer Nanosecond Comparison Register

Name: NSC
Offset: 0x10DC
Reset: 0x00000000
Property: -

Table 32-50. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
			NANOSEC[21:16]					
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
	NANOSEC[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
	NANOSEC[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 21:0 – NANOSEC[21:0] 1588 Timer Nanosecond Comparison Value
Value is compared to the bits [45:24] of the TSU timer count value (upper 22 bits of nanosecond value).

32.8.36. ETH 1588 Timer Second Comparison Low Register

Name: SCL
Offset: 0x10E0
Reset: 0x00000000
Property: -

Table 32-51. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	SEC[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
	SEC[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
	SEC[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
	SEC[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 – SEC[31:0] 1588 Timer Second Comparison Value
Value is compared to seconds value bits [31:0] of the TSU timer count value.

32.8.37. ETH 1588 Timer Second Comparison High Register

Name: SCH
Offset: 0x10E4
Reset: 0x00000000
Property: -

Table 32-52. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
	SEC[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
	SEC[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 – SEC[15:0] 1588 Timer Second Comparison Value
Value is compared to the top 16 bits (most significant 16 bits [47:32] of seconds value) of the TSU timer count value.

32.8.38. ETH PTP Event Frame Transmitted Seconds High Register

Name: EFTSH
Offset: 0x10E8
Reset: 0x00000000
Property: Read-only

Table 32-53. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
	RUD[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
	RUD[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – RUD[15:0] Register Update
The register is updated with the value that the IEEE 1588 timer seconds register held when the SFD of a PTP transmit primary event crosses the MII interface. An interrupt is issued when the register is updated.

32.8.39. ETH PTP Event Frame Received Seconds High Register

Name: EFRSH
Offset: 0x10EC
Reset: 0x00000000
Property: Read-only

Table 32-54. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
	RUD[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
	RUD[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – RUD[15:0] Register Update
The register is updated with the value that the IEEE 1588 timer seconds register held when the SFD of a PTP transmit primary event crosses the MII interface. An interrupt is issued when the register is updated.

32.8.40. ETH PTP Peer Event Frame Transmitted Seconds High Register

Name: PEFTSH
Offset: 0x10F0
Reset: 0x00000000
Property: -

Table 32-55. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
	RUD[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
	RUD[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – RUD[15:0] Register Update
The register is updated with the value that the IEEE 1588 timer seconds register held when the SFD of a PTP transmit peer event crosses the MII interface. An interrupt is issued when the register is updated.

32.8.41. ETH PTP Peer Event Frame Received Seconds High Register

Name: PEFRSH
Offset: 0x10F4
Reset: 0x00000000
Property: -

Table 32-56. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
	RUD[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
	RUD[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – RUD[15:0] Register Update
The register is updated with the value that the 1588 timer seconds register held when the SFD of a PTP transmit peer event crosses the MII interface. An interrupt is issued when the register is updated.

32.8.42. ETH Octets Transmitted Low Register

Name: OTLO
Offset: 0x1100
Reset: 0x00000000
Property: Read-Only(Cleared on Read)

When reading the Octets Transmitted and Octets Received Registers, bits [31:0] should be read prior to bits [47:32] to ensure reliable operation.

Table 32-57. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	TXO[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
	TXO[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
	TXO[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
	TXO[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – TXO[31:0] Transmitted Octets
Transmitted octets in valid frames of any type without errors, bits [31:0]. This counter is 48-bits, and is read through two registers. This count does not include octets from automatically generated pause frames.

32.8.43. ETH Octets Transmitted High Register

Name: OTHI
Offset: 0x1104
Reset: 0x00000000
Property: Read-Only(Cleared on Read)

When reading the Octets Transmitted and Octets Received Registers, bits [31:0] should be read prior to bits [47:32] to ensure reliable operation.

Table 32-58. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
	TXO[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
	TXO[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – TXO[15:0] Transmitted Octets
Transmitted octets in valid frames of any type without errors, bits [47:32]. This counter is 48-bits, and is read through two registers. This count does not include octets from automatically generated pause frames.

32.8.44. ETH Frames Transmitted

Name: FT
Offset: 0x1108
Reset: 0x00000000
Property: Read-only(Cleared on Read)

Table 32-59. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	FTX[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
	FTX[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
	FTX[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
	FTX[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – FTX[31:0] Frames Transmitted without Error
Frames transmitted without error. This register counts the number of frames successfully transmitted, i.e., no underrun and not too many retries. Excludes pause frames.

32.8.45. ETH Broadcast Frames Transmitted Register

Name: BCFT
Offset: 0x110C
Reset: 0x00000000
Property: Read-only(Cleared on Read)

Table 32-60. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	BFTX[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
	BFTX[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
	BFTX[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
	BFTX[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – BFTX[31:0] Broadcast Frames Transmitted without Error
This register counts the number of broadcast frames successfully transmitted without error, i.e., no underrun and not too many retries. Excludes pause frames.

32.8.46. ETH Multicast Frames Transmitted Register

Name: MFT
Offset: 0x1110
Reset: 0x00000000
Property: Read-Only(Cleared on Read)

Table 32-61. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	MFTX[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
	MFTX[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
	MFTX[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
	MFTX[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – MFTX[31:0] Multicast Frames Transmitted without Error
This register counts the number of multicast frames successfully transmitted without error, i.e., no underrun and not too many retries. Excludes pause frames.

32.8.47. ETH Pause Frames Transmitted Register

Name: PFT
Offset: 0x1114
Reset: 0x00000000
Property: -

Table 32-62. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
	PFTX[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
	PFTX[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – PFTX[15:0] Pause Frames Transmitted Register
This register counts the number of pause frames transmitted. Only pause frames triggered by the register interface or through the external pause pins are counted as pause frames. Pause frames received through the FIFO interface are counted in the frames transmitted counter.

32.8.48. ETH 64-Byte Frames Transmitted Register

Name: BFT64
Offset: 0x1118
Reset: 0x00000000
Property: Read-only

Table 32-63. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	NFTX[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
	NFTX[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
	NFTX[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
	NFTX[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – NFTX[31:0] 64-Byte Frames Transmitted without Error
This register counts the number of 64-byte frames successfully transmitted without error, i.e., no underrun and not too many retries. Excludes pause frames.

32.8.49. ETH 65 to 127 Byte Frames Transmitted Register

Name: TBFT127
Offset: 0x101C
Reset: 0x00000000
Property: -

Table 32-64. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	NFTX[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
	NFTX[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
	NFTX[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
	NFTX[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – NFTX[31:0] 65 to 127 Byte Frames Transmitted without Error
This register counts the number of 65 to 127 byte frames successfully transmitted without error, i.e., no underrun and not too many retries. Excludes pause frames.

32.8.50. ETH 128 to 255 Byte Frames Transmitted Register

Name: TBFT255
Offset: 0x1120
Reset: 0x00000000
Property: -

Table 32-65. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	NFTX[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
	NFTX[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
	NFTX[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
	NFTX[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – NFTX[31:0] 128 to 255 Byte Frames Transmitted without Error
This register counts the number of 128 to 255 byte frames successfully transmitted without error, i.e., no underrun and not too many retries.

32.8.51. ETH 256 to 511 Byte Frames Transmitted Register

Name: TBFT511
Offset: 0x1124
Reset: 0x00000000
Property: -

Table 32-66. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	NFTX[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
	NFTX[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
	NFTX[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
	NFTX[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – NFTX[31:0] 256 to 511 Byte Frames Transmitted without Error
This register counts the number of 256 to 511 byte frames successfully transmitted without error, i.e., no underrun and not too many retries.

32.8.52. ETH 512 to 1023 Byte Frames Transmitted Register

Name: TBFT1023
Offset: 0x1128
Reset: 0x00000000
Property: -

Table 32-67. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	NFTX[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
	NFTX[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
	NFTX[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
	NFTX[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – NFTX[31:0] 512 to 1023 Byte Frames Transmitted without Error
This register counts the number of 512 to 1023 byte frames successfully transmitted without error, i.e., no underrun and not too many retries.

32.8.53. ETH 1024 to 1518 Byte Frames Transmitted Register

Name: TBFT1518
Offset: 0x112C
Reset: 0x00000000
Property: -

Table 32-68. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	NFTX[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
	NFTX[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
	NFTX[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
	NFTX[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – NFTX[31:0] 1024 to 1518 Byte Frames Transmitted without Error
This register counts the number of 1024 to 1518 byte frames successfully transmitted without error, i.e., no underrun and not too many retries.

32.8.54. ETH Greater Than 1518 Byte Frames Transmitted Register

Name: GTBFT1518
Offset: 0x1130
Reset: 0x00000000
Property: Read-only

Table 32-69. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	NFTX[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
	NFTX[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
	NFTX[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
	NFTX[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – NFTX[31:0] Greater than 1518 Byte Frames Transmitted without Error
This register counts the number of 1518 or above byte frames successfully transmitted without error i.e., no underrun and not too many retries.

32.8.55. ETH Transmit Underruns Register

Name: TUR
Offset: 0x1134
Reset: 0x00000000
Property: -

Table 32-70. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
							TXUNR[9:8]	
Access							R	R
Reset							0	0

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

Bits 9:0 – TXUNR[9:0] Transmit Underruns
This register counts the number of frames not transmitted due to a transmit underrun. If this register is incremented then no other statistics register is incremented.

32.8.56. ETH Single Collision Frames Register

Name: SCF
Offset: 0x1138
Reset: 0x00000000
Property: -

Table 32-71. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
							SCOL[17:16]	
Access							R	R
Reset							0	0

Bit	15	14	13	12	11	10	9	8
	SCOL[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
	SCOL[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 17:0 – SCOL[17:0] Single Collision
This register counts the number of frames experiencing a single collision before being successfully transmitted i.e., no underrun.

32.8.57. ETH Multiple Collision Frames Register

Name: MCF
Offset: 0x113C
Reset: 0x00000000
Property: -

Table 32-72. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
							MCOL[17:16]	
Access							R	R
Reset							0	0

Bit	15	14	13	12	11	10	9	8
	MCOL[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
	MCOL[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 17:0 – MCOL[17:0] Multiple Collision
This register counts the number of frames experiencing between two and fifteen collisions prior to being successfully transmitted, i.e., no underrun and not too many retries.

32.8.58. ETH Excessive Collisions Register

Name: EC
Offset: 0x1140
Reset: 0x00000000
Property: Read-only

Table 32-73. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
							XCOL[9:8]	
Access							R	R
Reset							0	0

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

Bits 9:0 – XCOL[9:0] Excessive Collisions
This register counts the number of frames that failed to be transmitted because they experienced 16 collisions.

32.8.59. ETH Late Collisions Register

Name: LC
Offset: 0x1144
Reset: 0x00000000
Property: -

Table 32-74. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
							LCOL[9:8]	
Access							R	R
Reset							0	0

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

Bits 9:0 – LCOL[9:0] Late Collisions
This register counts the number of late collisions occurring after the slot time (512 bits) has expired. In 10/100 mode, late collisions are counted twice i.e., both as a collision and a late collision.

32.8.60. ETH Deferred Transmission Frames Register

Name: DTF
Offset: 0x1148
Reset: 0x00000000
Property: Read-only

Table 32-75. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
							DEFT[17:16]	
Access							R	R
Reset							0	0

Bit	15	14	13	12	11	10	9	8
	DEFT[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
	DEFT[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 17:0 – DEFT[17:0] Deferred Transmission
This register counts the number of frames experiencing deferral due to carrier sense being active on their first attempt at transmission. Frames involved in any collision are not counted nor are frames that experienced a transmit underrun.

32.8.61. ETH Carrier Sense Errors Register

Name: CSE
Offset: 0x114C
Reset: 0x00000000
Property: Read-only

Table 32-76. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
							CSR[9:8]	
Access							R	R
Reset							0	0

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

Bits 9:0 – CSR[9:0] Carrier Sense Error
This register counts the number of frames transmitted with carrier sense was not seen during transmission or where carrier sense was de-asserted after being asserted in a transmit frame without collision (no underrun). Only incremented in half duplex mode. The only effect of a carrier sense error is to increment this register. The behavior of the other statistics registers is unaffected by the detection of a carrier sense error.

32.8.62. ETH Octets Received Low Register

Name: ORLO
Offset: 0x1150
Reset: 0x00000000
Property: Read-Only(Cleared on read)

When reading the Octets Transmitted and Octets Received Registers, bits [31:0] should be read prior to bits [47:32] to ensure reliable operation.

Table 32-77. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	RXO[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
	RXO[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
	RXO[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
	RXO[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – RXO[31:0] Received Octets
Received octets in frame without errors [31:0]. The number of octets received in valid frames of any type. This counter is 48-bits and is read through two registers. This count does not include octets from pause frames, and is only incremented if the frame is successfully filtered and copied to memory.

32.8.63. ETH Octets Received High Register

Name: ORHI
Offset: 0x1154
Reset: 0x00000000
Property: Read-only(Cleared on Read)

When reading the Octets Transmitted and Octets Received Registers, bits 31:0 should be read prior to bits 47:32 to ensure reliable operation.

Table 32-78. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
	RXO[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
	RXO[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – RXO[15:0] Received Octets

Received octets in frame without errors [47:32]. The number of octets received in valid frames of any type. This counter is 48-bits and is read through two registers. This count does not include octets from pause frames, and is only incremented if the frame is successfully filtered and copied to memory.

32.8.64. ETH Frames Received Register

Name: FR
Offset: 0x1158
Reset: 0x00000000
Property: Read-only(Cleared on Read)

Table 32-79. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	FRX[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
	FRX[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
	FRX[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
	FRX[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – FRX[31:0] Frames Received without Error
This bit field counts the number of frames successfully received, excluding pause frames. It is only incremented if the frame is successfully filtered and copied to memory.

32.8.65. ETH Broadcast Frames Received Register

Name: BCFR
Offset: 0x115C
Reset: 0x00000000
Property: Read-only(Cleared on Read)

Table 32-80. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	BFRX[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
	BFRX[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
	BFRX[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
	BFRX[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – BFRX[31:0] Broadcast Frames Received without Error
Broadcast frames received without error. This bit field counts the number of broadcast frames successfully received. This excludes pause frames, and is only incremented if the frame is successfully filtered and copied to memory.

32.8.66. ETH Multicast Frames Received Register

Name: MFR
Offset: 0x1160
Reset: 0x00000000
Property: Read-only(Cleared on Read)

Table 32-81. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	MFRX[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
	MFRX[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
	MFRX[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
	MFRX[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – MFRX[31:0] Multicast Frames Received without Error
This register counts the number of multicast frames successfully received without error, excluding pause frames, and is only incremented if the frame is successfully filtered and copied to memory.

32.8.67. ETH Pause Frames Received Register

Name: PFR
Offset: 0x1164
Reset: 0x00000000
Property: -

Table 32-82. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
	PFRX[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
	PFRX[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – PFRX[15:0] Pause Frames Received Register
This register counts the number of pause frames received without error.

32.8.68. ETH 64-Byte Frames Received Register

Name: BFR64
Offset: 0x1168
Reset: 0x00000000
Property: Read-only

Table 32-83. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	NFRX[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
	NFRX[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
	NFRX[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
	NFRX[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – NFRX[31:0] 64-Byte Frames Received without Error
This bit field counts the number of 64-byte frames successfully received without error. Excludes pause frames, and is only incremented if the frame is successfully filtered and copied to memory.

32.8.69. ETH 65 to 127 Byte Frames Received Register

Name: TBFR127
Offset: 0x116C
Reset: 0x00000000
Property: -

Table 32-84. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	NFRX[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
	NFRX[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
	NFRX[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
	NFRX[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – NFRX[31:0] 65 to 127 Byte Frames Received without Error
This bit field counts the number of 65 to 127 byte frames successfully received without error. Excludes pause frames, and is only incremented if the frame is successfully filtered and copied to memory.

32.8.70. ETH 128 to 255 Byte Frames Received Register

Name: TBFR255
Offset: 0x1170
Reset: 0x00000000
Property: -

Table 32-85. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	NFRX[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
	NFRX[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
	NFRX[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
	NFRX[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – NFRX[31:0] 128 to 255 Byte Frames Received without Error
This bit field counts the number of 128 to 255 byte frames successfully received without error. Excludes pause frames, and is only incremented if the frame is successfully filtered and copied to memory.

32.8.71. ETH 256 to 511 Byte Frames Received Register

Name: TBFR511
Offset: 0x1174
Reset: 0x00000000
Property: -

Table 32-86. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	NFRX[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
	NFRX[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
	NFRX[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
	NFRX[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – NFRX[31:0] 256 to 511 Byte Frames Received without Error
This bit fields counts the number of 256 to 511 byte frames successfully received without error. Excludes pause frames, and is only incremented if the frame is successfully filtered and copied to memory.

32.8.72. ETH 512 to 1023 Byte Frames Received Register

Name: TBFR1023
Offset: 0x1178
Reset: 0x00000000
Property: -

Table 32-87. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	NFRX[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
	NFRX[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
	NFRX[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
	NFRX[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – NFRX[31:0] 512 to 1023 Byte Frames Received without Error
This bit field counts the number of 512 to 1023 byte frames successfully received without error. Excludes pause frames, and is only incremented if the frame is successfully filtered and copied to memory.

32.8.73. ETH 1024 to 1518 Byte Frames Received Register

Name: TBFR1518
Offset: 0x117C
Reset: 0x00000000
Property: -

Table 32-88. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	NFRX[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
	NFRX[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
	NFRX[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
	NFRX[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – NFRX[31:0] 1024 to 1518 Byte Frames Received without Error
This bit field counts the number of 1024 to 1518 byte frames successfully received without error, i.e., no underrun and not too many retries.

32.8.74. ETH 1519 to Maximum Byte Frames Received Register

Name: TMXBFR
Offset: 0x1180
Reset: 0x00000000
Property: -

Table 32-89. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	NFRX[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
	NFRX[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
	NFRX[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
	NFRX[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – NFRX[31:0] 1519 to Maximum Byte Frames Received without Error
This bit field counts the number of 1519 Byte or above frames successfully received without error. Maximum frame size is determined by the Maximum Frame Size bit (MAXFS, 1536 Bytes) or Jumbo Frame Size bit (JFRAME, 10240 Bytes) in the Network Configuration Register (NCFGR). Excludes pause frames, and is only incremented if the frame is successfully filtered and copied to memory.

32.8.75. ETH Undersized Frames Received Register

Name: UFR
Offset: 0x1184
Reset: 0x00000000
Property: -

Table 32-90. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
							UFRX[9:8]	
Access							R	R
Reset							0	0

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

Bits 9:0 – UFRX[9:0] Undersize Frames Received
This bit field counts the number of frames received less than 64 bytes in length (10/100 mode, full duplex) that do not have either a CRC error or an alignment error.

32.8.76. ETH Oversized Frames Received Register

Name: OFR
Offset: 0x1188
Reset: 0x00000000
Property: -

Table 32-91. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
							OFRX[9:8]	
Access							R	R
Reset							0	0

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

Bits 9:0 – OFRX[9:0] Oversized Frames Received
This pit field counts the number of frames received exceeding 1518 Bytes in length (1536 Bytes if NCFGR.MAXFS is written to '1') but do not have either a CRC error, an alignment error, nor a receive symbol error.

32.8.77. ETH Jabbers Received Register

Name: JR
Offset: 0x118C
Reset: 0x00000000
Property: -

Table 32-92. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
							JRX[9:8]	
Access							R	R
Reset							0	0

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

Bits 9:0 – JRX[9:0] Jabbers Received
This bit field counts the number of frames received exceeding 1518 Bytes in length (1536 Bytes if NCFGR.MAXFS is written to '1') and have either a CRC error, an alignment error or a receive symbol error.

32.8.78. ETH Frame Check Sequence Errors Register

Name: FCSE
Offset: 0x1190
Reset: 0x00000000
Property: Read-only

Table 32-93. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
							FCKR[9:8]	
Access							R	R
Reset							0	0

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

Bits 9:0 – FCKR[9:0] Frame Check Sequence Errors

The register counts frames that are an integral number of bytes, have bad CRC and are between 64 and 1518 bytes in length (1536 Bytes if NCFGR.MAXFS is written to '1'). This register is also incremented if a symbol error is detected and the frame is of valid length and has an integral number of bytes.

This register is incremented for a frame with bad FCS, regardless of whether it is copied to memory due to ignore FCS mode (enabled by writing NCFGR.IRXFCS=1).

32.8.79. ETH Length Field Frame Errors Register

Name: LFFE
Offset: 0x1194
Reset: 0x00000000
Property: -

Table 32-94. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
							LFER[9:8]	
Access							R	R
Reset							0	0

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

Bits 9:0 – LFER[9:0] Length Field Frame Errors

This bit field counts the number of frames received that have a measured length shorter than that extracted from the length field (Bytes 13 and 14). This condition is only counted if the value of the length field is less than 0x0600, the frame is not of excessive length and checking is enabled by writing a '1' to the Length Field Error Frame Discard bit in the Network Configuration Register (NCFGR.LFERD).

32.8.80. ETH Receive Symbol Errors Register

Name: RSE
Offset: 0x1198
Reset: 0x00000000
Property: Read-only

Table 32-95. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
							RXSE[9:8]	
Access							R	R
Reset							0	0

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

Bits 9:0 – RXSE[9:0] Receive Symbol Errors
This bit field counts the number of frames that had RXER asserted during reception. For 10/100 mode symbol errors are counted regardless of frame length checks.

32.8.81. ETH Alignment Errors Register

Name: AE
Offset: 0x119C
Reset: 0x00000000
Property: Read-only

Table 32-96. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
							AER[9:8]	
Access							R	R
Reset							0	0

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

Bits 9:0 – AER[9:0] Alignment Errors
This bit field counts the frames that are not an integral number of bytes long and have bad CRC when their length is truncated to an integral number of Bytes and are between 64 and 1518 Bytes in length (1536 if NCFGR.MAXFS=1). This register is also incremented if a symbol error is detected and the frame is of valid length and does not have an integral number of bytes.

32.8.82. ETH Receive Resource Errors Register

Name: RRE
Offset: 0x11A0
Reset: 0x00000000
Property: -

Table 32-97. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
							RXRER[17:16]	
Access							R	R
Reset							0	0

Bit	15	14	13	12	11	10	9	8
	RXRER[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
	RXRER[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 17:0 – RXRER[17:0] Receive Resource Errors

This bit field counts frames that are not an integral number of bytes long and have bad CRC when their length is truncated to an integral number of Bytes and are between 64 and 1518 Bytes in length (1536 if NCFGR.MAXFS=1). This bit field is also incremented if a symbol error is detected and the frame is of valid length and does not have an integral number of Bytes.

32.8.83. ETH Receive Overruns Register

Name: ROE
Offset: 0x11A4
Reset: 0x00000000
Property: -

Table 32-98. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
							RXOVR[9:8]	
Access							R	R
Reset							0	0

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

Bits 9:0 – RXOVR[9:0] Receive Overruns
This bit field counts the number of frames that are address recognized but were not copied to memory due to a receive overrun.

32.8.84. ETH IP Header Checksum Errors Register

Name: IHCE
Offset: 0x11A8
Reset: 0x00000000
Property: -

Table 32-99. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

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

Bits 7:0 – HCKER[7:0] IP Header Checksum Errors
This register counts the number of frames discarded due to an incorrect IP header checksum, but are between 64 and 1518 Bytes (1536 Bytes if NCFGR.MAXFS=1) and do not have a CRC error, an alignment error, nor a symbol error.

32.8.85. ETH TCP Checksum Errors Register

Name: TCE
Offset: 0x11AC
Reset: 0x00000000
Property: -

Table 32-100. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

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

Bits 7:0 – TCKER[7:0] TCP Checksum Errors
This register counts the number of frames discarded due to an incorrect TCP checksum, but are between 64 and 1518 Bytes (1536 Bytes if NCFGR.MAXFS=1) and do not have a CRC error, an alignment error, nor a symbol error.

32.8.86. ETH UDP Checksum Errors Register

Name: UCE
Offset: 0x11B0
Reset: 0x00000000
Property: -

Table 32-101. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

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

Bits 7:0 – UCKER[7:0] UDP Checksum Errors
This register counts the number of frames discarded due to an incorrect UDP checksum, but are between 64 and 1518 Bytes (1536 Bytes if NCFGR.MAXFS=1) and do not have a CRC error, an alignment error, nor a symbol error.

32.8.87. ETH 1588 Timer Increment Sub-nanoseconds Register

Name: TISUBN
Offset: 0x11BC
Reset: 0x00000000
Property: -

Table 32-102. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
	LSBTIR[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
	LSBTIR[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 – LSBTIR[15:0] Lower Significant Bits of Timer Increment Register
Lower significant bits of Timer Increment Register [15:0], giving a 24-bit timer_increment counter. These bits are the sub-ns value which the 1588 timer will be incremented each clock cycle. Bit n = $2^{(n-16)}$ ns giving a resolution of approximately 15.2E^{-15} sec.

32.8.88. ETH 1588 Timer Seconds High Register

Name: TSH
Offset: 0x11C0
Reset: 0x00000000
Property: -

Table 32-103. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
	TCS[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
	TCS[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 – TCS[15:0] Timer Count in Seconds
This register is writable. It increments by 1 when the IEEE 1588 nanoseconds counter counts to one second. It may also be incremented when the Timer Adjust Register is written.

32.8.89. ETH 1588 Timer Sync Strobe Seconds Low Register

Name: TSSSL
Offset: 0x11C8
Reset: 0x00000000
Property: -

Table 32-104. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	VTS[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
	VTS[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
	VTS[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
	VTS[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 – VTS[31:0] Value of Timer Seconds Register Capture
The lowest significant 32-bit value of the Timer Seconds register captured when both CTRLB.TSUINCand CTRLB.TSUMS are zero.

32.8.90. ETH 1588 Timer Sync Strobe Nanoseconds Register

Name: TSSSN
Offset: 0x11CC
Reset: 0x00000000
Property: -

Table 32-105. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	VTN[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
	VTN[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
	VTN[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
	VTN[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 – VTN[31:0] Value of Timer Nanoseconds Register Capture
The value of the Timer Nanoseconds register captured when both CTRLB.TSUINCand CTRLB.TSUMS are zero.

32.8.91. ETH 1588 Timer Seconds Low Register

Name: TSL
Offset: 0x11D0
Reset: 0x00000000
Property: -

Table 32-106. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	TCS[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
	TCS[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
	TCS[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
	TCS[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 – TCS[31:0] Timer Count in Seconds
This register is writable. It increments by 1 when the IEEE 1588 nanoseconds counter counts to one second. It may also be incremented when the Timer Adjust Register is written.

32.8.92. ETH 1588 Timer Nanoseconds Register

Name: TN
Offset: 0x11D4
Reset: 0x00000000
Property: -

Table 32-107. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
			TNS[29:24]					
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
	TNS[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
	TNS[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
	TNS[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 29:0 – TNS[29:0] Timer Count in Nanoseconds
This register is writable. It can also be adjusted by writes to the IEEE 1588 Timer Adjust Register. It increments by the value of the IEEE 1588 Timer Increment Register each clock cycle.

32.8.93. ETH 1588 Timer Adjust Register

Name: TA
Offset: 0x11D8
Reset: 0x00000000
Property: -

Table 32-108. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	ADJ		ITDT[29:24]					
Access	W		W	W	W	W	W	W
Reset	0		0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	ITDT[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
	ITDT[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
	ITDT[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit 31 – ADJ Adjust 1588 Timer
Write as '1' to subtract from the 1588 timer. Write as '0' to add to it.

Bits 29:0 – ITDT[29:0] Increment/Decrement
The number of nanoseconds to increment or decrement the IEEE 1588 Timer Nanoseconds Register. If necessary, the IEEE 1588 Seconds Register will be incremented or decremented.

32.8.94. ETH IEEE 1588 Timer Increment Register

Name: TI
Offset: 0x11DC
Reset: 0x00000000
Property: -

Table 32-109. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	NIT[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
	ACNS[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
	CNS[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 – NIT[7:0] Number of Increments
The number of increments after which the alternative increment is used.

Bits 15:8 – ACNS[7:0] Alternative Count Nanoseconds
Alternative count of nanoseconds by which the 1588 Timer Nanoseconds Register will be incremented each clock cycle.

Bits 7:0 – CNS[7:0] Count Nanoseconds
A count of nanoseconds by which the IEEE 1588 Timer Nanoseconds Register will be incremented each clock cycle.

32.8.95. ETH PTP Event Frame Transmitted Seconds Low Register

Name: EFTSL
Offset: 0x11E0
Reset: 0x00000000
Property: Read-only

Table 32-110. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	RUD[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
	RUD[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
	RUD[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
	RUD[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – RUD[31:0] Register Update
The register is updated with the value that the IEEE 1588 Timer Seconds Register holds when the SFD of a PTP transmit primary event crosses the MII interface. An interrupt is issued when the register is updated.

32.8.96. ETH PTP Event Frame Transmitted Nanoseconds Register

Name: EFTN
Offset: 0x11E4
Reset: 0x00000000
Property: Read-only

Table 32-111. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
			RUD[29:24]					
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	RUD[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
	RUD[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
	RUD[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 29:0 – RUD[29:0] Register Update
The register is updated with the value that the IEEE 1588 Timer Nanoseconds Register holds when the SFD of a PTP transmit primary event crosses the MII interface. An interrupt is issued when the bit field is updated.

32.8.97. ETH PTP Event Frame Received Seconds Low Register

Name: EFRSL
Offset: 0x11E8
Reset: 0x00000000
Property: Read-only

Table 32-112. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	RUD[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
	RUD[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
	RUD[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
	RUD[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – RUD[31:0] Register Update
The register is updated with the value that the IEEE 1588 Timer Seconds Register holds when the SFD of a PTP receive primary event crosses the MII interface. An interrupt is issued when the register is updated.

32.8.98. ETH PTP Event Frame Received Nanoseconds Register

Name: EFRN
Offset: 0x11EC
Reset: 0x00000000
Property: Read-only

Table 32-113. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
			RUD[29:24]					
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	RUD[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
	RUD[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
	RUD[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 29:0 – RUD[29:0] Register Update

The register is updated with the value that the IEEE 1588 Timer Nanoseconds Register holds when the SFD of a PTP receive primary event crosses the MII interface. An interrupt is issued when the register is updated.

32.8.99. ETH PTP Peer Event Frame Transmitted Seconds Low Register

Name: PEFTSL
Offset: 0x11F0
Reset: 0x00000000
Property: -

Table 32-114. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	RUD[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
	RUD[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
	RUD[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
	RUD[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – RUD[31:0] Register Update
The register is updated with the value that the IEEE 1588 Timer Seconds Register holds when the SFD of a PTP transmit peer event crosses the MII interface. An interrupt is issued when the register is updated.

32.8.100. ETH PTP Peer Event Frame Transmitted Nanoseconds Register

Name: PEFTN
Offset: 0x11F4
Reset: 0x00000000
Property: -

Table 32-115. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
			RUD[29:24]					
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	RUD[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
	RUD[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
	RUD[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 29:0 – RUD[29:0] Register Update
The register is updated with the value that the 1588 Timer Nanoseconds Register holds when the SFD of a PTP transmit peer event crosses the MII interface. An interrupt is issued when the register is updated.

32.8.101. ETH PTP Peer Event Frame Received Seconds Low Register

Name: PEFRSL
Offset: 0x11F8
Reset: 0x00000000
Property: -

Table 32-116. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	RUD[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
	RUD[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
	RUD[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
	RUD[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – RUD[31:0] Register Update

The register is updated with the value that the IEEE 1588 Timer Seconds Register holds when the SFD of a PTP receive primary event crosses the MII interface. An interrupt is issued when the register is updated.

32.8.102. ETH PTP Peer Event Frame Received Nanoseconds Register

Name: PEFRN
Offset: 0x11FC
Reset: 0x00000000
Property: -

Table 32-117. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
			RUD[29:24]					
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	RUD[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
	RUD[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
	RUD[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 29:0 – RUD[29:0] Register Update

The register is updated with the value that the IEEE 1588 Timer Nanoseconds Register holds when the SFD of a PTP receive primary event crosses the MII interface. An interrupt is issued when the register is updated.

32.8.103. Received LPI Transitions

Name: RLPITR
Offset: 0x1270
Reset: 0x00000000
Property: Read-only

Table 32-118. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
	RLPITR[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
	RLPITR[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – RLPITR[15:0] Count of Received LPI Transitions
A count of the number of times there is a transition from receiving normal idle to receiving low power idle.
Cleared on read.

32.8.104. Received LPI Time

Name: RLPITI
Offset: 0x1274
Reset: 0x00000000
Property: Read-only

Table 32-119. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	RLPITI[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
	RLPITI[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
	RLPITI[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 23:0 – RLPITI[23:0] Time in LPI

This field increments once every 16 MCK cycles when the bit RXLPIS (LPI Indication (bit 7)) is set in the NSR.

Cleared on read.

32.8.105. Transmit LPI Transitions

Name: TLPITR
Offset: 0x1278
Reset: 0x00000000
Property: Read-only

Table 32-120. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	TLPITR[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
	TLPITR[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
	TLPITR[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 23:0 – TLPITR[23:0] Count of LIP Transitions

A count of the number of times the bit TXLPIEN (Enable LPI Transmission (bit 19)) goes from low to high in the NCR.

Cleared on read.

32.8.106. Transmit LPI Time

Name: TLPITI
Offset: 0x127C
Reset: 0x00000000
Property: Read-only

Table 32-121. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	TLPITI[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
	TLPITI[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
	TLPITI[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 23:0 – TLPITI[23:0] Time in LPI

This field increments once every 16 MCK cycles when the bit TXLPIEN (Enable LPI Transmission (bit 19)) is set in NCR.

Cleared on read.

32.8.107. ETH Interrupt Status Register Priority Queue x

Name: ISRQ
Offset: 0x1400 + (n-1)*0x04 [n=1..5]
Reset: 0x00000000
Property: Read/Write

Table 32-122. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
					HRESP	ROVR		
Access					R/W	R/W		
Reset					0	0		

Bit	7	6	5	4	3	2	1	0
	TCOMP	TFC	RLEX	TUR	TXUBR	RXUBR	RCOMP	
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	0	0	0	0	0	0	0	

Bit 11 – HRESP HRESP Not OK

Bit 10 – ROVR Receive Overrun

Bit 7 – TCOMP Transmit Complete

Bit 6 – TFC Transmit Frame Corruption Due to AXI Error
Transmit frame corruption due to AXI error—set if an error occurs whilst midway through reading transmit frame from the AXI, including HRESP errors and buffers exhausted mid frame.

Bit 5 – RLEX Retry Limit Exceeded or Late Collision

Bit 4 – TUR Transmit Underrun

Bit 3 – TXUBR TX Used Bit Read

Bit 2 – RXUBR RX Used Bit Read

Bit 1 – RCOMP Receive Complete

32.8.108. ETH Transmit Buffer Queue Base Address Register Priority Queue x

Name: TBPQB
Offset: 0x1440 + (n-1)*0x04 [n=1..5]
Reset: 0x00000000
Property: Read/Write

These registers hold the start address of the transmit buffer queues (transmit buffers descriptor lists) for the additional queues and must be initialized to the address of valid descriptors, even if the priority queues are not used.

Table 32-123. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	TXBQBA[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
	TXBQBA[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
	TXBQBA[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
	TXBQBA[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 – TXBQBA[29:0] Transmit Buffer Queue Base Address
Contains the address of the start of the transmit queue.

32.8.109. ETH Receive Buffer Queue Base Address Register Priority Queue x

Name: RBPQB
Offset: 0x1480 + (n-1)*0x04 [n=1..5]
Reset: 0x00000000
Property: Read/Write

These registers hold the start address of the receive buffer queues (receive buffers descriptor lists) for the additional queues used when priority queues are employed.

Table 32-124. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	RXBQBA[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
	RXBQBA[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
	RXBQBA[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
	RXBQBA[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 – RXBQBA[29:0] Receive Buffer Queue Base Address
Holds the address of the start of the receive queue.

32.8.110. Receive Buffer Size Register Priority Queue x

Name: RBQSZ
Offset: 0x14A0 + (n-1)*0x04 [n=1..5]
Reset: 0x00000002
Property: Read/Write

Table 32-125. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
	RBS[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
	RBS[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	1	0

Bits 15:0 – RBS[15:0] Receive Buffer Size

DMA receive buffer size in AXI system memory. The value defined by these bits determines the size of buffer to use in main AXI system memory when writing received data.
The value is defined in multiples of 64 Bytes such that a value of 0x01 corresponds to buffers of 64 Bytes, 0x02 corresponds to 128 Bytes etc.
Examples:

- 0x18: 1536 Bytes (1 × max length frame/buffer)
- 0xA0: 10240 Bytes (1 × 10K jumbo frame/buffer)

Note: This value should never be written as zero.

32.8.111. Screening Type 1 Register x Priority Queue

Name: SCRT1
Offset: 0x1500 + n*0x04 [n=0..3]
Reset: 0x00000000
Property: Read/Write

Screening type 1 registers are used to allocate up to 6 priority queues to received frames based on certain IP or UDP fields of incoming frames.

Table 32-126. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
			UDPE	DSTCE	UDPP[15:12]			
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
	UDPP[11:4]							
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
	UDPP[3:0]				DSTCM[7:4]			
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	7	6	5	4	3	2	1	0
	DSTCM[3:0]					QNMBR[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 29 – UDPE UDP Port Match Enable
When this bit is written to '1', the UDP Destination Port of the received UDP frame is matched against the value stored in the bit field UDPP.

Bit 28 – DSTCE Differentiated Services or Traffic Class Match Enable
When this bit is written to '1', the DS (differentiated services) field of the received IPv4 header or TC field (traffic class) of IPv6 headers are matched against the value stored in bit field DSTCM.

Bits 27:12 – UDPP[15:0] UDP Port Match
When UDP port match enable is set (UDPME=1), the UDP Destination Port of the received UDP frame is matched against this bit field.

Bits 11:4 – DSTCM[7:0] Differentiated Services or Traffic Class Match
When DS/TC match enable is set (DSTCE), the DS (differentiated services) field of the received IPv4 header or TC field (traffic class) of IPv6 headers are matched against this bit field.

Bits 2:0 – QNMBR[2:0] Queue Number
If a match is successful, then the queue value programmed in this bit field is allocated to the frame.

32.8.112. ETH Screening Type 2 Register x Priority Queue

Name: SCRT2
Offset: 0x1540 + n*0x04 [n=0..7]
Reset: 0x00000000
Property: Read/Write

Screening type 2 registers are used to allocate up to 6 priority queues to received frames based on the VLAN priority field of received Ethernet frames.

Table 32-127. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
		COMPCE	COMPC[4:0]					COMPBE
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
	COMPB[4:0]					COMPAE	COMPA[4:3]	
Access	R/W	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
	COMPA[2:0]			ETHE	I2ETH[2:0]			VLANE
Access	R/W	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
		VLANP[2:0]				QNB[2:0]		
Access		R/W	R/W	R/W		R/W	R/W	R/W
Reset		0	0	0		0	0	0

Bit 30 – COMPCE Compare C Enable

Value	Description
0	Compare C is disabled.
1	Comparison via the register designated by index COMPC is enabled.

Bits 29:25 – COMPC[4:0] Index of Screening Type 2 Compare Word 0/Word 1 register x
COMPC is a pointer to the compare registers SCRT2CMP0 and SCRT2CMP1. When COMPCE=1, the compare is true if the data at the frame offset ANDed with the value MASKVAL is equal to the value of COMPVAL ANDed with the value of MASKVAL.

Bit 24 – COMPBE Compare B Enable

Value	Description
0	Compare B is disabled.
1	Comparison via the register designated by index COMPB is enabled.

Bits 23:19 – COMPB[4:0] Index of Screening Type 2 Compare Word 0/Word 1 register x
COMPB is a pointer to the compare registers SCRT2CMP0 and SCRT2CMP1. When COMPBE=1, the compare is true if the data at the frame offset ANDed with the value MASKVAL is equal to the value of COMPVAL ANDed with the value of MASKVAL.

Bit 18 – COMPAE Compare A Enable

Value	Description
0	Compare A is disabled.
1	Comparison via the register designated by index COMPA is enabled.

Bits 17:13 – COMPA[4:0] Index of Screening Type 2 Compare Word 0/Word 1 register x
COMPA is a pointer to the compare registers SCRT2CMP0 and SCRT2CMP1. When COMPAE=1, the compare is true if the data at the frame offset ANDed with the value MASKVAL is equal to the value of COMPVAL ANDed with the value of MASKVAL.

Bit 12 – ETHE EtherType Enable

Value	Description
0	EtherType match is disabled
1	EtherType match with bits [15:0] of the register designated by the value in I2ETH is enabled

Bits 11:9 – I2ETH[2:0] Index of Screening Type 2 EtherType register x
When EtherType is enabled (ETHE=1), the EtherType field (last EtherType in the header if the frame is VLAN-tagged) is compared with bits [15:0] in the register designated by the value of this bit field.

Bit 8 – VLANE VLAN Enable

Value	Description
0	VLAN match disabled
1	VLAN match is enabled

Bits 6:4 – VLANP[2:0] VLAN Priority
When VLAN match is enabled (VLANE=1), the VLAN Priority field of the received frame is matched against the value of this bit field.

Bits 2:0 – QNB[2:0] Queue Number
If a match is successful, then the queue value programmed in QNB is allocated to the frame.

32.8.113. ETH Interrupt Enable Register Priority Queue x

Name: IERQ
Offset: 0x1600 + (n-1)*0x04 [n=1..5]
Reset: –
Property: Write-only

The following values are valid for all listed bit names of this register:

0: No effect.

1: Enables the corresponding interrupt.

Table 32-128. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
					HRESP	ROVR		
Access					W	W		
Reset					–	–		

Bit	7	6	5	4	3	2	1	0
	TCOMP	TFC	RLEX	TUR	TXUBR	RXUBR	RCOMP	
Access	W	W	W	W	W	W	W	
Reset	–	–	–	–	–	–	–	

Bit 11 – HRESP HRESP Not OK

Bit 10 – ROVR Receive Overrun

Bit 7 – TCOMP Transmit Complete

Bit 6 – TFC Transmit Frame Corruption Due to AXI Error

Bit 5 – RLEX Retry Limit Exceeded or Late Collision

Bit 4 – TUR Transmit Underrun

Bit 3 – TXUBR TX Used Bit Read

Bit 2 – RXUBR RX Used Bit Read

Bit 1 – RCOMP Receive Complete

32.8.114. ETH Interrupt Disable Register Priority Queue x

Name: IDRQ
Offset: 0x1620 + (n-1)*0x04 [n=1..5]
Reset: –
Property: Write-only

The following values are valid for all listed bit names of this register:

0: No effect.

1: Disables the corresponding interrupt.

Table 32-129. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
					HRESP	ROVR		
Access					W	W		
Reset					–	–		

Bit	7	6	5	4	3	2	1	0
	TCOMP	TFC	RLEX	TUR	TXUBR	RXUBR	RCOMP	
Access	W	W	W	W	W	W	W	
Reset	–	–	–	–	–	–	–	

Bit 11 – HRESP HRESP Not OK

Bit 10 – ROVR Receive Overrun

Bit 7 – TCOMP Transmit Complete

Bit 6 – TFC Transmit Frame Corruption Due to AXI Error

Bit 5 – RLEX Retry Limit Exceeded or Late Collision

Bit 4 – TUR Transmit Underrun

Bit 3 – TXUBR TX Used Bit Read

Bit 2 – RXUBR RX Used Bit Read

Bit 1 – RCOMP Receive Complete

32.8.115. ETH Interrupt Mask Register Priority Queue x

Name: IMRQ
Offset: 0x1640 + (n-1)*0x04 [n=1..5]
Reset: 0x00000876
Property: Read/Write

A read of this register returns the value of the receive complete interrupt mask.

A write to this register directly affects the state of the corresponding bit in the Interrupt Status Register, causing an interrupt to be generated if a '1' is written.

The following values are valid for all listed bit names of this register:

0: Corresponding interrupt is enabled.

1: Corresponding interrupt is disabled.

Table 32-130. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
					HRESP	ROVR		
Access					R/W	R/W		
Reset					1	1		

Bit	7	6	5	4	3	2	1	0
	TCOMP	AXI	RLEX	TUR	TXUBR	RXUBR	RCOMP	
Access	R/W	R/W	R/W	W	W	R/W	R/W	
Reset	1	1	1	1	1	1	1	

Bit 11 – HRESP HRESP Not OK

Bit 10 – ROVR Receive Overrun

Bit 7 – TCOMP Transmit Complete

Bit 6 – AXI AXI Error

Bit 5 – RLEX Retry Limit Exceeded or Late Collision

Bit 4 – TUR Transmit Underrun

Bit 3 – TXUBR TX Used Bit Read

Bit 2 – RXUBR RX Used Bit Read

Bit 1 – RCOMP Receive Complete

32.8.116. ETH Screening Type 2 EtherType Register x

Name: SCRT2ET
Offset: 0x16E0 + n*0x04 [n=0..3]
Reset: 0x00000000
Property: Read/Write

Table 32-131. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
	COMPVAL[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
	COMPVAL[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 – COMPVAL[15:0] EtherType Compare Value

When the bit SCRT2.ETHE is written to '1', the EtherType (last EtherType in the header if the frame is VLAN tagged) is compared with bits [15:0] in the register designated by SCRT2.I2ETH.

32.8.117. ETH Screening Type 2 Compare Word 0 Register x

Name: SCRT2CMP0
Offset: 0x1700 + n*0x08 [n=0..23]
Reset: 0x00000000
Property: Read/Write

Table 32-132. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	COMPVAL[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
	COMPVAL[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
	MASK[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
	MASK[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 – COMPVAL[15:0] Compare Value

The byte stored in bits [23:16] is compared against the first byte of the 2 bytes extracted from the frame.
The byte stored in bits [31:24] is compared against the second byte of the 2 bytes extracted from the frame.

Bits 15:0 – MASK[15:0] Mask Value

The value of MASK ANDed with the 2 bytes extracted from the frame is compared to the value of MASK ANDed with the value of COMPVAL.

32.8.118. ETH Screening Type 2 Compare Word 1 Register x

Name: SCRT2CMP1
Offset: 0x1704 + n*0x08 [n=0..23]
Reset: 0x00000000
Property: Read/Write

Table 32-133. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
								OFFSSTRT[1]
Access								R/W
Reset								0

Bit	7	6	5	4	3	2	1	0
	OFFSSTRT[0]	OFFSVAL[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

Bits 8:7 – OFFSSTRT[1:0] Ethernet Frame Offset Start

Value	Name	Description
0	FRAMESTART	Offset from the start of the frame
1	ETHERTYPE	Offset from the byte after the EtherType field
2	IP	Offset from the byte after the IP header field
3	TCP_UDP	Offset from the byte after the TCP/UDP header field

Bits 6:0 – OFFSVAL[6:0] Offset Value in Bytes

The value of OFFSVAL ranges from 0 to 127 bytes, and is counted from either the start of the frame, the byte after the EtherType field (last EtherType in the header if the frame is VLAN tagged), the byte after the IP header (IPv4 or IPv6) or the byte after the TCP/UDP header.

33. Event System (EVSYS)

33.1. Overview

The Event System (EVSYS) allows autonomous, low-latency and configurable communication between peripherals. Several peripherals can be configured to generate and 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 many event generators and can have multiple event users.

Communication is made without CPU intervention and without consuming system resources, such as bus or RAM bandwidth. This reduces the load on the CPU and other system resources, compared to a traditional interrupt-based system.

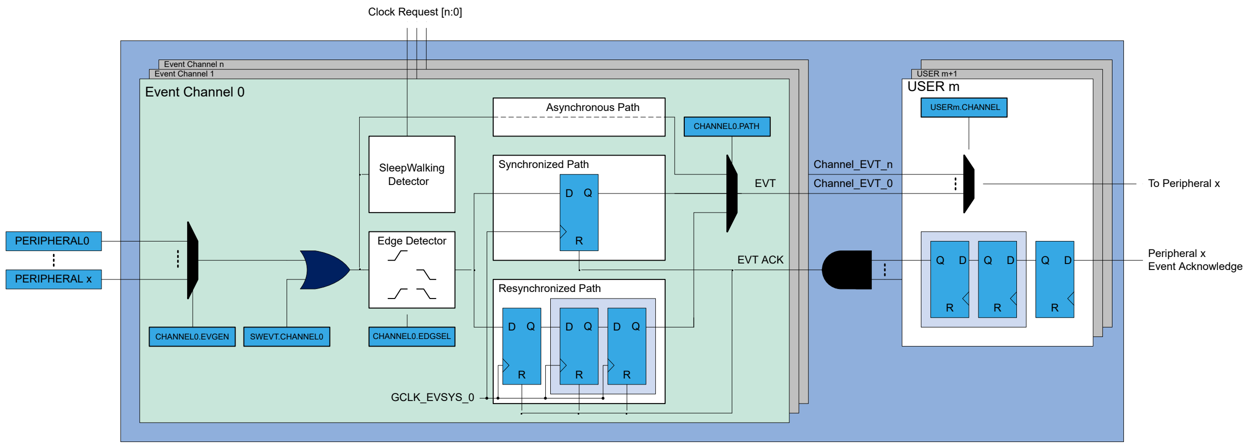
33.2. Features

The following are key features of the EVSYS module:

- Up to 32 configurable event channels:
  - All channels can be connected to any event generator
  - All channels provide a pure asynchronous path
  - Configurable number of channels which can provide a resynchronized path using their dedicated generic clock (GCLK_EVSYS_CHANNEL_n)
- Device-specific event generators
- Device-specific 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

33.3. Block Diagram

Figure 33-1. Event System Block Diagram



33.4. Peripheral Dependencies

Peripheral Name	Base Address	NVIC IRQ Index: Source	MCLK AHBx/APBx Clock Enable Mask Bit	GCLK Peripheral Channel Clock Name:Register	PAC Peripheral Identifier (PAC.WRCTRL.PERIDx)	Power Domain
EVSYS	0x4480_E000	39 : EVD_0, OVR_0 40 : EVD_1, OVR_1 41 : EVD_2, OVR_2 42 : EVD_3, OVR_3 43 : EVD_4, OVR_4, EVD_5, OVR_5, EVD_6, OVR_6, EVD_7, OVR_7, EVD_8, OVR_8, EVD_9, OVR_9, EVD_10, OVR_10, EVD_11, OVR_11, NSCHK	MCLK.CLKMSK2[6]	GCLK_EVSYS_CH0 : GCLK.PCHCTRL[6] GCLK_EVSYS_CH1 : GCLK.PCHCTRL[7] GCLK_EVSYS_CH2 : GCLK.PCHCTRL[8] GCLK_EVSYS_CH3 : GCLK.PCHCTRL[9] GCLK_EVSYS_CH4 : GCLK.PCHCTRL[10] GCLK_EVSYS_CH5 : GCLK.PCHCTRL[11] GCLK_EVSYS_CH6 : GCLK.PCHCTRL[12] GCLK_EVSYS_CH7 : GCLK.PCHCTRL[13] GCLK_EVSYS_CH8 : GCLK.PCHCTRL[14] GCLK_EVSYS_CH9 : GCLK.PCHCTRL[15] GCLK_EVSYS_CH10 : GCLK.PCHCTRL[16] GCLK_EVSYS_CH11 : GCLK.PCHCTRL[17]	27	VDDCORE_SW

33.5. Power Management

The EVSYS can be used to wake up the CPU from a sleep mode (except BACKUP and OFF Mode), even if the clock used by the EVSYS channel and the EVSYS bus clock are disabled. Refer to the *PM –Power Manager* for details on the different sleep modes.

Although the clock for the EVSYS is stopped, the device still can wake up the EVSYS clock. Some event generators can generate an event when their clocks are stopped. The generic clock for the channel (GCLK_EVSYS_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 asynchronous or resynchronized.

33.6. Clocks

The EVSYS bus clock (CLK_EVSYS_APB) can be enabled and disabled in the Main Clock module (MCLK). The EVSYS APB BUS interface clocks, CLK_EVSYS_APB, are enabled by default on reset. (See MCLK).

Each EVSYS channel which can be configured as 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.

33.7. Functional Description

33.7.1. 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 or resynchronized mode of operation. The mode of operation must be selected based on the requirements of the application.

When using a resynchronized path, the Event System includes options to transfer events to users when rising, falling or both edges are detected on event generators.

For further details, refer to the Channel Path section of this chapter.

33.7.2. Basic Operation

33.7.2.1. 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 have to be configured. The recommended sequence is:

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 (i.e., TCC.EVCTRL.MCEO1, AC.EVCTRL.WINEO0, RTC.EVCTRL.OVFEO).
2. Configure the EVSYS:
   a. Configure the Event User multiplexer by writing the respective EVSYS.USERm register, see User Multiplexer Setup.
   b. Configure the Event Channel by writing the respective EVSYS.CHANNELn register, see 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 (i.e., TC.EVCTRL.EVACT, PDEC.EVCTRL.EVACT). Note: not all peripherals require this step.
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 (i.e., AC.EVCTRL.IVEI0, ADC.EVCTRL.STARTEI).

33.7.2.2. 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.

33.7.2.3. 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.

33.7.2.4. 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).

33.7.2.5. Channel Path

There are different ways to propagate the event from an event generator:

- Asynchronous 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.

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.

Note: If the re-sync path is required, keep the EVSYS and Event User running during standby mode.

33.7.2.6. Edge Detection

When 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).

33.7.2.7. 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.
- 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.

33.7.2.8. 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 resynchronized paths. In the case of asynchronous path, the CHINTFLAGn.OVR is always read as zero.

33.7.2.9. 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 resynchronized path. In the case of an asynchronous path, the CHINTFLAGn.EVD is always zero.

33.7.2.10. Channel Status

The Channel Status register (CHSTATUS) shows the status of the channels when using a 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.

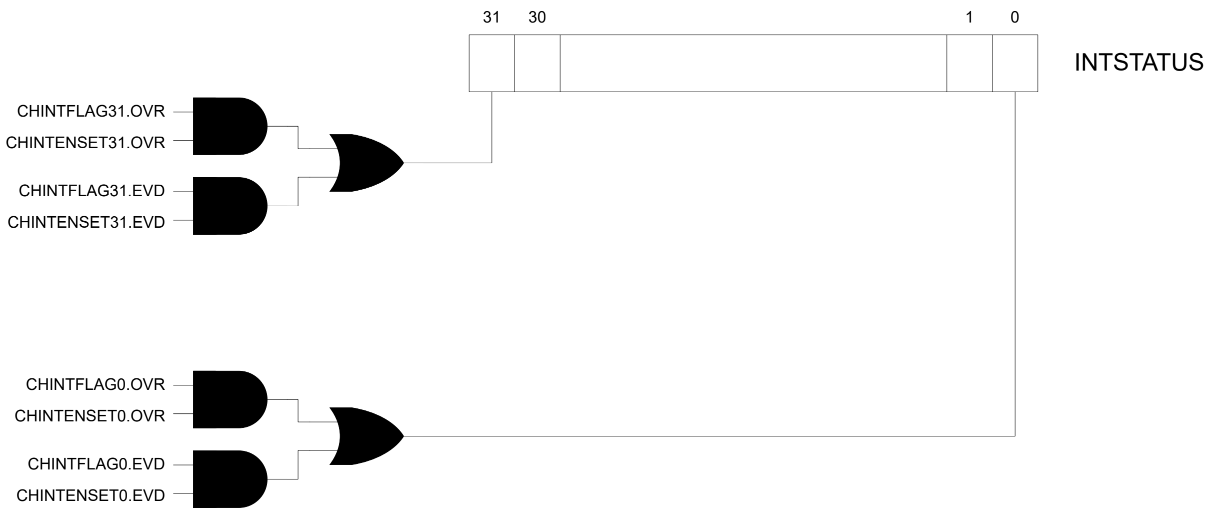
33.7.2.11. Software Event

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.

33.7.2.12. Interrupt Status and Interrupts Arbitration

The Interrupt Status register stores all channels with pending interrupts, as shown below.

Figure 33-2. 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 33-3. 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 33-4. 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.

33.7.3. 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.

33.7.4. 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 33-1. Event Channel Sleep Behavior

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	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.

33.8. Register Summary

For descriptions and definitions of both Register and bitfield properties, refer to Register Properties.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRLA	7:0								SWRST
0x01 ... 0x03	Reserved									
0x04	SWEVT	31:24	CHANNEL31	CHANNEL30	CHANNEL29	CHANNEL28	CHANNEL27	CHANNEL26	CHANNEL25	CHANNEL24
		23:16	CHANNEL23	CHANNEL22	CHANNEL21	CHANNEL20	CHANNEL19	CHANNEL18	CHANNEL17	CHANNEL16
		15:8	CHANNEL15	CHANNEL14	CHANNEL13	CHANNEL12	CHANNEL11	CHANNEL10	CHANNEL9	CHANNEL8
		7:0	CHANNEL7	CHANNEL6	CHANNEL5	CHANNEL4	CHANNEL3	CHANNEL2	CHANNEL1	CHANNEL0
0x08	PRICTRL	7:0	RREN			PRI[4:0]				
0x09 ... 0x0F	Reserved									
0x10	INTPEND	15:8	BUSY	READY					EVD	OVR
		7:0				ID[4:0]				
0x12 ... 0x13	Reserved									
0x14	INTSTATUS	31:24								
		23:16								
		15:8					CHINT11	CHINT10	CHINT9	CHINT8
		7:0	CHINT7	CHINT6	CHINT5	CHINT4	CHINT3	CHINT2	CHINT1	CHINT0
0x18	BUSYCH	31:24								
		23:16								
		15:8					BUSYCH11	BUSYCH10	BUSYCH9	BUSYCH8
		7:0	BUSYCH7	BUSYCH6	BUSYCH5	BUSYCH4	BUSYCH3	BUSYCH2	BUSYCH1	BUSYCH0
0x1C	READYUSR	31:24								
		23:16								
		15:8					READYUSR11	READYUSR10	READYUSR9	READYUSR8
		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[7:0]							
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[7:0]							
0x2C	CHINTENCLR1	7:0							EVD	OVR
0x2D	CHINTENSET1	7:0							EVD	OVR
0x2E	CHINTFLAG1	7:0							EVD	OVR
0x2F	CHSTATUS1	7:0							BUSYCH	RDYUSR
0x30	CHANNEL2	31:24								
		23:16								
		15:8	ONDEMAND	RUNSTDBY			EDGSEL[1:0]		PATH[1:0]	
		7:0	EVGEN[7:0]							
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[7:0]							
0x3C	CHINTENCLR3	7:0							EVD	OVR

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
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[7:0]							
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[7:0]							
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[7:0]							
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[7:0]							
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 ... 0x011F	Reserved									
0x0120	USER0	31:24								
		23:16								
		15:8								
		7:0	CHANNEL[7:0]							
...										
0x019C	USER31	31:24								
		23:16								
		15:8								
		7:0	CHANNEL[7:0]							

33.8.1. Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00
Property: PAC Write-Protection

Table 33-2. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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

Bit 0 – SWRST Software Reset

Writing '0' to this bit has no effect.
Writing '1' to this bit resets all registers in the EVSYS to their initial state. It will always take precedence, meaning that all other writes in the same write-operation will be discarded.
Note: Before applying a Software Reset it is recommended to disable the event generators.

33.8.2. Software Event

Name: SWEVT
Offset: 0x04
Reset: 0x00000000
Property: PAC Write-Protection

Table 33-3. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	CHANNEL31	CHANNEL30	CHANNEL29	CHANNEL28	CHANNEL27	CHANNEL26	CHANNEL25	CHANNEL24
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	CHANNEL23	CHANNEL22	CHANNEL21	CHANNEL20	CHANNEL19	CHANNEL18	CHANNEL17	CHANNEL16
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	CHANNEL15	CHANNEL14	CHANNEL13	CHANNEL12	CHANNEL11	CHANNEL10	CHANNEL9	CHANNEL8
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CHANNEL7	CHANNEL6	CHANNEL5	CHANNEL4	CHANNEL3	CHANNEL2	CHANNEL1	CHANNEL0
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – CHANNELx Channel x Software Selection [x=0..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.

33.8.3. Priority Control

Name: PRICTRL
Offset: 0x08
Reset: 0x00
Property: PAC Write-Protection

Table 33-4. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
	RREN			PRI[4:0]				
Access	R/W			R/W	R/W	R/W	R/W	R/W
Reset	0			0	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 4:0 – PRI[4:0] 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).

33.8.4. Channel Pending Interrupt

Name: INTPEND
Offset: 0x10
Reset: 0x4000
Property: -

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.

Table 33-5. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	15	14	13	12	11	10	9	8
	BUSY	READY					EVD	OVR
Access	R	R					R/W	R/W
Reset	0	1					0	0

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

Bit 15 – 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 4:0 – ID[4:0] 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.

33.8.5. Interrupt Status

Name: INTSTATUS
Offset: 0x14
Reset: 0x00000000

Table 33-6. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
					CHINT11	CHINT10	CHINT9	CHINT8
Access					R	R	R	R
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CHINT7	CHINT6	CHINT5	CHINT4	CHINT3	CHINT2	CHINT1	CHINT0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11 – CHINTx Channel x Pending Interrupt

This bit is set when Channel x has a pending interrupt.

This bit is cleared when the corresponding Channel x interrupts are disabled, or the source interrupt sources are cleared.

33.8.6. Busy Channels

Name: BUSYCH
Offset: 0x18
Reset: 0x00000000
Property: -

Table 33-7. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
					BUSYCH11	BUSYCH10	BUSYCH9	BUSYCH8
Access					R	R	R	R
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	BUSYCH7	BUSYCH6	BUSYCH5	BUSYCH4	BUSYCH3	BUSYCH2	BUSYCH1	BUSYCH0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11 – 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'.

33.8.7. Ready Users

Name: READYUSR
Offset: 0x1C
Reset: 0xFFFFFFFF
Property: -

Table 33-8. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
					READYUSR11	READYUSR10	READYUSR9	READYUSR8
Access					R	R	R	R
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	READYUSR7	READYUSR6	READYUSR5	READYUSR4	READYUSR3	READYUSR2	READYUSR1	READYUSR0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11 – READYUSR 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.

33.8.8. Channel n Control

Name: CHANNELn
Offset: 0x20 + n*0x08 [n=0..7]
Reset: 0x00008000
Property: PAC Write-Protection

This register allows the user to configure channel n. To write to this register, do a single, 32-bit write of all the configuration data.

Table 33-9. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
	ONDEMAND	RUNSTDBY			EDGSEL[1:0]		PATH[1:0]	
Access	R/W	R/W			R/W	R/W	R/W	R/W
Reset	1	0			0	0	0	0

Bit	7	6	5	4	3	2	1	0
	EVGEN[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 – ONDEMAND Generic Clock On Demand

Value	Description
0	Generic clock for a channel is always on, if the channel is configured and generic clock source is enabled.
1	Generic clock is requested on demand while an event is handled

Bit 14 – RUNSTDBY Run in Standby
This bit is used to define the behavior during standby sleep mode.

Value	Description
0	The channel is disabled in standby sleep mode.
1	The channel is not stopped in standby sleep mode and depends on the CHANNEL.ONDEMAND bit.

Bits 11:10 – EDGSEL[1:0] Edge Detection Selection
These bits set the type of edge detection to be used on the channel.
These bits must be written to zero when using the asynchronous path.

Value	Name	Description
0x0	NO_EVT_OUTPUT	No event output when using the resynchronized path

Value	Name	Description
`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. Only a channel with an index less than 12, embeds resynchronous capabilities. The rest of available channels support only asynchronous path selection.

Important: When the resynchronized path is enabled, event inversion feature in peripherals must not be enabled (EVCTRL.xxxINV = 0.

Important: To avoid spurious EVSYS detections, EVSYS must be write protected by configuring the WRCTRL register in the PAC before being used.

Value	Name	Description
`0x0`	ASYNCHRONOUS	Asynchronous path
`0x1`	RESYNCHRONIZED	Resynchronized path
`0x2`	-	Reserved
`Other`	-	Reserved

Bits 7:0 – EVGEN[7:0] Event Generator Selection

These bits are used to choose the event generator to connect to the selected channel.

Table 33-10. Event Generator (EVGEN) Mapping

Module Name	Name of Generator	Value	Description
SUPC	SUPC LVDET	1	Low Voltage Detection
OSCCTRL	XOSC FAIL	2	XOSC fail detection
OSC32KCTRL	XOSC32K_FAIL	3	XOSC32K fail detection
FREQM	DONE	4	Measurement Done
	WINMON	5	Window Monitor Condition Met
RTC	RTC-PERx	6-13	RTC period x=0..7
	PERD	14	RTC Daily Period
	RTC-CMPx	15-18	RTC comparison x=0..3
	RTC-TAMPER	19	RTC tamper detection
	RTC-OVF	20	RTC overflow
EIC	EXTINTx	21-36	EIC external interrupt x=0..15
PAC	PAC_ACCERR	37	PAC Access Error
DMA0	DMA0_CHx	38-45	DMA channel x=0...7
DMA1	DMA1_CHx	46-49	DMA channel x=0...3
TCC0	OVF	50	TCC0 Overflow
	TRG	51	TCC0 Trigger Event
	CNT	52	TCC0 Counter
	MCx	53-58	TCC0 Match/Compare x=0..5
TCC1	OVF	59	TCC1 Overflow
	TRG	60	TCC1 Trigger Event
	CNT	61	TCC1 Counter
	MCx	62-67	TCC1 Match/Compare x=0..5

Table 33-10. Event Generator (EVGEN) Mapping (continued)

Module Name	Name of Generator	Value	Description
TCC2	OVF	68	TCC2 Overflow
	TRG	69	TCC2 Trigger Event
	CNT	70	TCC2 Counter
	MCx	71-76	TCC2 Match/Compare x=0..5
TCC3	OVF	77	TCC3 Overflow
	TRG	78	TCC3 Trigger Event
	CNT	79	TCC3 Counter
	MCx	80-85	TCC3 Match/Compare x=0..5
TCC4	OVF	86	TCC4 Overflow
	TRG	87	TCC4 Trigger Event
	CNT	88	TCC4 Counter
	MCx	89,90	TCC4 Match/Compare x=0..1
TCC5	OVF	91	TCC5 Overflow
	TRG	92	TCC5 Trigger Event
	CNT	93	TCC5 Counter
	MCx	94,95	TCC5 Match/Compare x=0..1
TCC6	OVF	96	TCC6 Overflow
	TRG	97	TCC6 Trigger Event
	CNT	98	TCC6 Counter
	MCx	99,100	TCC6 Match/Compare x=0..1
TCC7	OVF	101	TCC7 Overflow
	TRG	102	TCC7 Trigger Event
	CNT	103	TCC7 Counter
	MCx	104,105	TCC7 Match/Compare x=0..1
ADC	ADC RESRDY	106	ADC Ready
	ADC CMP	107	ADC Compare Event
AC	AC COMPx	108,109	AC Comparator x=0..1
	AC WIN	110	AC0 Window
PTC	EOC	111	PTC end of Conversion
	WCOMP	112	PTC Window Compare
SPI_IXS	GEN	113	Frame Pulse (??)
CCL	LUTOUT_x	114-117	LUTx Output, x=0...3
PDEC	DIR	118	Direction Output
	ERR	119	Error Output
	MCX_0	121	Match Channel 0
	MCX_1	122	Match Channel 1
	VLC	123	Velocity Output
ETH	TSU_CMP	124	Time Stamp Unit (TSU) Compare
TRNG	READY	125	TRNG ready

33.8.9. Channel n Interrupt Enable Clear

Name: CHINTENCLRn
Offset: 0x24 + n*0x08 [n=0..7]
Reset: 0x00
Property: PAC Write-Protection

Table 33-11. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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

Bit 1 – EVD Channel Event Detected Interrupt Disable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Event Detected Channel Interrupt Enable bit, which disables the Event Detected Channel interrupt.

Value	Description
0	The Event Detected Channel interrupt is disabled.
1	The Event Detected Channel interrupt is enabled.

Bit 0 – OVR Channel Overrun Interrupt Disable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Overrun Channel Interrupt Enable bit, which disables the Overrun Channel interrupt.

Value	Description
0	The Overrun Channel interrupt is disabled.
1	The Overrun Channel interrupt is enabled.

33.8.10. Channel n Interrupt Enable Set

Name: CHINTENSETn
Offset: 0x25 + n*0x08 [n=0..7]
Reset: 0x00
Property: PAC Write-Protection

Table 33-12. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
							EVD	OVR
Access							R/W	R/W
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.

33.8.11. Channel n Interrupt Flag Status and Clear

Name: CHINTFLAGn
Offset: 0x26 + n*0x08 [n=0..7]
Reset: 0x00

Note: Interrupt flags must be cleared and then read back to confirm the clear before exiting the ISR to avoid double interrupts.

Table 33-13. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
							EVD	OVR
Access							R/W	R/W
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.

33.8.12. Channel n Status

Name: CHSTATUSn
Offset: 0x27 + n*0x08 [n=0..7]
Reset: 0x01

Table 33-14. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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

Bit 1 – BUSYCH Busy Channel
This bit is cleared when 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 ‘0’.

33.8.13. Event User m

Name: USERm
Offset: 0x0120 + m*0x04 [m=0..31]
Reset: 0x00000000
Property: PAC Write-Protection

Table 33-15. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	CHANNEL[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 – CHANNEL[7:0] Channel Event Selection

These bits select channel n to connect to the event user m. The following table lists all of the Event Users and the associated 'm' value to determine which USERm register to define the desired Event Channel.

Notes:

1. A value x of this bit field selects channel n = x-1.
2. By default, a channel is asynchronous. The Channel resynchronous path can be enabled if its index is < 12 and resynchronous selection is written to CHANNEL.PATH bit field.

Table 33-16. Event User Mapping

User Macro	User Multiplexor	USER INDEX	Description	Path Type [1]
FREQM	START	0	Start Measurement	AR
RTC	TAMPER	1	RTC Tamper	A
PORT	EVx	2-5	PORT Event x=0..3	AR
DMA0	CHx-Start	6-13	Channel Start x=0..7	AR
DMA0	CHx-Aux	14-21	Channel Aux x=0..7	AR
DMA1	CHx-Start	22-25	Channel Start x=0..3	AR
DMA1	CHx-Aux	26-29	Channel Aux x=0..3	AR
TCC0	EVx	30,31	EV x=0..1	AR
	MCx	32-37	MC x=0..5	AR

Table 33-16. Event User Mapping (continued)

User Macro	User Multiplexor	USER INDEX	Description	Path Type (1)
TCC1	EVx	38,39	EV x=0..1	AR
	MCx	40-45	MC x=0..5	AR
TCC2	EVx	46,47	EV x=0..1	AR
	MCx	48-53	MC x=0..5	AR
TCC3	EVx	54,55	EV x=0..1	AR
	MCx	56-61	MC x=0..5	AR
TCC4	EVx	62,63	EV x=0..1	AR
	MCx	64,65	MC x=0..1	AR
TCC5	EVx	66,67	EV x=0..1	AR
	MCx	68,69	MC x=0..1	AR
TCC6	EVx	70,71	EV x=0..1	AR
	MCx	72,73	MC x=0..1	AR
TCC7	EVx	74,75	EV x=0..1	AR
	MCx	76,77	MC x=0..1	AR
ADC	TRIGx	78-88	ADC TRIG x=0..10	AR
AC	SOCx	89,90	AC SOC x=0..1	AR
PTC	DSEQR	111	-	A
	STCONV	112	-	A
CCL	LUTIN_x	93-96	LUTx Input, x=0...3	???
PDEC	EVU_x	97-99	EVUx , x=0...2	???

Note:

1. A = Asynchronous path, R = Resynchronized path

34. I/O Pin Controller (PORT)

34.1. Overview

The I/O 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.

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-bit, 16-bit, or 32-bit write.

Note: If AVDD < 2.5V, the user must enable analog charge pumps in SUPC.VREGCTRL.CPEN[1:0].

34.2. Features

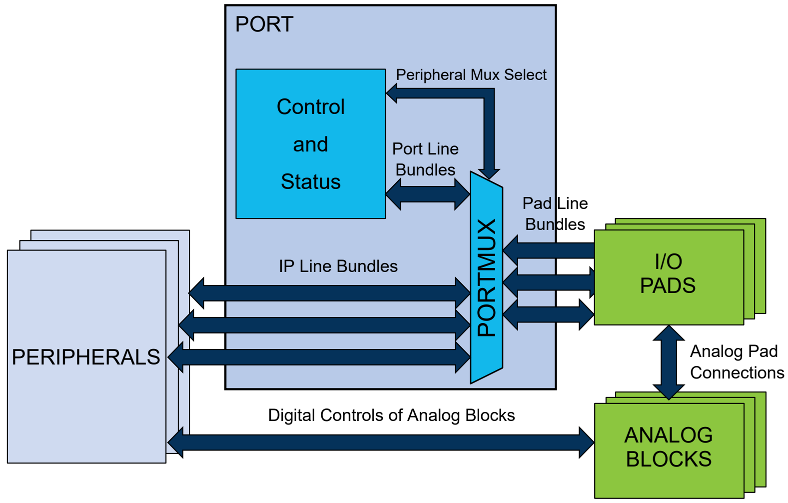
The following are key features of the PORT module:

- 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:
  - Totem-pole (push-pull)
  - Slew rate limit
  - Open drain
- Configurable input buffer and pull settings:
  - Internal pull-up or pull-down options
  - 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

Note: Selectable secured or non-secured attribution for each individual pin (**only on devices with security attribution**).

34.3. Block Diagram

Figure 34-1. PORT I/O Block Diagram



34.4. Signal Description

Table 34-1. Signal Description for PORT

Signal name	Type	Description
Pxn	Digital I/O	General-purpose I/O, Port x, Pin n (Example: PA1, PC20)

Refer to the Pinout for details on the pin mapping for this peripheral. One signal can be mapped on several pins.

34.5. Peripheral Dependencies

Peripheral Name	Base Address	NVIC IRQ Index: Source	MCLK AHBx/APBx Clock Enable Mask Bit	PAC Peripheral Identifier (PAC.WRCTRL.PERIDx)	EVSYS Users (EVSYS.USERm)	Power Domain
PORT	0x4480_0000	32:NSCHK_0, NSCHK_1, NSCHK_2, NSCHK_3	MCLK.CLKMSK2[0]	20	2-5 : EVx x=0,1,2,3	VDDREG

34.6. I/O Lines

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 PORTx, with letter x = A, B, C... Each pin of PORTx is identified by a number n = 0, 1, ...31 for the nth pin of that Port. In this way PORT pins are identified as 'Pxn' (for example: PA24, PC3).

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 additional information.

34.7. Clocks

The PORT bus clock (CLK_PORT_APB) can be enabled and disabled in the Main Clock module, and the default state of CLK_PORT_APB can be found in the Peripheral Clock Masking section of the MCLK – Main Clock.

The PORT requires an APB clock, which may be divided from the CPU main clock and allows the CPU to access the registers of PORT through the high-speed matrix and the AHB/APB bridge.

The PORT also requires an AHB clock for CPU AHBP accesses to the PORT, which have a higher priority than the APB accesses in case of concurrent PORT accesses. That AHB clock is the internal PORT clock.

34.8. CPU AHB Bus

The CPU AHB bus (AHBP) is an interface that connects the CPU directly to the PORT. 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.

34.9. Power Management

During Reset, all PORT lines are configured as inputs with input buffers, output buffers and pull disabled. The PORT peripheral will continue operating in any sleep mode where its source clock is running.

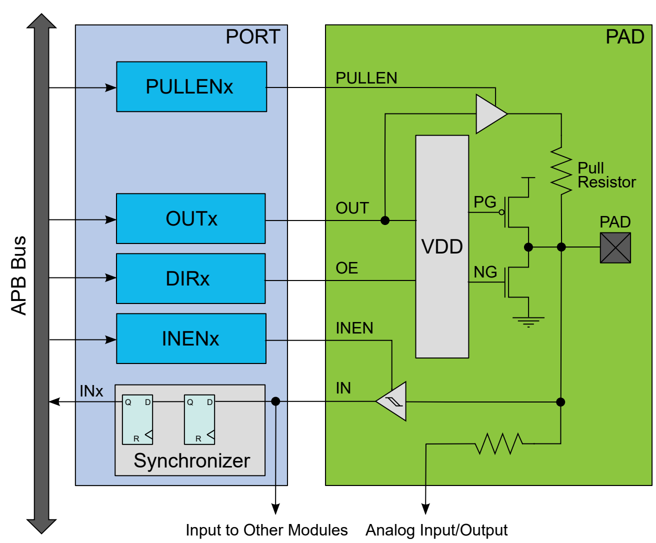
34.10. Debug Operation

When the CPU is halted in debug mode, this peripheral will continue normal operation.

In order to reduce CPU stalling when reading the IN register through the AHBP and waiting for input data synchronization, the Control register (CTRL) may be configured to continuous sampling of all pins to be read via the AHBP.

34.11. Functional Description

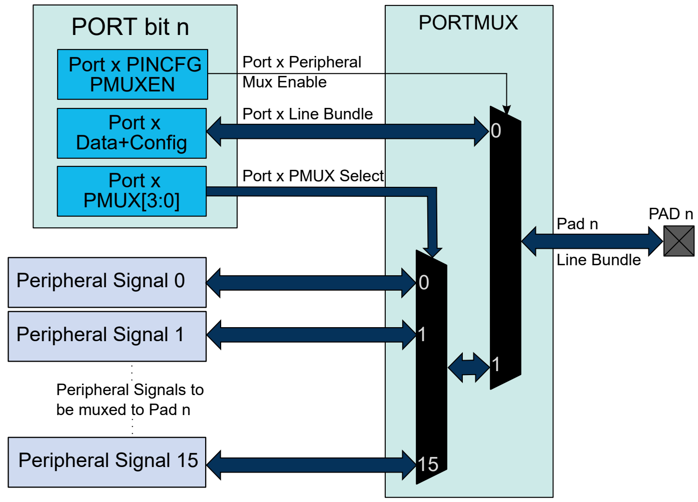
Figure 34-2. Overview of the PORT



34.11.1. 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 34-3. 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 n in OUT is written to '1', pin n is driven HIGH. If bit n in OUT is written to '0', pin n is driven LOW. Pin configuration can be set by Pin Configuration (PINCFGn) registers, with n=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 can be clocked only when system requires reading the input value, as specified in the SAMPLING field of the Control register (CTRL). 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 (PINCFGn.INEN) is '0', the input value will not be sampled.

In PORT, the Peripheral Multiplexer Enable bit in the PINCFGn register (PINCFGn.PMUXEN) can be written to '1' to enable the connection between peripheral functions and individual I/O pins. The Peripheral Multiplexing m (PMUXm, m=0,...15) registers select the peripheral function for the corresponding pin. This will override the connection between the PORT and that I/O pin, connecting the selected peripheral signal to the particular I/O pin instead of the PORT line bundle.

On devices with security attribution, 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.

34.11.2. Basic Operation

34.11.2.1. 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.

However, specific pins, such as those used for connection to a debugger, may be configured differently, as required by their special function.

34.11.2.2. Operation

Each I/O pin Pxn can be controlled by the registers in PORT. Each PORT group x has its own set of PORT registers, with a base address at byte address (PORT + 0x80 * group index) (A corresponds to group index 0, B to 1, and so on). Within that set of registers, the pin index is y, from 0 to 31.

Refer to the Pinout for additional information on available pin configuration and PORT groups.

Configuring Pins as Output

To use pin Pxn 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, this must be done first before writing the DIR bit to avoid glitching the output pin prior to setting the user desired OUT value, so the previous value of the OUT bit is not driven on the pin.

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 Pxn 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 (PINCFGn.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 cycles of the PORT clock. 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 Pxn as one of the available peripheral functions, the corresponding PMUXEN bit of the PINCFGn register must be '1'. The PINCFGn register for pin Pxn is at byte offset (PINCFG0 + n).

The peripheral function can be selected by setting the PMUXO or PMUXE in the PMUXm register. The PMUXO/ PMUXE is at byte offset PMUX0 + (n/2). The chosen peripheral must also be configured and enabled.

34.11.3. I/O Pin Configurations

The Pin Configuration register (PINCFGn) is used for additional I/O pin configuration. A pin can be set in a totem-pole, open-drain, 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 the following table:

Table 34-2. 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

Table 34-2. Pin Configurations Summary (continued)

DIR	INEN	PULLEN	OUT	Configuration
1	0	X	X	Output; input disabled
1	1	X	X	Output; input enabled

34.11.3.1. Input Configuration

Figure 34-4. I/O Configuration - Standard Input

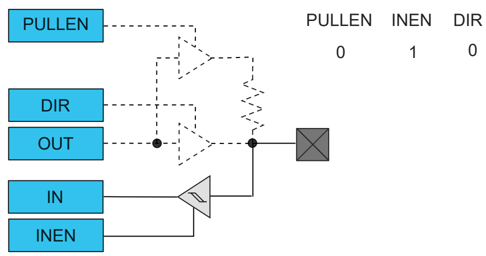
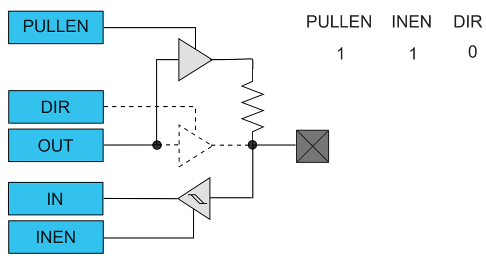


Figure 34-5. I/O Configuration - Input with Pull



Note: When pull is enabled, the pull value is defined by the OUT value.

34.11.3.2. 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 34-6. I/O Configuration - Totem-Pole Output with Disabled Input

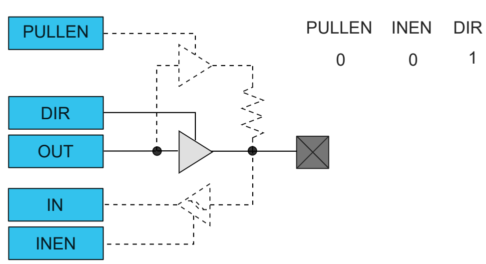


Figure 34-7. I/O Configuration - Totem-Pole Output with Enabled Input

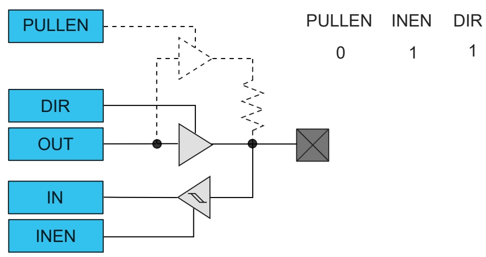


Figure 34-8. I/O Configuration - Output with Pull

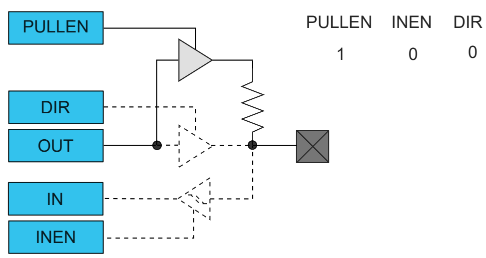
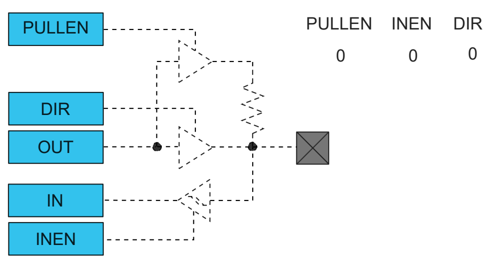


Figure 34-9. I/O Configuration - Reset or Analog I/O: Digital Output, Input and Pull Disabled



34.11.3.3. Secure Access Rights

Note: Only on devices with security attribution.

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 nonsecure 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.

34.11.3.4. Events

The PORT allows input events to control individual I/O pins. These input events 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.EVACTm). Writing a '1' to a PORT Event Enable Input m of the Event Control register (EVCTRL.PORTEIm) 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 Event System (EVSYS), for details on configuring the Event System.

Each event input can address 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 34-3. 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.

34.12. Register Summary

For descriptions and definitions of both Register and bitfield properties, refer to Register Properties.

> **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.

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			SLEWLIM[1:0]		ODRAIN	PULLEN	INEN	PMUXEN
		15:8	PINMASK[15:8]							
		7:0	PINMASK[7:0]							
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]				

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x30	PMUX0	7:0	PMUXO[3:0]				PMUXE[3:0]			
0x31	PMUX1	7:0	PMUXO[3:0]				PMUXE[3:0]			
0x32	PMUX2	7:0	PMUXO[3:0]				PMUXE[3:0]			
0x33	PMUX3	7:0	PMUXO[3:0]				PMUXE[3:0]			
0x34	PMUX4	7:0	PMUXO[3:0]				PMUXE[3:0]			
0x35	PMUX5	7:0	PMUXO[3:0]				PMUXE[3:0]			
0x36	PMUX6	7:0	PMUXO[3:0]				PMUXE[3:0]			
0x37	PMUX7	7:0	PMUXO[3:0]				PMUXE[3:0]			
0x38 ... 0x3F	Reserved									
0x40	PINCFG0	7:0			SLEWLIM[1:0]		ODRAIN	PULLEN	INEN	PMUXEN
0x41	PINCFG1	7:0			SLEWLIM[1:0]		ODRAIN	PULLEN	INEN	PMUXEN
0x42	PINCFG2	7:0			SLEWLIM[1:0]		ODRAIN	PULLEN	INEN	PMUXEN
0x43	PINCFG3	7:0			SLEWLIM[1:0]		ODRAIN	PULLEN	INEN	PMUXEN
0x44	PINCFG4	7:0			SLEWLIM[1:0]		ODRAIN	PULLEN	INEN	PMUXEN
0x45	PINCFG5	7:0			SLEWLIM[1:0]		ODRAIN	PULLEN	INEN	PMUXEN
0x46	PINCFG6	7:0			SLEWLIM[1:0]		ODRAIN	PULLEN	INEN	PMUXEN
0x47	PINCFG7	7:0			SLEWLIM[1:0]		ODRAIN	PULLEN	INEN	PMUXEN
0x48 ... 0x5F	Reserved									
0x60	INTENCLR	31:24								
		23:16								
		15:8								
		7:0								
0x64	INTENSET	31:24								
		23:16								
		15:8								
		7:0								
0x68	INTFLAG	31:24								
		23:16								
		15:8								
		7:0								
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]							

34.12.1. Data Direction

Name: DIR
Offset: 0x00
Reset: 0x00000000
Property: PAC Write-Protection

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.

Table 34-4. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	DIR[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
	DIR[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
	DIR[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
	DIR[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 – 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.

34.12.2. Data Direction Clear

Name: DIRCLR
Offset: 0x04
Reset: 0x00000000
Property: PAC Write-Protection

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.

Table 34-5. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	DIRCLR[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
	DIRCLR[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
	DIRCLR[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
	DIRCLR[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 – 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.

34.12.3. Data Direction Set

Name: DIRSET
Offset: 0x08
Reset: 0x00000000
Property: PAC Write-Protection

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.

Table 34-6. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	DIRSET[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
	DIRSET[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
	DIRSET[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
	DIRSET[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 – 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.

34.12.4. Data Direction Toggle

Name: DIRTGL
Offset: 0x0C
Reset: 0x00000000
Property: PAC Write-Protection

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.

Table 34-7. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	DIRTGL[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
	DIRTGL[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
	DIRTGL[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
	DIRTGL[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 – 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.

34.12.5. Data Output Value

Name: OUT
Offset: 0x10
Reset: 0x00000000
Property: PAC Write-Protection

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.

Table 34-8. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	OUT[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
	OUT[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
	OUT[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
	OUT[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 – 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.

34.12.6. Data Output Value Clear

Name: OUTCLR
Offset: 0x14
Reset: 0x00000000
Property: PAC Write-Protection

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.

Table 34-9. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	OUTCLR[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
	OUTCLR[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
	OUTCLR[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
	OUTCLR[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 – 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.

34.12.7. Data Output Value Set

Name: OUTSET
Offset: 0x18
Reset: 0x00000000
Property: PAC Write-Protection

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.

Table 34-10. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	OUTSET[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
	OUTSET[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
	OUTSET[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
	OUTSET[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 – 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.

34.12.8. Data Output Value Toggle

Name: OUTTGL
Offset: 0x1C
Reset: 0x00000000
Property: PAC Write-Protection

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.

Table 34-11. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	OUTTGL[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
	OUTTGL[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
	OUTTGL[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
	OUTTGL[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 – 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.

34.12.9. Data Input Value

Name: IN
Offset: 0x20
Reset: 0x40000000
Property: -

Table 34-12. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	IN[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
	IN[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
	IN[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
	IN[7:0]							
Access	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.

34.12.10. Control

Name: CTRL
Offset: 0x24
Reset: 0x00000000
Property: PAC Write-Protection

Table 34-13. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	SAMPLING[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
	SAMPLING[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
	SAMPLING[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
	SAMPLING[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 – 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. Therefore if any pins within a byte request continuous sampling, all pins in that eight pin sub-group will be continuously sampled.

Note: Bit 0 corresponds to Port A0/B0/C0, etc. Bit 1 to PORT A1, B1, C1, etc. All ports will not have 32 pins associated with them so some bits may not apply.

Value	Description
0	On demand sampling of I/O pin is enabled.
1	Continuous sampling of I/O pin is enabled.

34.12.11. Write Configuration

Name: WRCONFIG
Offset: 0x28
Reset: 0x00000000
Property: PAC Write-Protection, Write-Only

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.

Table 34-14. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	HWSEL	WRPINCFG		WRPMUX	PMUX[3:0]			
Access	W	W		W	W	W	W	W
Reset	0	0		0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
			SLEWLIM[1:0]		ODRAIN	PULLEN	INEN	PMUXEN
Access			W	W	W	W	W	W
Reset			0	0	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
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
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 PINCFG
This bit determines whether the atomic write operation will update the Pin Configuration register (PINCFGn) 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.SLEWLIM, WRCONFIG.ODRAIN, WRCONFIG.PULLEN, WRCONFIG.INEN, WRCONFIG.PMUXEN, and WRCONFIG.PINMASK values.

This bit will always read as zero.

Value	Description
0	The PINCFGn registers of the selected pins will not be updated.
1	The PINCFGn registers of the selected pins will be updated.

Bit 28 – WRPMUX Write PMUX

This bit determines whether the atomic write operation will update the Peripheral Multiplexing register (PMUXm, m=0,...15) 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.
Note: Not all PINCFGn registers are supported by all I/O port groups.

Value	Description
0	The PMUXm registers of the selected pins will not be updated.
1	The PMUXm 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 (PMUXm, m = 0,...15) 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.

Bits 21:20 – SLEWLIM[1:0] Output Driver Slew Rate Selection

This bit determines the new value written to PINCFGn.SLEWLIM 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.
Notes:

1. Only some pins have output TRISE/TFALL slew limit control.
2. Slew rate control can be used to improve signal integrity for high-speed signals if improper external HDW resistor termination was not utilized. One side effect however can be that if excessive slew is used it can affect maximum signal rates.
3. If an I^{2}C function is enabled on a pin, set PINCFGn.SLEWLIM = 0x00 (disabled).

Value	Description
0x00	Slew rate control disabled (fast rise/fall time operation)
0x01	Slew rate control enabled (4x slower)
0x02	Slew rate control enabled (8x slower)
0x03	Slew rate control enabled (12x slower)

Bit 19 – ODRAIN Open Drain Output

This bit determines the new value written to PINCFGn.ODRAIN 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.

Value	Description
0	Output pin is Totem Pole (i.e. Push-Pull) output
1	Open drain output is enabled

Bit 18 – PULLEN Pull Enable

This bit determines the new value written to PINCFGn.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.

Value	Description
0	Internal pin Pull-Up is disabled
1	Internal pin Pull-Up is enabled

Bit 17 – INEN Input Enable
This bit determines the new value written to PINCFGn.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.

Value	Description
0	Disable input pin function
1	Enable input pin function

Bit 16 – PMUXEN Peripheral Multiplexer Enable
This bit determines the new value written to PINCFGn.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.

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.

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.

34.12.12. Event Input Control

Name: EVCTRL
Offset: 0x2C
Reset: 0x00000000
Property: PAC Write-Protection, Secure

There are up to four input event pins for each PORT group. Each byte of this register addresses one Event input pin.

Table 34-15. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	PORTEI3	EVACT3[1:0]		PID3[4:0]				
Access	R/W	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
	PORTEI2	EVACT2[1:0]		PID2[4:0]				
Access	R/W	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
	PORTEI1	EVACT1[1:0]		PID1[4:0]				
Access	R/W	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
	PORTEI0	EVACT0[1:0]		PID0[4:0]				
Access	R/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, 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.

Table 34-16. 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.
0x3	TGL	Toggle output register of pin on event.

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 the following table.

Table 34-17. 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.

34.12.13. Peripheral Multiplexing m

Name: PMUXm
Offset: 0x30 + m*0x01 [m=0..7]
Reset: 0x00
Property: PAC Write-Protection

There are up to 16 Peripheral Multiplexing registers in each group, one for every set of two subsequent I/O lines. The m denotes the number of the set of I/O lines.

Table 34-18. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
	PMUXO[3:0]				PMUXE[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

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 PINCFGn.PMUXEN bit is '1'.
Not all possible values for this selection may be valid. For more details, refer to the Pinout.

Table 34-19. Peripheral Port Mux Control Mapping

Port PINCFGn.MUXEN value	Port WRCONFIG.PMUX value	Peripheral Function	Description
0	N/A	Port	Normal Port I/O functions
1	0x0	EIC	External Interrupts
1	0x1	ADCN/ADC/AC	ADC and Analog Comparator
1	0x2	SERCOM	SERCOMn (UART, I²C, SPI)
1	0x3	SERCOM Alt	SERCOMn (UART, I²C, SPI)
1	0x4	EBI	External Bus Interface
1	0x5	TCC	Timer/Counter controller
1	0x6	TCC Alt/PDEC	Timer/Counter controller and Positional Decoder
1	0x7	COM/CM33 Trace	SQI/CAN/USB/CM33 Trace
1	0x8	SDMMC	SD/MMC Host Controller (Memory Card Interface)
1	0x9	SPI_IXS	SPI_IXS Audio
1	0xa	PCC	Parallel Capture Controller
1	0xb	ETH	Ethernet
1	0xc	MISC	GCLK/CCL/AC Alt
1	0xd	PTC Debug	Peripheral Touch Controller and JTAG/SWD Debug Interfaces

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 PINCFGn.PMUXEN bit is '1'.
Not all possible values for this selection may be valid. For more details, refer to the Pinout.

Table 34-20. Peripheral Port Mux Control Mapping

Port PINCFGn.MUXEN value	Port WRCONFIG.PMUX value	Peripheral Function	Description
0	N/A	Port	Normal Port I/O functions
1	0x0	EIC	External Interrupts
1	0x1	ADCN/ADC/AC	ADC and Analog Comparator
1	0x2	SERCOM	SERCOMn (UART, I²C, SPI)
1	0x3	SERCOM Alt	SERCOMn (UART, I²C, SPI)
1	0x4	EBI	External Bus Interface
1	0x5	TCC	Timer/Counter controller
1	0x6	TCC Alt/PDEC	Timer/Counter controller and Positional Decoder
1	0x7	COM/CM33 Trace	SQI/CAN/USB/CM33 Trace
1	0x8	SDMMC	SD/MMC Host Controller (Memory Card Interface)
1	0x9	SPI_IXS	SPI_IXS Audio
1	0xa	PCC	Parallel Capture Controller
1	0xb	ETH	Ethernet
1	0xc	MISC	GCLK/CCL/AC Alt
1	0xd	PTC Debug	Peripheral Touch Controller and JTAG/SWD Debug Interfaces

Table 34-21. Peripheral Port Mux Control Mapping for PIC32CK1012 Variant Only

Port PINCFGn.MUXEN value	Port WRCONFIG.PMUX value	Peripheral Function	Description
0	N/A	Port	Normal Port I/O functions
1	0x0	EIC	External Interrupts
1	0x1	ADCN/ADC/AC	ADC and Analog Comparator
1	0x2	SERCOM	SERCOMn (UART, I2C, SPI)
1	0x3	SERCOM Alt	SERCOMn (UART, I2C, SPI)
1	0x4	TCC	Timer/Counter controller
1	0x5	TCC Alt/PDEC	Timer/Counter controller and Positional Decoder
1	0x6	COM/CM33 Trace	SQI/CAN/USB/CM33 Trace
1	0x7	SDMMC	SD/MMC Host Controller (Memory Card Interface)
1	0x8	SPI_IXS	SPI_IXS Audio
1	0x9	PCC	Parallel Capture Controller
1	0xa	ETH	Ethernet
1	0xb	MISC	GCLK/CCL/AC Alt
1	0xc	PTC/DEBUG	Peripheral Touch Controller and JTAG/SWD Debug Interfaces
1	0xd	ETH Alt	Ethernet (48-pin variant only)
1	0xe	SQI Alt	Serial Quad Interface (48-pin variant only)

34.12.14. Pin Configuration n

Name: PINCFGn
Offset: 0x40 + n*0x01 [n=0..7]
Reset: 0x00
Property: PAC Write-Protection

There are up to 32 Pin Configuration registers in each PORT group, one for each I/O line.

Note: All undefined values of WRCONFIG.PMUX are reserved.

Table 34-22. Port Pin Configuration Register Mapping

Port Group	Packages			
	144 pin TQFP	100 pin TQFP	64 pin TQFP	48 pin VQFN
PORT A (GROUP 0)	PA[0:28]	PA[0:18]	PA[0:12}	PA[0:12]
PINCFGn Registers	n=[0-28]	n=[0-18]	n=[0-12]	n=[0-12]
PORT B (GROUP 1)	PB[0:26]	PB[0:17]	PB[0:10]	PB[0:6]
PINCFGn Registers	n=[0-26]	n=[0-17]	n=[0-10]	n=[0-6]
PORT C (GROUP 2)	PC[0:29]	PC[0:20]	PC[0:13]	PC[0:11]
PINCFGn Registers	n=[0-29]	n=[0-20]	n=[0-13]	n=[0-11]
PORT D (GROUP 3)	PD[0:21]	PD[0:12]	PD[0:11]	PD[0:7]
PINCFGn Registers	n=[0-21]	n=[0-12]	n=[0-11]	n=[0-7]

EXAMPLE:

PORT_REGS->GROUP[2].PORT_PINCFG4 = 0x12; /* I/O pin PC4, Slew rate control enabled (4x slower), input buffer enabled */

Table 34-23. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
			SLEWLIM[1:0]		ODRAIN	PULLEN	INEN	PMUXEN
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bits 5:4 – SLEWLIM[1:0] Output Driver Slew Rate Selection

Notes:

1. Slew rate control can be used to improve signal integrity for high-speed signals if improper external HDW resistor termination was not utilized.
2. If an I^2C function is enabled on a pin, the corresponding PINCFGn.SLEWLIM MUST = 0x00.
3. Not all pins have slew rate control. See Table 34-23 for a list.

Value	Description
0	Slew rate control disabled (fast rise/fall time operation).
1	Slew rate control enabled (4x slower).
2	Slew rate control enabled (8x slower).
3	Slew rate control enabled (12x slower).

Bit 3 – ODRAIN Open Drain Output

Value	Description
0	The open drain output is disabled.
1	The open drain output is enabled.

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 (PMUXm, m=0,...15) 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 logic level via the Data Output Value register (OUT). The peripheral multiplexer value in PMUXm is ignored. Writing '1' to this bit enables the peripheral selection in PMUXm 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.

34.12.15. Interrupt Enable Clear

Name: INTENCLR
Offset: 0x60
Reset: 0x00000000
Property: PAC Write-Protection

Important: This register is only present in devices with security attribution.

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
Access								
Reset								

34.12.16. Interrupt Enable Set

Name: INTENSET
Offset: 0x64
Reset: 0x00000000
Property: PAC Write-Protection

Important: This register is only present on devices with security attribution.

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	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
Access								
Reset								

34.12.17. Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x68
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
Access								
Reset								

34.12.18. Security Attribution

Name: NONSEC
Offset: 0x6C
Reset: 0x00000000
Property: PAC Write-Protection

Important: This register is only present on devices with security attribution.

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	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
	NONSEC[23:16]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	NONSEC[15:8]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	NONSEC[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	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.

34.12.19. Security Attribution Check

Name: NSCHK
Offset: 0x70
Reset: 0x00000000
Property: PAC Write-Protection

Important: This register is only present on devices with security attribution.

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
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
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
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
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.

35. Serial Communication Interface (SERCOM)

35.1. Overview

There are up to eight instances of the Serial Communication interface (SERCOM) peripheral.

Each SERCOM provides hardware support for one of the I^{2}C, SPI, LIN, or USART protocols. It can operate as an I^{2}C Host, I^{2}C Client, SPI Host, SPI Client, LIN Host, LIN Client, or a USART. When a SERCOM is configured and enabled, the resources of that SERCOM are dedicated to the selected protocol and operational mode. It can use the internal generic clock or an external clock which allows it to operate in all Sleep modes.

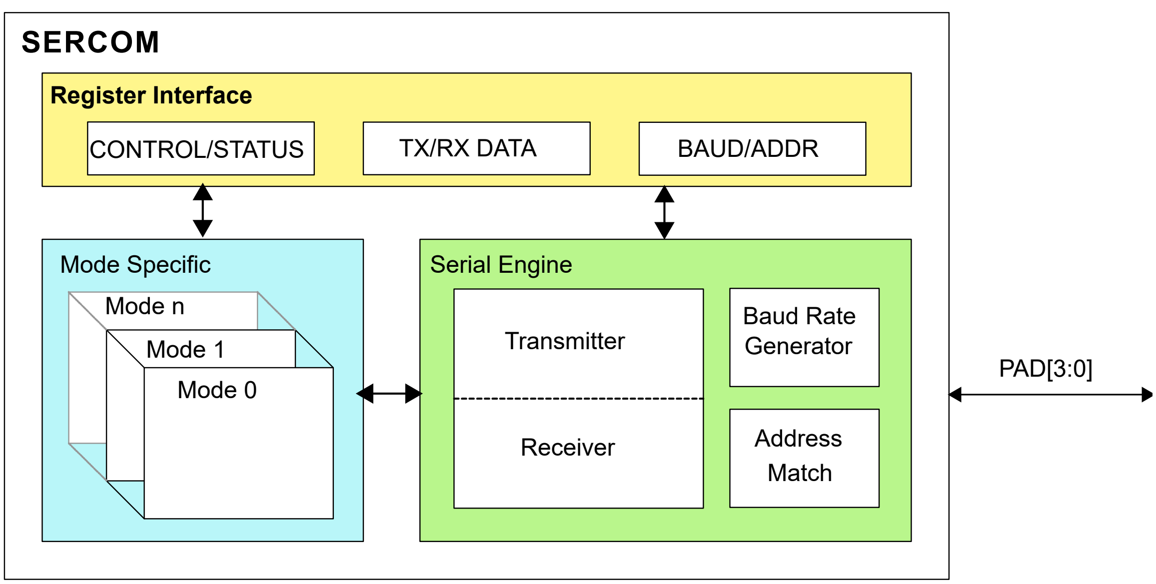
35.2. Features

The following are key features of the module:

- Interface for configuring operating mode:
  - Inter-Integrated Circuit (I^{2}C) two-wire serial interface
  - System Management Bus (SMBus™) compatible
  - Serial Peripheral Interface (SPI)
  - Universal Synchronous/Asynchronous Receiver/Transmitter (USART) (Plus LIN Host, LIN Client when operating in USART mode)
- Baud Rate Generator (BRG)
- Address Match/mask Logic
- Operational in all Sleep modes with an External Clock Source
- Can be used with DMA
- Up to 16-byte Internal FIFOs for Transmit and Receive

35.3. Block Diagram

Figure 35-1. SERCOM Block Diagram



35.4. Peripheral Dependencies

Peripheral Name	Base Address	NVIC IRQ Index: Source	MCLK AHBx/APBx **Clock Enable Mask Bit**	GCLK Peripheral Channel **Clock Name:Register**	PAC Peripheral Identifier **(PAC.WRCTRL.PERIDx)**	DMA Trigger Index:Source **(DMAC.CHCTRLBk.TRIGx)**	Power **Domain**	FIFO Size
SERCOM0	0x4481_0000	44 : ERROR, RXBRK 45 : DRE/PREC 46 : TXC/ AMATCH 47 : RXC/DRDY 48 : RXS/SSL/ TXFE 49 : CTSIC/RXFF	MCLK.CLKMKS2[7]	GCLK_SERCOM0_SLOW: GCLK.PCHCTRL[18] GCLK_SERCOM0_CORE : GCLK.PCHCTRL[19]	28	5 : RX 6 : TX	VDDCORE_SW	16 Bytes
SERCOM1	0x4481_2000	50 : ERROR, RXBRK 51 : DRE/PREC 52 : TXC/ AMATCH 53 : RXC/DRDY 54 : RXS/SSL/ TXFE 55 : CTSIC/RXFF	MCLK.CLKMKS2[8]	GCLK_SERCOM1_SLOW : GCLK.PCHCTRL[18] GCLK_SERCOM1_CORE : GCLK.PCHCTRL[20]	29	7 : RX 8 : TX	VDDCORE_SW	8 Bytes
SERCOM2	0x4481_4000	56 : ERROR, RXBRK 57 : DRE/PREC 58 : TXC/ AMATCH 59 : RXC/DRDY 60 : RXS/SSL/ TXFE 61 : CTSIC/RXFF	MCLK.CLKMKS2[9]	GCLK_SERCOM2_SLOW : GCLK.PCHCTRL[18] GCLK_SERCOM2_CORE : GCLK.PCHCTRL[21]	30	9 : RX 10 : TX	VDDCORE_SW	8 Bytes
SERCOM3	0x4481_6000	62 : ERROR, RXBRK 63 : DRE/PREC 64 : TXC/ AMATCH 65 : RXC/DRDY 66 : RXS/SSL/ TXFE 67 : CTSIC/RXFF	MCLK.CLKMKS2[10]	GCLK_SERCOM3_SLOW : GCLK.PCHCTRL[18] GCLK_SERCOM3_CORE : GCLK.PCHCTRL[22]	31	11 : RX 12 : TX	VDDCORE_SW	8 Bytes
SERCOM4	0x4500_0000	96 : ERROR, RXBRK 97 : DRE/PREC 98 : TXC/ AMATCH 99 : RXC/DRDY 100 : RXS/SSL/ TXFE 101 : CTSIC/ RXFF	MCLK.CLKMKS3[0]	GCLK_SERCOM4_SLOW : GCLK.PCHCTRL[18] GCLK_SERCOM4_CORE : GCLK.PCHCTRL[25]	36	41 : RX 42 : TX	VDDCORE_SW	8 Bytes

Peripheral Dependencies (continued)

Peripheral Name	Base Address	NVIC IRQ Index: Source	MCLK AHBx/APBx Clock Enable Mask Bit	GCLK Peripheral Channel Clock Name:Register	PAC Peripheral Identifier (PAC.WRCTRL.PERIDx)	DMA Trigger Index:Source (DMAC.CHCTRLBk.TRIGx)	Power Domain	FIFO Size
SERCOM5	0x4500_2000	102 : ERROR, RXBRK 103 : DRE/PREC 104 : TXC/ AMATCH 105 : RXC/DRDY 106 : RXS/SSL/ TXFE 107 : CTSIC/ RXFF	MCLK.CLKMKS3[1]	GCLK_SERCOM5_SLOW : GCLK.PCHCTRL[18] GCLK_SERCOM5_CORE : GCLK.PCHCTRL[26]	37	43 : RX 44 : TX	VDDCORE_SW	8 Bytes
SERCOM6	0x4500_4000	108 : ERROR, RXBRK 109 : DRE/PREC 110 : TXC/ AMATCH 111 : RXC/DRDY 112 : RXS/SSL/ TXFE 113 : CTSIC/ RXFF	MCLK.CLKMKS3[2]	GCLK_SERCOM6_SLOW : GCLK.PCHCTRL[18] GCLK_SERCOM6_CORE : GCLK.PCHCTRL[27]	38	45 : RX 46 : TX	VDDCORE_SW	8 Bytes
SERCOM7	0x4500_8000	114 : ERROR, RXBRK 115 : DRE/PREC 116 : TXC/ AMATCH 117 : RXC/DRDY 118 : RXS/SSL/ TXFE 119 : CTSIC/ RXFF	MCLK.CLKMKS3[3]	GCLK_SERCOM7_SLOW : GCLK.PCHCTRL[18] GCLK_SERCOM7_CORE : GCLK.PCHCTRL[28]	39	47 : RX 48 : TX	VDDCORE_SW	8 Bytes

35.4.1. I/O Lines

Using the SERCOM I/O lines requires the I/O pins to be configured using port configuration (PORT). The SERCOM has four internal pads, PAD[3:0], and the signals from I^{2}C, SPI and USART are routed through these SERCOM pads through a multiplexer. The configuration of the multiplexer is available from the different SERCOM modes. Refer to the mode specific sections below for additional information.

35.4.2. Power Management

The SERCOM can operate in any Sleep mode provided the selected clock source is running. SERCOM interrupts can be configured to wake the device from sleep modes.

35.4.3. Clocks

The SERCOM bus clock (CLK_SERCOMx_APB) can be enabled and disabled in the Main Clock Controller. The SERCOM APB BUS interface clocks are enabled by default on reset.

The SERCOM uses 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 sections below for details.

These clocks must be configured and enabled in the Generic Clock Controller (GCLK) before using the SERCOM.

The generic clocks are asynchronous to the user interface clock (CLK_SERCOMx_APB). Due to this asynchronicity, writing to certain registers will require synchronization between the clock domains. Refer to Synchronization for details.

35.4.4. DMA

The DMA request lines are connected to the DMA Controller (DMAC). The DMAC must be configured before the SERCOM DMA requests are used.

Concurrent DMA and CPU accesses to the DATA register must be avoided, as this may lead to unpredictable behavior.

35.4.5. 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. Refer to the Debug Control (DBGCTRL) register for details.

- USART.DBGCTRL
- SPI.DBGCTRL
- I^2C Host DGBCTRL

35.4.6. Register Access Protection

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

- Interrupt Flag Clear and Status register (INTFLAG)
- Status register (STATUS)
- Data register (DATA)
- Address register (ADDR)

Optional write protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write Protection" property in each individual register description.

PAC write protection does not apply to accesses through an external debugger.

35.5. Functional Description

35.5.1. Principle of Operation

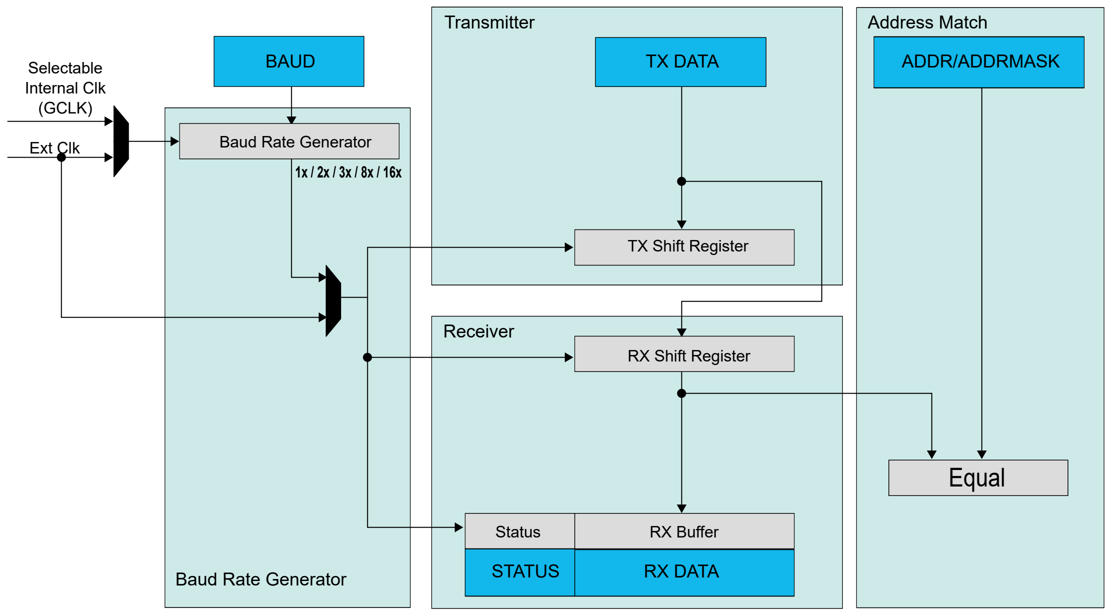
The basic structure of the SERCOM serial engine is shown in the following figure.

- The transmitter consists of a single write buffer and a Shift register
- The receiver consists of a one-level (I^2C), two-level (USART, SPI) receive buffer and a Shift register

The baud-rate generator is capable of running on the GCLK_SERCOMn_CORE clock or an external clock.

Address matching logic is included for SPI and I^2C operation.

Figure 35-2. SERCOM Serial Engine



35.5.2. Basic Operation

35.5.2.1. Initialization

The SERCOM must be configured to the desired mode by writing the Operating Mode bits in the Control A register (CTRLA.MODE).

Table 35-1. SERCOM Modes

CTRLA.MODE	Description
0x0	USART with external clock
0x1	USART with internal clock
0x2	SPI in client operation
0x3	SPI in host operation
0x4	I^{2}C client operation
0x5	I^{2}C host operation
0x6-0x7	Reserved

35.5.2.2. 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 descriptions (one for each mode of operation) for details.

- USART.CTRLA
- SPI.CTRLA
- I^{2}C Host CTRLA
- I^{2}C Client CTRLA

35.5.2.3. Clock Generation – Baud-Rate Generator

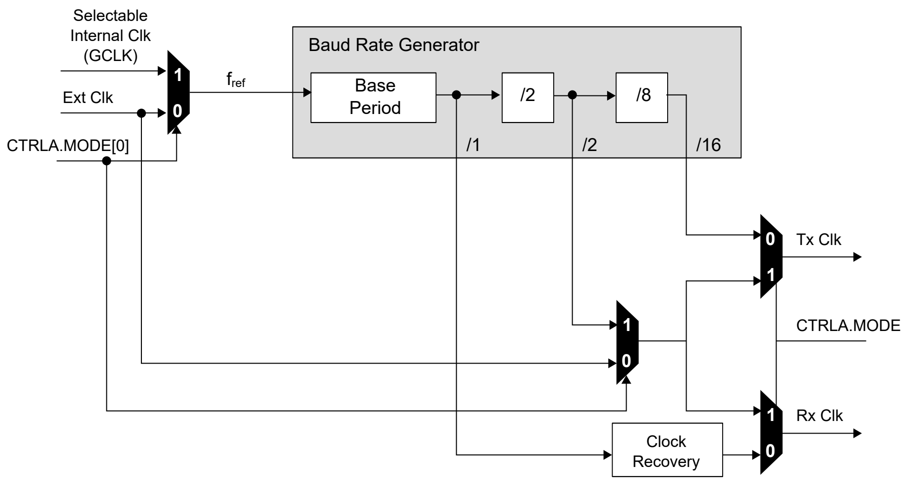
Each SERCOM contains a baud rate generator as shown in the following figure, which can generate an internal clock for asynchronous or synchronous communication. The output frequency (f_{BAUD}) is determined by the Baud register (BAUD) setting and the baud reference clock frequency (f_{ref}). The baud 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 35-3. 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 Arithmetic mode, the BAUD register value is 16 bits (0 to 65,535).

For Asynchronous Fractional mode, the BAUD register value contains 13 bits of integer (BAUD.BAUD) as well as a 3 bit fractional adjustment (BAUD.FP). In this mode the BAUD setting must be greater than or equal to 1. This mode is used for auto baud detection. See CTRLA.FORM and CRTLA.SAMPR.

For Synchronous operation, the BAUD register value is 8 bits (0 to 255). CRTLA.CMODE=1.

Table 35-2. Baud Rate Equations

Operating Mode	Condition	Baud Rate (Bits Per Second)	BAUD Register Value Calculation
Asynchronous Arithmetic	$f_{BAUD} \leq \frac{f_{ref}}{16}$	$f_{BAUD} = \frac{f_{ref}}{16}\left(1 - \frac{BAUD}{65536}\right)$	$BAUD = 65536 \cdot \left(1 - 16 \cdot \frac{f_{BAUD}}{f_{ref}}\right)$
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 baud rate error is represented by the following formula:

$$\text{Error} = 1 - \left(\frac{\text{ExpectedBaudRate}}{\text{ActualBaudRate}}\right)$$

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.

The following table shows the BAUD register value versus baud frequency f_{BAUD} at a serial engine frequency of 48 MHz. This assumes a *D* value of 8 bits and an *S* value of 2 bits (10 bits, including start and stop bits).

Table 35-3. BAUD Register Value vs. Baud Frequency

BAUD Register Value	Serial Engine CPF	f_{BAUD} at 48MHz Serial Engine Frequency (f_{REF})
0 – 406	160	3 MHz
407 – 808	161	2.981 MHz
809 – 1205	162	2.963 MHz
...	...	...
65206	31775	15.11 kHz
65207	31871	15.06 kHz
65208	31969	15.01 kHz

35.5.3. Additional Features

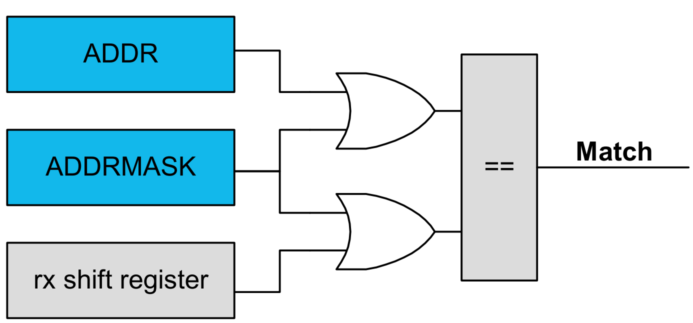
35.5.3.1. Address Match and Mask

In either SPI client or I^2C client mode, 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.

35.5.3.1.1. Address With Mask (CTRLB.AMODE = 0X0)

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 if the bit values in the shift register match the values in the Address register OR the associated bit in the Address Mask register is a 1. 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.

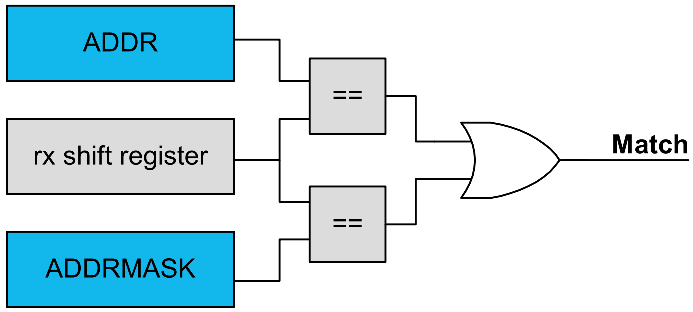
Figure 35-4. Address With Mask



35.5.3.1.2. Two Unique Addresses (CTRLB.AMODE = 0X1)

Either of the two addresses written to ADDR and ADDRMASK will cause a match.

Figure 35-5. Two Unique Addresses



35.5.3.1.3. Address Range (CTRLB.AMODE = 0X2)

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 35-6. Address Range

35.5.3.1.4. 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.

35.5.3.1.5. 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.

35.6. SERCOM USART - SERCOM Synchronous and Asynchronous Receiver and Transmitter

35.6.1. 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, BAUD Rate Generator, and either an internal or external clock.

The transmitter consists of a 16-byte FIFO, a Shift register, and control logic for different frame formats. The write buffer supports data transmission without any delay between frames. The receiver consists of a 16-byte FIFO 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.

35.6.2. USART Features

The USART features are as follows:

- Full-duplex Operation
- Asynchronous (with Clock Reconstruction) or Synchronous Operation
- Internal or External Clock source for Asynchronous and Synchronous Operation
- Baud-Rate Generator (BRG)
- Supports Serial Frames with 5, 6, 7, 8 or 9 Data bits and 1 or 2 Stop bits
- Odd or Even Parity Generation and Parity Check
- Selectable LSB or MSB first Data Transfer
- Buffer Overflow and Frame Error Detection
- Noise Filtering, Including False Start bit Detection and Digital Low-pass Filter
- Collision Detection
- Can Operate in all Sleep modes
- Operation at Speeds up to Half the System Clock for Internally Generated Clocks
- Operation at Speeds up to the System Clock for Externally Generated Clocks
- RTS and CTS Flow Control
- IrDA Modulation and Demodulation up to 115.2 kbps
- LIN Host Support
- LIN Client Support
  - Auto-baud and break character detection
- ISO 7816 T = 0 or T = 1 protocols for Smart Card Interfacing
- RS485 support
- Start-of-frame detection
- Up to 16-byte Receive FIFO with FIFO enabled
- Up to 16-byte Transmit FIFO with FIFO enabled
- Can work with DMA
- 32-bit Extension for Better System Bus Utilization

35.6.3. Block Diagram

Figure 35-7. USART Block Diagram

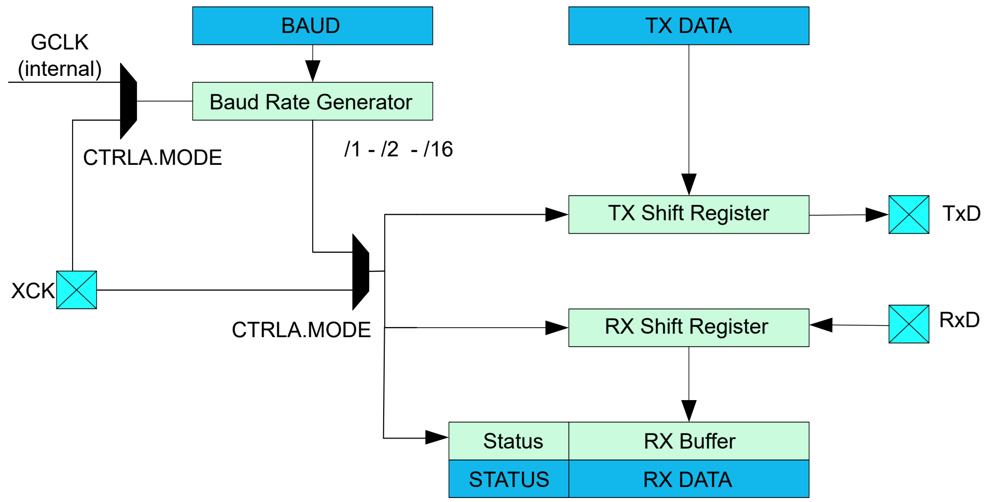


Table 35-4. USART Pin Configuration

Pin	Pin Configuration
TxD	Output
RxD	Input
XCK	Output or input

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 pad assignments of the USART signals as shown in the table above.

35.6.4. Functional Description

35.6.4.1. Principle of Operation

The USART uses the following lines for data transfer:

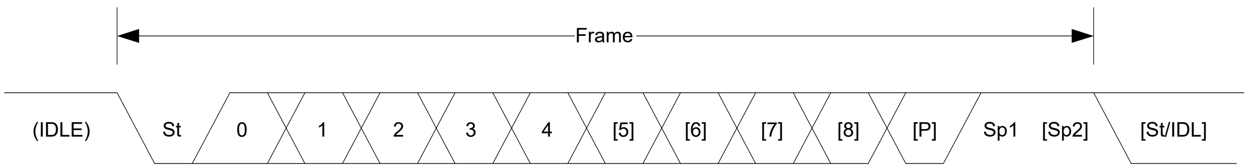
- RxD for receiving
- TxD for transmitting
- XCK for the transmission clock in synchronous operation

USART data transfer is frame based. A serial frame consists of the following:

- 1 start bit
- From 5 to 9 data bits (MSB or LSB first)
- None, an even or an odd parity bit
- 1 or 2 stop bits

A frame starts with the Start bit followed by one character of Data bits. If enabled, the parity bit is inserted after the Data bits and before the first Stop bit. After the stop bit(s) of a frame, either the next frame can follow immediately, or the communication line can return to the Idle (high) state. The following figure illustrates the possible frame formats. Brackets denote optional bits.

Figure 35-8. 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.

35.6.4.2. Basic Operation

35.6.4.2.1. 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)

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 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).

35.6.4.2.2. 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.

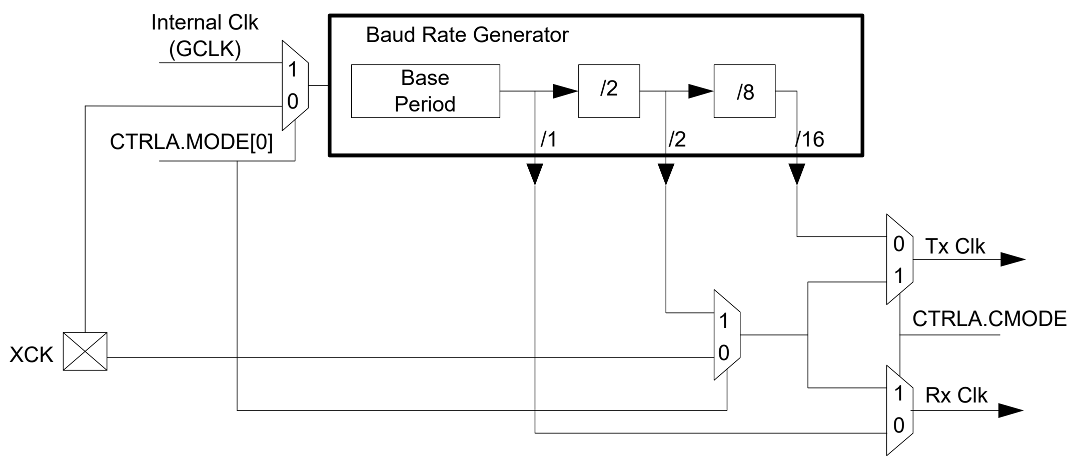
The Synchronous mode is selected by writing a '1' to the Communication Mode bit in the Control A register (CTRLA.CMODE), the Asynchronous mode is selected by writing a zero to CTRLA.CMODE.

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 35-9. Clock Generation



Synchronous Clock Operation

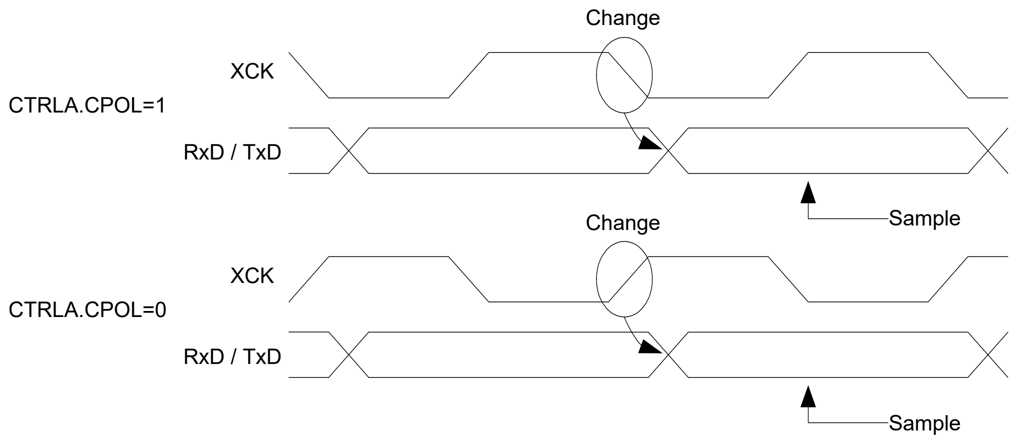
In Synchronous mode, the CTRLA.MODE bit field determines whether the transmission clock line (XCK) serves either as input or output. The dependency between clock edges, data sampling, and data change is the same for internal and external clocks. Data input on the RxD pin is sampled at the opposite XCK 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.

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, therefore can operate at frequencies up to the system frequency.

Figure 35-10. Synchronous Mode XCK Timing



35.6.4.2.3. 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.

35.6.4.2.4. Data Transmission

Data transmission is initiated by writing the data to be sent into the DATA register. Then, the data in TxDATA will be moved to the Shift register when the Shift register is empty and ready to send a new frame. After the Shift register is loaded with data, the data frame will be transmitted.

When the entire data frame including Stop bit(s) has been transmitted and no new data was written to DATA, 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.

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 TxDATA to transmit.

35.6.4.2.5. 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.

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.

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.

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.

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 following table.

Table 35-5. 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:

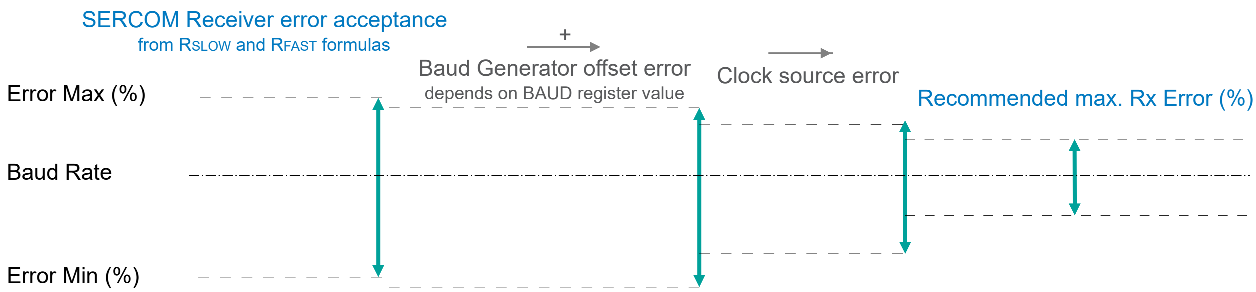
$$R_{\mathrm{SLOW}} = \frac{(D+1)S}{S-1+D\cdot S+S_F} \quad , \qquad R_{\mathrm{FAST}} = \frac{(D+2)S}{(D+1)S+S_M}$$

- 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.

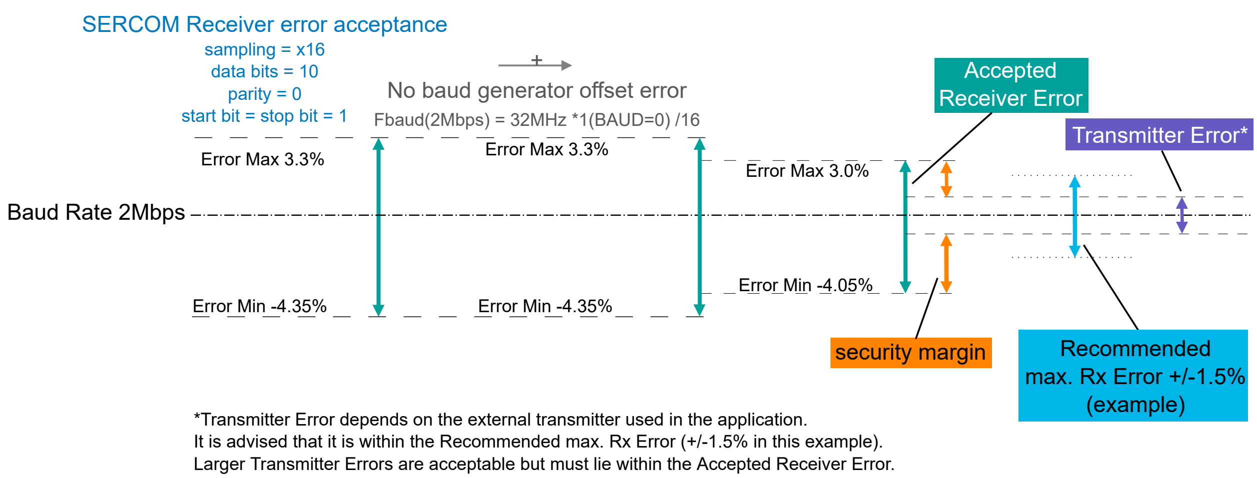
The recommended maximum Rx Error assumes that the receiver and transmitter equally divide the maximum total error. Its connection to the SERCOM Receiver error acceptance is depicted in this figure:

Figure 35-11. 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 35-12. USART Rx Error Calculation Example



35.6.4.3. Additional Features

35.6.4.3.1. Parity

Even or odd parity can be selected for error checking by writing 0x1 or 0x5 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.

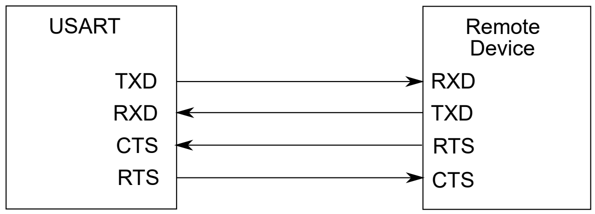
If *odd parity* is selected (CTRLB.PMODE = 1), the Parity bit of an outgoing frame is '1' if the data contains an even number of bits that are '0', making the total number of '1' odd.

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.

35.6.4.3.2. 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 following figure.

Figure 35-13. Connection with a Remote Device for Hardware Handshaking

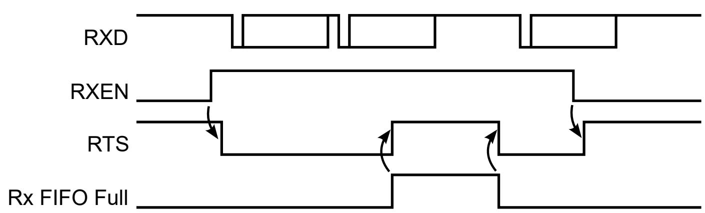


Hardware handshaking is only available in the following configuration:

- USART with internal clock (CTRLA.MODE=1),
- Asynchronous mode (CTRLA.CMODE=0), and
- Flow control pinout (CTRLA.TXPO=2).

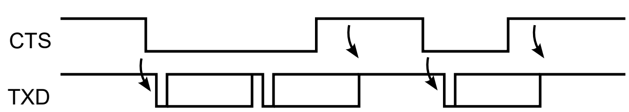
When the receiver is disabled or the receive FIFO is full, the receiver will drive the RTS pin high. This notifies the remote device to stop transfer after the ongoing transmission. Enabling and disabling the receiver by writing to CTRLB.RXEN will set/clear the RTS pin after a synchronization delay. 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 35-14. 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 35-15. Transmitter Behavior when Operating with Hardware Handshaking



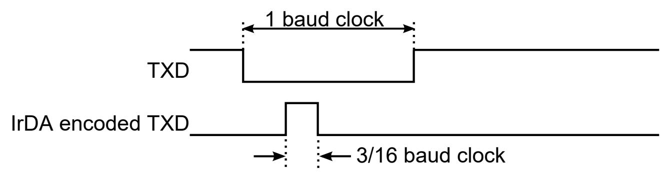
35.6.4.3.3. IrDA Modulation and Demodulation

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)

During transmission, each low bit is transmitted as a high pulse. The pulse width is 3/16 of the baud rate period, as illustrated in the following figure.

Figure 35-16. IrDA Transmit Encoding



The reception decoder has two main functions.

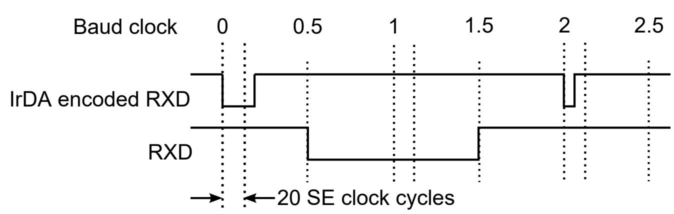
The first is to synchronize the incoming data to the IrDA baud rate counter. Synchronization is performed at the start of each zero pulse.

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 following figure 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 35-17. IrDA Receive Decoding



35.6.4.3.4. Break Character Detection and Auto-Baud/LIN Client

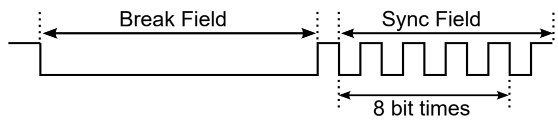
Break character detection and auto-baud are available in this configuration:

- Auto-baud frame format (CTRLA.FORM = 0x04 or 0x05),
- Asynchronous mode (CTRLA.CMODE = 0),
- 8 bits in a character (CTRLB.CHSIZE = 0x0), 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 ‘0’ 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 35-18. 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.

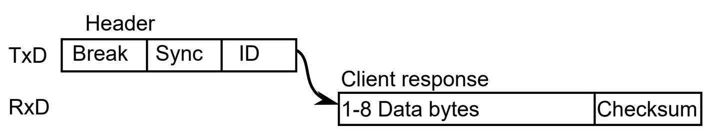
35.6.4.3.5. SERCOM USART LIN Host

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 35-19. 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:

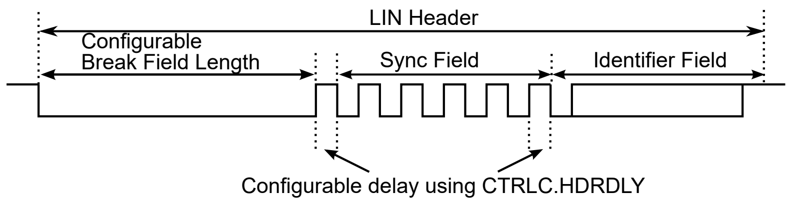
- CTRLB.LINCMD is written to 0x1
- DATA register written to 0x00. This triggers transmission of the break field by hardware. Note that writing the DATA register with any other value will also result in the transmission of the break field by hardware.
- DATA register written to 0x55. The 0x55 value (sync) 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 35-20. LIN Header Generation



After header transmission is complete, the client responds with 1-8 data bytes plus checksum.

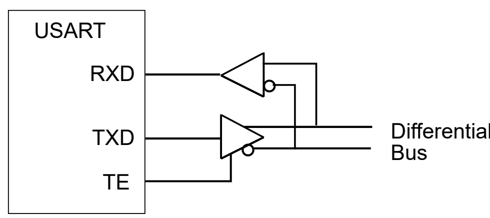
35.6.4.3.6. RS485

RS485 is available with the following configuration:

- USART frame format (CTRLA.FORM = 0x00 or 0x01)
- RS485 pinout (CTRLA.TXPO=0x3).

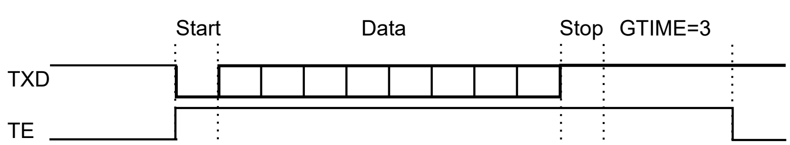
The RS485 feature enables control of an external line driver as shown in the following figure. While operating in RS485 mode, the transmit enable pin (TE) is driven high when the transmitter is active.

Figure 35-21. 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 35-22. 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.

35.6.4.3.7. 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.

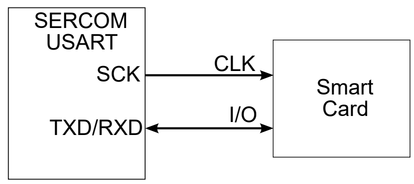
ISO 7816 is available with the following configuration:

- 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 35-23. Connection of a Smart Card to the SERCOM USART



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

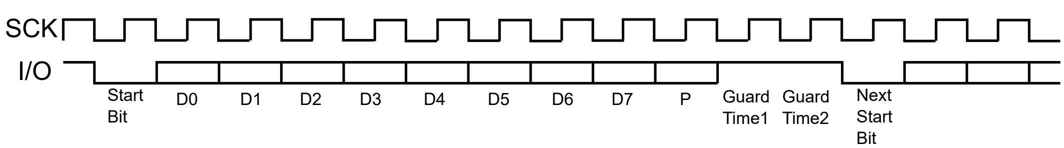
In T=0 protocol, a character is made up of:

- 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).

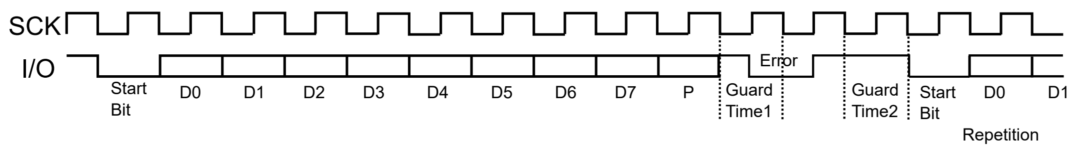
If no parity error is detected, the I/O line remains during the guard time and the transmitter can continue with the transmission of the next character, as shown in the following figure.

Figure 35-24. 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 following figure. This error bit is also named NACK, for Non Acknowledge. In this case, the character lasts 1 bit time more, as the guard time length is the same and is added to the error bit time, which lasts 1 bit time.

Figure 35-25. 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 (CTRLA.FORM=0x7) T=1 (CTRLA.FORM=0x7), the transmission is asynchronous (CTRLA.CMODE=0) with one or two stop bits. After the stop bits are sent, the transmitter does not drive the I/O line.

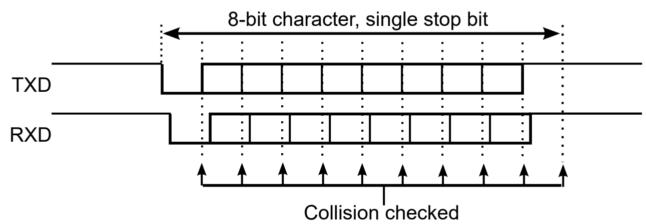
Parity is generated when transmitting and checked when receiving. Parity error detection sets the STATUS.PERR bit, and the erroneous character is written to the receive FIFO. When using T=1 protocol, the receiver does not signal errors on the I/O line and the transmitter does not retransmit.

35.6.4.3.8. 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).

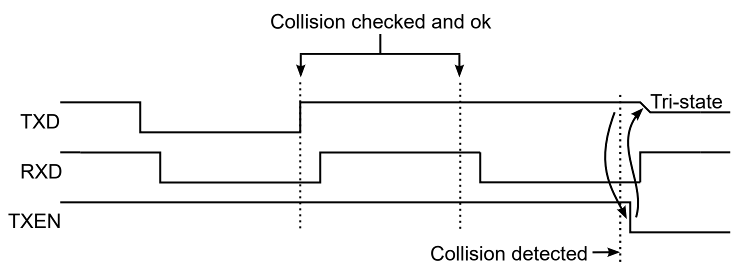
Collision detection is performed for each bit transmitted by comparing the received value with the transmit value, as shown in the figure below. While the transmitter is idle (no transmission in progress), characters can be received on RxD without triggering a collision.

Figure 35-26. 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.

Figure 35-27. 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.

35.6.4.3.9. 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.

35.6.4.3.10. Start-of-Frame Detection

The USART start-of-frame detector can wake-up the CPU when it detects a Start bit. In Standby Sleep mode, the internal fast start-up oscillator must be selected as the GCLK_SERCOMx_CORE source.

When a 1-to-0 transition is detected on RxD, the 8 MHz Internal Oscillator is powered up and the USART clock is enabled. After start-up, the rest of the data frame can be received, provided that the baud rate is slow enough in relation to the fast start-up internal oscillator start-up time. Refer to the *Electrical Characteristics* chapters for details. The start-up time of this oscillator varies with supply voltage and temperature.

The USART start-of-frame detection works both in Asynchronous and Synchronous modes. It is enabled by writing '1' to the Start of Frame Detection Enable bit in the Control B register (CTRLB.SFDE).

If the Receive Start Interrupt Enable bit in the Interrupt Enable Set register (INTENSET.RXS) is set, the Receive Start interrupt is generated immediately when a start is detected.

When using start-of-frame detection without the Receive Start interrupt, start detection will force the 8 MHz internal oscillator and USART clock active while the frame is being received. In this case, the CPU will not wake up until the receive complete interrupt is generated.

35.6.4.3.11. 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.

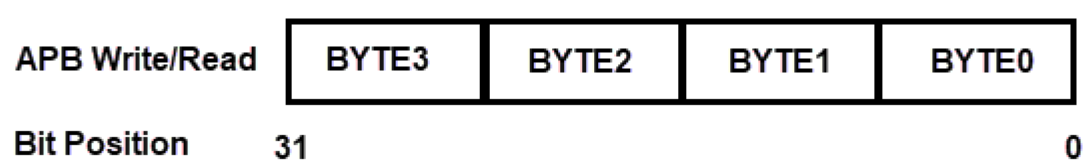
35.6.4.3.12. 32-bit Extension

For better system bus utilization, 32-bit data receive and transmit can be enabled separately by writing to the Data 32-bit 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.

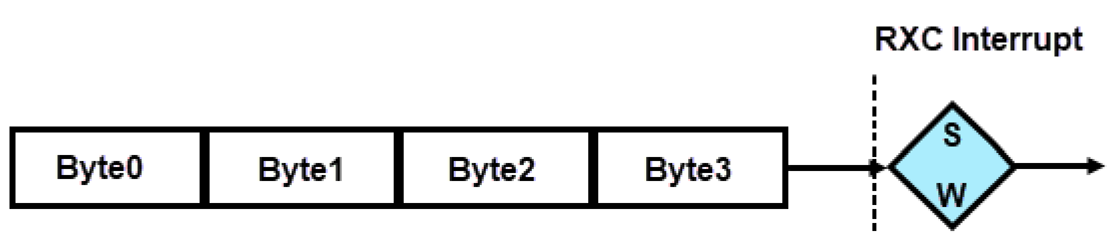
The figure below shows the order of transmit and receive when using 32-bit extension. Bytes are transmitted or received, and stored in order from 0 to 3. Only 8-bit and smaller character sizes are supported. If the character size is less than 8 bits, characters will still be 8-bit aligned within the 32-bit APB write or read. The unused bits within each byte will be zero for received data and unused for transmit data.

Figure 35-28. 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 35-29. 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.

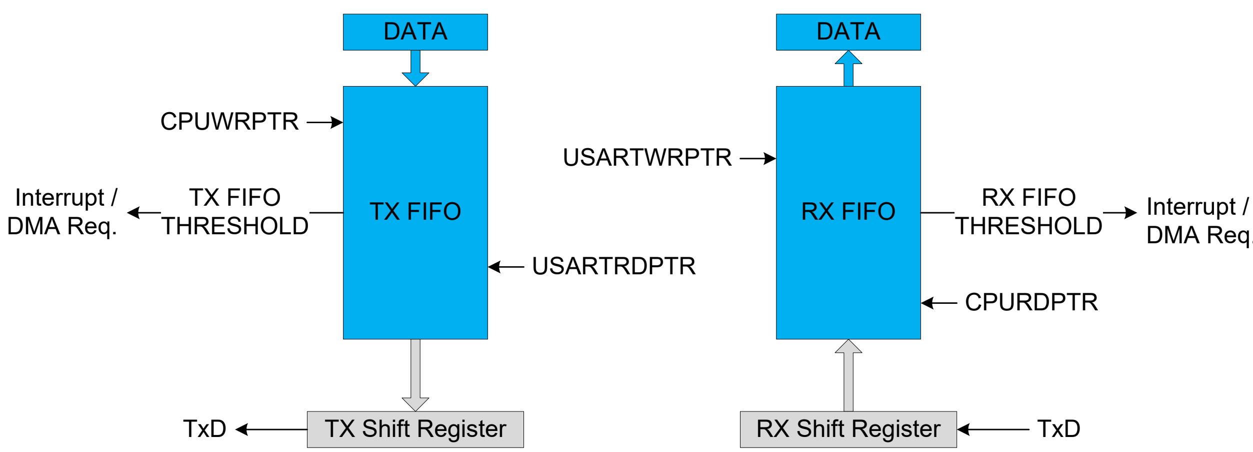
Writing the LENGTH register while a frame is in progress will produce unpredictable results. If LENGTH.LENEN is not set and a frame is not a multiple of 4 Bytes, the remainder may be lost. Attempting to use the length counter for transmit and receive at the same time will produce unpredictable results.

35.6.4.3.13. 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 35-30. 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 respectively, 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 respectively, 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 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.

If the USART is halted when debugging, the RX FIFO CPU read pointer can be accessed by writing the CPURDPTR bits in FIFOPTR register (FIFOPTR.CPURDPTR). These bits will not increment if new data is read from the RX FIFO memory.

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.

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 writing the CPURDPTR bits in FIFOPTR register (FIFOPTR.CPURDPTR). These bits will not increment if a new data is read from the RX FIFO memory.

35.6.4.3.14. DMA and Interrupts

Table 35-6. Module Request for SERCOM USART

Condition	Request	
	DMA	Interrupt
Standard (DRE): Data Register Empty FIFO (DRE): at least TXTRHOLD locations in TX FIFO are empty	Yes (request cleared when data is written)	Yes

Table 35-6. Module Request for SERCOM USART (continued)

Condition	Request	
	DMA	Interrupt
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	N/A	Yes
Receive Start (RXS)	N/A	Yes
Clear to Send Input Change (CTSIC)	N/A	Yes
Receive Break (RXBRK)	N/A	Yes
Error (ERROR)	N/A	Yes

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.

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 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 the Nested Vector Interrupt Controller for details.

35.6.4.4. Sleep Mode Operation

The behavior in sleep mode depends 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 Start and 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 Start and 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.

35.6.4.5. Synchronization

Due to asynchronicity between the main clock domain and the peripheral clock domains, some SERCOM registers need to be synchronized when written ("Write-Synchronized") or read ("Read-Synchronized").

The following bits are synchronized when written:

- Software Reset bit in the CTRLA register (CTRLA.SWRST)
- Enable bit in the CTRLA register (CTRLA.ENABLE)
- Receiver Enable bit in the CTRLB register (CTRLB.RXEN)
- Transmitter Enable bit in the Control B register (CTRLB.TXEN)
- LENGTH

The following bit is synchronized when read:

- RXERRCNT

Note: CTRLB.RXEN is write-synchronized somewhat differently. See the Register Summary for CTRLB details.

Required write synchronization is denoted by the "Write-Synchronized" property in the register description. If a write-synchronized register is written while a synchronization is ongoing, a Bus Error exception will be generated.

35.6.5. Register Summary

For descriptions and definitions of both Register and bitfield properties, refer to Register Properties.

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]			

35.6.5.1. Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Table 35-7. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
		DORD	CPOL	CMODE	FORM[3:0]			
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	SAMPA[1:0]		RXPO[1:0]				TXPO[1:0]	
Access	R/W	R/W	R/W	R/W			R/W	R/W
Reset	0	0	0	0			0	0

Bit	15	14	13	12	11	10	9	8
	SAMPR[2:0]					RXINV	TXINV	IBON
Access	R/W	R/W	R/W			R/W	R/W	R/W
Reset	0	0	0			0	0	0

Bit	7	6	5	4	3	2	1	0
	RUNSTDBY			MODE[2:0]			ENABLE	SWRST
Access	R/W			R/W	R/W	R/W	R/W	R/W
Reset	0			0	0	0	0	0

Bit 30 – DORD Data Order

This bit selects the data order when a character is shifted out from the Data register.
This bit is not synchronized.

Value	Description
0	MSB is transmitted first.
1	LSB is transmitted first.

Bit 29 – CPOL Clock Polarity

This bit selects the relationship between data output change and data input sampling in synchronous mode.
This bit is not synchronized.

CPOL	TxD Change	RxD Sample
0x0	Rising XCK edge	Falling XCK edge
0x1	Falling XCK edge	Rising XCK edge

Bit 28 – CMODE Communication Mode

This bit selects asynchronous or synchronous communication.
This bit is not synchronized.

Value	Description
0	Asynchronous communication.
1	Synchronous communication.

Bits 27:24 – FORM[3:0] Frame Format
These bits define the frame format.
These bits are not synchronized.

FORM[3:0]	Description
0x0	USART frame
0x1	USART frame with parity
0x2	LIN Hosst - Break and sync generation. See LIN Command (CTRLB.LINCMD).
0x3	Reserved
0x4	Auto-baud (LIN Client) - break detection and auto-baud.
0x5	Auto-baud - break detection and auto-baud with parity
0x6	Reserved
0x7	ISO 7816
0x8-0xF	Reserved

Bits 23:22 – SAMPA[1:0] Sample Adjustment
These bits define the sample adjustment.
These bits are not synchronized.

SAMPA[1:0]	16x Over-sampling (CTRLA.SAMPR=0 or 1)	8x Over-sampling (CTRLA.SAMPR=2 or 3)
0x0	7-8-9 (Recommended)	3-4-5 (Recommended)
0x1	9-10-11	4-5-6
0x2	11-12-13	5-6-7
0x3	13-14-15	6-7-8

Bits 21:20 – RXPO[1:0] Receive Data Pinout
These bits define the receive data (RxD) pin configuration.
These bits are not synchronized.

RXPO[1:0]	Name	Description
0x0	PAD[0]	SERCOM PAD[0] is used for data reception
0x1	PAD[1]	SERCOM PAD[1] is used for data reception
0x2	PAD[2]	SERCOM PAD[2] is used for data reception
0x3	PAD[3]	SERCOM PAD[3] is used for data reception

Bits 17:16 – TXPO[1:0] Transmit Data Pinout
These bits define the transmit data (TxD) and XCK pin configurations.
This bit is not synchronized.

TXPO	TxD Pin Location	XCK Pin Location (When Applicable)	RTS/TE	CTS
0x0	SERCOM PAD[0]	SERCOM PAD[1]	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.
These bits are not synchronized.

SAMPR[2:0]	Description
0x0	16x over-sampling using arithmetic baud rate generation.
0x1	16x over-sampling using fractional baud rate generation.
0x2	8x over-sampling using arithmetic baud rate generation.

SAMPR (continued)

SAMPR[2:0]	Description
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.

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.

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.

Value	Description
0	STATUS.BUFOVF is asserted when it occurs in the data stream.
1	STATUS.BUFOVF is asserted immediately upon buffer overflow.

Bit 7 – RUNSTDBY Run In Standby

This bit defines the functionality in standby sleep mode.

This bit is not synchronized.

RUNSTDBY	External Clock	Internal Clock
0x0	External clock is disconnected when ongoing transfer is finished. All reception is dropped.	Generic clock is disabled when ongoing transfer is finished. The device will not wake up on either Receive Start or Transfer Complete interrupt unless the appropriate ONDEMAND bits are set in the clocking chain.
0x1	Wake on Receive Start or Receive Complete interrupt.	Generic clock is enabled in all sleep modes. Any interrupt can wake up the device.

Bits 4:2 – MODE[2:0] Operating Mode

These bits select the USART serial communication interface of the SERCOM.

These bits are not synchronized.

Value	Description
0x0	USART with external clock
0x1	USART with internal clock

Bit 1 – ENABLE Enable

Due to synchronization, there is delay from writing CTRLA.ENABLE until the peripheral is enabled/disabled. The value written to CTRLA.ENABLE will read back immediately and the Enable Synchronization Busy bit in the Synchronization Busy register (SYNCBUSY.ENABLE) will be set. SYNCBUSY.ENABLE is cleared when the operation is complete.

This bit is not enable-protected.

Value	Description
0	The peripheral is disabled or being disabled.
1	The peripheral is enabled or being enabled.

Bit 0 – SWRST Software Reset

Writing '0' to this bit has no effect.

Writing '1' to this bit resets all registers in the SERCOM, except DBGCTRL, to their initial state, and the SERCOM 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 a bus error. Reading any register will return the reset value of the register.

Due to synchronization, there is a delay from writing CTRLA.SWRST until the reset is complete. CTRLA.SWRST and SYNCBUSY.SWRST will both be cleared when the reset is complete.

This bit is not enable-protected.

Notes:

1. When the CTRLA.SWRST is written, the user should poll the SYNCB.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 the hardware.

Value	Description
0	There is no reset operation ongoing.
1	The reset operation is ongoing.

35.6.5.2. Control B

Name: CTRLB
Offset: 0x04
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected, Write-Synchronized

Write to this register only when SYNCBUSY.CTRLB = 0, otherwise a bus error will result.

Table 35-8. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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.
These bits are not enable-protected.

Value	Description
`0x0`	Normal USART transmission.
`0x1`	Break field is transmitted when DATA is written.
`0x2`	Break, sync and identifier are automatically transmitted when DATA is written with the identifier.
`0x3`	Reserved

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.
These bits are not enable-protected.

FIFOCLR[1:0]	Name	Description
0x0	NONE	No action
0x1	TXFIFO	Clear TX FIFO

FIFOCLR (continued)

FIFOCLR[1:0]	Name	Description
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 CTRLB.RXEN when the USART is disabled will set CTRLB.RXEN immediately. When the USART is enabled, CTRLB.RXEN will be cleared, and SYNCBUSY.CTRLB will be set and remain set until the receiver is enabled. When the receiver is enabled, CTRLB.RXEN will read back as '1'.
Writing '1' to CTRLB.RXEN when the USART is enabled will set SYNCBUSY.CTRLB, which will remain set until the receiver is enabled, and CTRLB.RXEN will read back as '1'.
This bit is not enable-protected.

Value	Description
0	The receiver is disabled or being enabled.
1	The receiver is enabled or 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 CTRLB.TXEN when the USART is disabled will set CTRLB.TXEN immediately. When the USART is enabled, CTRLB.TXEN will be cleared, and SYNCBUSY.CTRLB will be set and remain set until the transmitter is enabled. When the transmitter is enabled, CTRLB.TXEN will read back as '1'.
Writing '1' to CTRLB.TXEN when the USART is enabled will set SYNCBUSY.CTRLB, which will remain set until the transmitter is enabled, and CTRLB.TXEN will read back as '1'.
This bit is not enable-protected.

Value	Description
0	The transmitter is disabled or being enabled.
1	The transmitter is enabled or 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 0x1 or 0x5). 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.
This bit is not synchronized.

Value	Description
0	Even parity.
1	Odd parity.

Bit 10 – ENC Encoding Format

This bit selects the data encoding format.
This bit is not synchronized.

Value	Description
0	Data is not encoded.
1	Data is IrDA encoded.

Bit 9 – SFDE Start of Frame Detection Enable

This bit controls whether the start-of-frame detector will wake up the device when a start bit is detected on the RxD line.
This bit is not synchronized.

SFDE	INTENSET.RXS	INTENSET.RXC	Description
0	X	X	Start-of-frame detection disabled.
1	0	0	Reserved
1	0	1	Start-of-frame detection enabled. RXC wakes up the device from all sleep modes.
1	1	0	Start-of-frame detection enabled. RXS wakes up the device from all sleep modes.
1	1	1	Start-of-frame detection enabled. Both RXC and RXS wake up the device from all sleep modes.

Bit 8 – COLDEN Collision Detection Enable

This bit enables collision detection.
This bit is not synchronized.

Value	Description
0	Collision detection is not enabled.
1	Collision detection is enabled.

Bit 6 – SBMODE Stop Bit Mode

This bit selects the number of stop bits transmitted.
This bit is not synchronized.

Value	Description
0	One stop bit.
1	Two stop bits.

Bits 2:0 – CHSIZE[2:0] Character Size

These bits select the number of bits in a character.
These bits are not synchronized.

CHSIZE[2:0]	Description
0x0	8 bits
0x1	9 bits
0x2-0x4	Reserved
0x5	5 bits
0x6	6 bits
0x7	7 bits

35.6.5.3. Control C

Name: CTRLC
Offset: 0x08
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Note: Care should be taken when applying protocol specific settings to this register. Not all combinations of settings are compliant with standards.

Table 35-9. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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.

TXTRHOLD	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.

RXTRHOLD	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.
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.FORM=0x7 and CTRLA.CMODE=1).

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 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.

35.6.5.4. Baud

Name: BAUD
Offset: 0x0C
Reset: 0x0000
Property: Enable-Protected, PAC Write-Protection

Table 35-10. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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 Clock Generation – Baud-Rate Generator section.

- **Bits 12:0 - BAUD[12:0]: Baud Value**

  These bits control the clock generation, as described in the Clock Generation – Baud-Rate Generator section.

35.6.5.5. Receive Pulse Length Register

Name: RXPL
Offset: 0x0E
Reset: 0x00
Property: PAC Write-Protection

Table 35-11. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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}:

$$PULSE \geq \left(\mathrm{RXPL} + 1\right) \cdot SE_{per}$$

35.6.5.6. 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). On read, a bit value of zero indicates the associated interrupt is disabled while a bit value of one indicates the associated interrupt is enabled.

Table 35-12. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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 Disable
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.

Bit 5 – RXBRK Receive Break Interrupt Disable
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.

Bit 4 – CTSIC Clear to Send Input Change Interrupt Disable
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.

Bit 3 – RXS Receive Start Interrupt Disable
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.

Bit 2 – RXC Receive Complete Interrupt Disable
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.

Bit 1 – TXC Transmit Complete Interrupt Disable
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.

Bit 0 – DRE Data Register Empty Interrupt Disable
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.

35.6.5.7. Interrupt Enable Set

Name: INTENSET
Offset: 0x16
Reset: 0x00
Property: PAC Write-Protection

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear register (INTENCLR). On read, a bit value of zero indicates the associated interrupt is disabled while a bit value of one indicates the associated interrupt is enabled.

Table 35-13. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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.

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.

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.

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.

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.

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.

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.

35.6.5.8. Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x18
Reset: 0x00
Property: -

Note: Interrupt flags must be cleared and then read back to confirm the clear before exiting the ISR to avoid double interrupts.

Table 35-14. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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.

35.6.5.9. Status

Name: STATUS
Offset: 0x1A
Reset: 0x0000
Property: -

Table 35-15. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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.

Value	Description
0	Max. # of iterations not reached
1	Max # of iterations reached

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.

Value	Description
0	Transmission not complete, transmitter not empty
1	Transmission Complete, transmitter empty

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.

Value	Description
0	No collisions detected
1	Collision detected

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 = 0x4 or 0x5) 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.

Value	Description
0	No inconsistent sync fields
1	Inconsistent sync field received

Bit 3 – CTS Clear to Send

This bit indicates the current level of the CTS pin when flow control is enabled (CTRLA.TXPO).

Value	Description
0	CTS low
1	CTS high

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.

Value	Description
0	No buffer overflow
1	Buffer Overflow

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.

Value	Description
0	No Frame Errors
1	Frame Error received

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.

Value	Description
0	No Parity Errors
1	Parity Error detected

35.6.5.10. Synchronization Busy

Name: SYNCBUSY
Offset: 0x1C
Reset: 0x00000000
Property: -

Table 35-16. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
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, a bus 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 set 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, a bus 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.

35.6.5.11. Receive Error Count

Name: RXERRCNT
Offset: 0x20
Reset: 0x00
Property: Read-Synchronized

Write to this register only when SYNCBUSY.RXERRCNT = 0, otherwise a bus error will result.

Table 35-17. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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 in ISO7816 mode (CTRLA.FORM=0x7).
This register is automatically cleared on read.

35.6.5.12. Length

Name: LENGTH
Offset: 0x22
Reset: 0x00
Property: Write-Synchronized

Write to this register only when SYNCBUSY.LENGTH = 0, otherwise a bus error will result.

Table 35-18. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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

35.6.5.13. Data

Name: DATA
Offset: 0x28
Reset: 0x0000
Property: -

Table 35-19. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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.

35.6.5.14. Debug Control

Name: DBGCTRL
Offset: 0x30
Reset: 0x00
Property: PAC Write-Protection

Table 35-20. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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.

35.6.5.15. FIFO Space

Name: FIFOSPACE
Offset: 0x34
Reset: 0x0000

This register allows the user to identify the number of bytes present in each TX and RX FIFO.

Table 35-21. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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).

35.6.5.16. FIFO CPU Pointers

Name: FIFOPTR
Offset: 0x36
Reset: 0x0000

This register provides a copy of internal CPU TX and RX FIFO pointers.

Table 35-22. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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 Pointer
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 Pointer
These bits return the CPUWRPTR pointer value. These bits can be written only if the SERCOM is halted during debugging. When writing to DATA register, the DATA will be written to TXFIFO[CPUWRPTR] location.

35.7. SERCOM Serial Peripheral Interface (SPI)

35.7.1. 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 the Block Diagram. Each side, host and client, depicts a separate SPI containing a Shift register, a transmit buffer and a two-level receive buffer. In addition, the SPI host uses the SERCOM baud-rate generator, while the SPI client can use the SERCOM address match logic. Labels in capital letters are synchronous to CLK_SERCOMx_APB and accessible by the CPU, while labels in lowercase letters are synchronous to the SCK clock.

35.7.2. Features

SERCOM SPI includes the following features:

- Full-duplex, four-wire interface (MISO, MOSI, SCK, SS)
- One-level transmit buffer, two-level receive buffer
- Supports all four SPI modes of operation
- Single data direction operation allows alternate function on MISO or MOSI pin

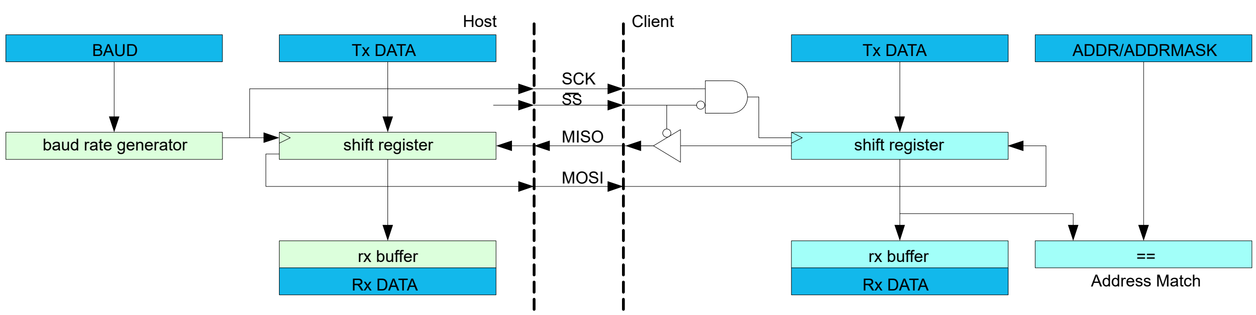
- Selectable LSB- or MSB-first data transfer
- Can be used with DMA
- 32-bit Extension for better system bus utilization
- Framed SPI protocol support in both Host and Client operating mode, with hardware controlled FSYNC
- Up to 16-bytes internal FIFO
- Host operation:
  - Serial clock speed, fSCK=1/tSCK[(1)]
  - 8-bit clock generator
  - Hardware controlled SS
  - Optional inter-character spacing
- Client Operation:
  - Serial clock speed, fSCK=1/tSSCK[(1)]
  - Optional 8-bit address match operation
  - Operation in all sleep modes
  - Wake on SS transition

Note:

1. For tSCK and tSSCK values, refer to SPI Timing Characteristics.

35.7.3. Block Diagram

Figure 35-31. Full-Duplex SPI Host Client Interconnection



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. Both PORT control bits PINCFGn.PULLEN and PINCFGn.DRVSTR are still effective. If the receiver is disabled, the data input pin can be used for other purposes. In host mode, the client select line ($\overline{SS}$) is hardware controlled when the Host Client Select Enable bit in the Control B register (CTRLB.MSSEN) is '1'.

Table 35-23. SPI Pin Configuration

Pin	Host SPI	Client SPI
MOSI	Output	Input
MISO	Input	Output
SCK	Output	Input
	Output (CTRLB.MSSEN=1) and (CTRLC.FRMEN = 0)	Input (CTRLC.FRMEN = 0)
$\overline{SS}$ / FSYNC	Output (CTRLC.FRMEN = 1) and (CTRLC.FMODE = 0)	Output (CTRLC.FRMEN = 1) and (CTRLC.FMODE = 0)
	Input (CTRLC.FRMEN = 1) and (CTRLC.FMODE = 1)	Input (CTRLC.FRMEN = 1) and (CTRLC.FMODE = 1)

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.

35.7.4. Functional Description

35.7.4.1. Principle of Operation

The SPI is a high-speed synchronous data transfer interface. It allows high-speed communication between the device and peripheral devices.

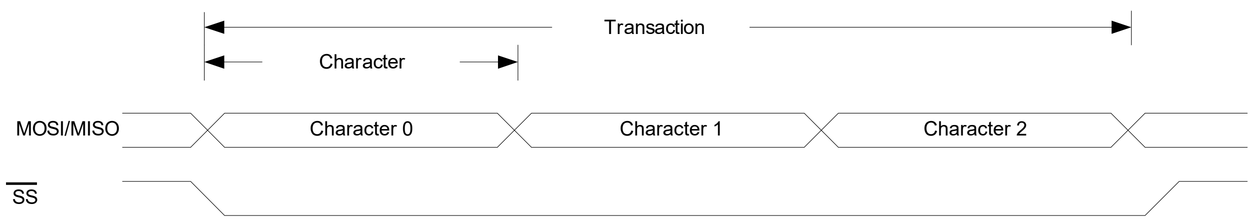
The SPI can operate as host or client. As host, the SPI initiates and controls all data transactions. The SPI is single buffered for transmitting and double buffered for receiving.

When transmitting data, the Data register can be loaded with the next character to be transmitted during the current transmission.

When receiving, the data is transferred to the two-level receive buffer, and the receiver is ready for a new character.

The SPI transaction format is shown in SPI Transaction Format. Each transaction can contain one or more characters. The character size is configurable, and can be either 8 or 9 bits.

Figure 35-32. SPI Transaction Format



The SPI host must pull the client 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 a 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.

35.7.4.2. Basic Operation

35.7.4.2.1. Initialization

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 Client 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).

35.7.4.2.2. 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.

35.7.4.2.3. Transfer Modes

There are four combinations of SCK phase and polarity to transfer serial data. The SPI Data Transfer modes are shown in the SPI Transfer Modes table, and the SPI Transfer Modes figure.

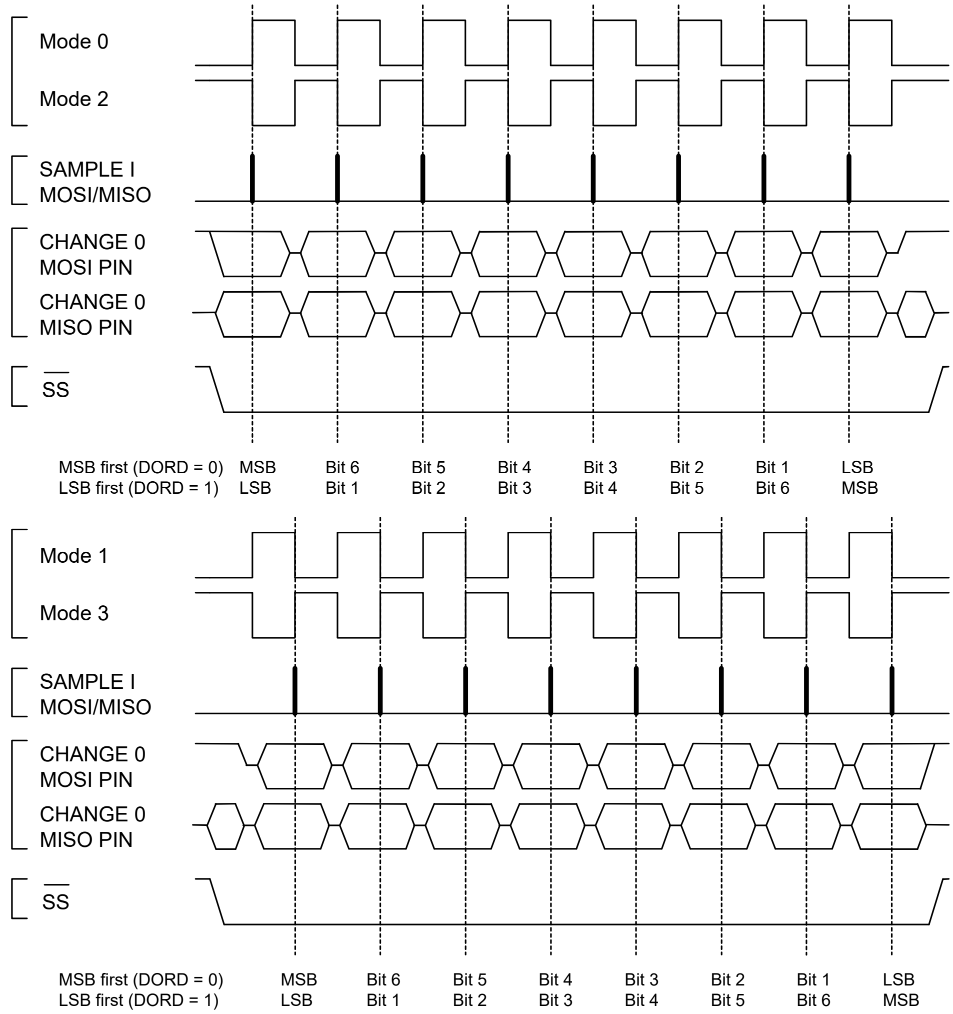
SCK phase is configured by the Clock Phase bit in the CTRLA register (CTRLA.CPHA). SCK polarity is programmed by the Clock Polarity bit in the CTRLA register (CTRLA.CPOL). Data bits are shifted out and latched in on opposite edges of the SCK signal. This ensures sufficient time for the data signals to stabilize.

Table 35-24. 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 35-33. SPI Transfer Modes



35.7.4.2.4. Transferring Data

Host

In Host mode (CTRLA.MODE=0x3), when Host Client Enable Select (CTRLB.MSSEN) is ‘1’, hardware will control the $\overline{\text{SS}}$ line.

When Host Client Select Enable (CTRLB.MSSEN) is '0', the $\overline{\text{SS}}$ line must be configured as an output. $\overline{\text{SS}}$ can be assigned to any general purpose I/O pin. When the SPI is ready for a data transaction, software must pull the $\overline{\text{SS}}$ 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.

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.

When the transaction is finished, the host must pull the $\overline{SS}$ line high to notify the client. If Host Client Select Enable (CTRLB.MSSEN) is set to '0', the software must pull the $\overline{SS}$ line high.

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 DATA at any time as long as the Data Register Empty flag in the Interrupt Status and Clear register (INTFLAG.DRE) is set.

When $\overline{SS}$ is pulled low and SCK is running, the client will sample and shift out data according to the Transaction mode set. When the content of TxDATA has been loaded into the Shift register, INTFLAG.DRE will be set, and new data can be written to DATA.

Similar to the host, the client will receive one character for each character transmitted. A character will be transferred into the two-level 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.

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 of 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.

Once the DATA register is empty, it takes three CLK_SERCOM_APB cycles for INTFLAG.DRE to be set.

35.7.4.2.5. 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):

1. 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.
2. 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.

35.7.4.3. Additional Features

35.7.4.3.1. 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.

35.7.4.3.2. 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.

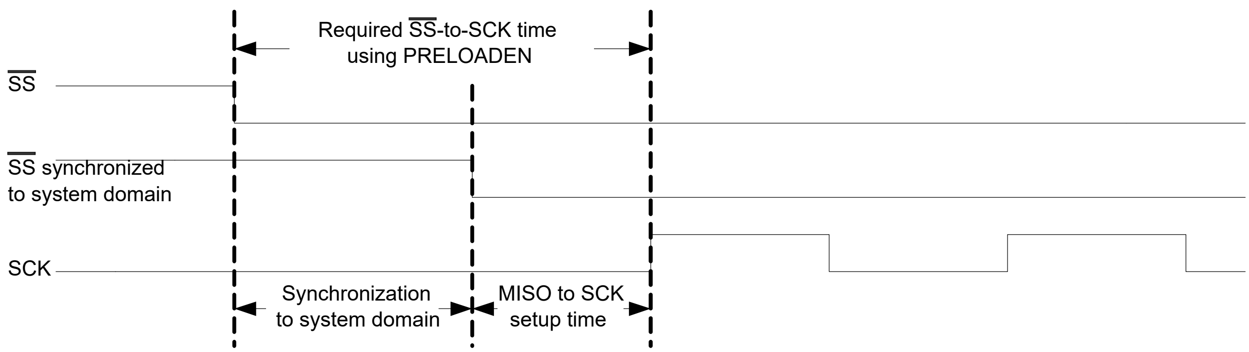
Preloading can be used to preload data into the shift register while $\overline{SS}$ is high: this eliminates sending a dummy character when starting a transaction. If the shift register is not preloaded, the current contents of the shift register will be shifted out.

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. See also the *Electrical Characteristics* chapters for timing details.

Preloading is enabled by writing '1' to the Client Data Preload Enable bit in the CTRLB register (CTRLB.PLOADEN).

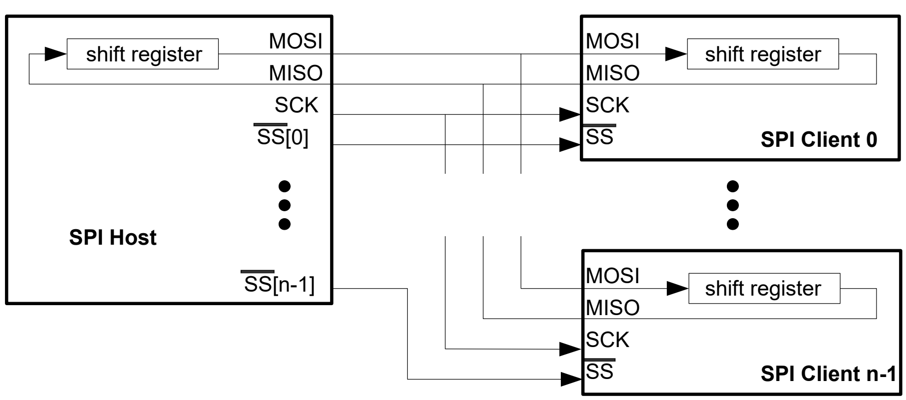
Figure 35-34. Timing Using Preloading



35.7.4.3.3. Host with Several Clients

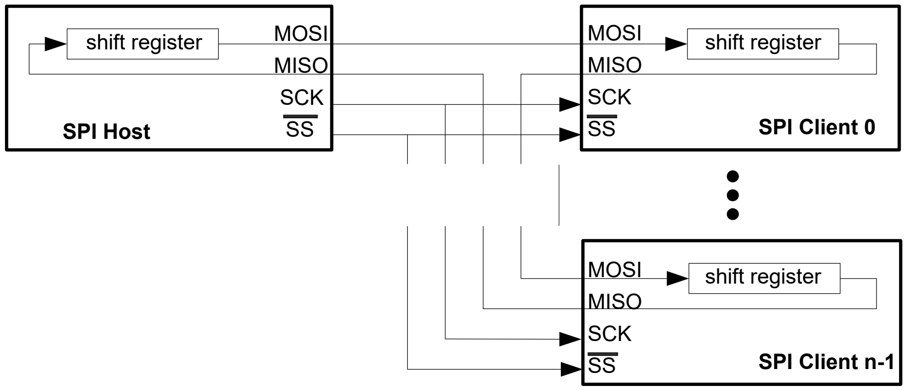
Host with multiple clients in parallel is only available when Host Client Select Enable (CTRLB.MSSEN) is set to zero and hardware $\overline{SS}$ control is disabled. If the bus consists of several SPI clients, a SPI host can use general purpose I/O pins to control the $\overline{SS}$ line to each of the clients on the bus, as shown in the following figure. In this configuration, the single selected SPI client will drive the tri-state MISO line.

Figure 35-35. 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 $\overline{\text{SS}}$ line is provided to all clients, enabling them simultaneously. The host must shift n characters for a complete transaction. Depending on the Host Client Select Enable bit (CTRLB.MSSEN), the $\overline{\text{SS}}$ line can be controlled either by hardware or user software and normal GPIO.

Figure 35-36. Multiple Clients in Series



35.7.4.3.4. Loopback Mode

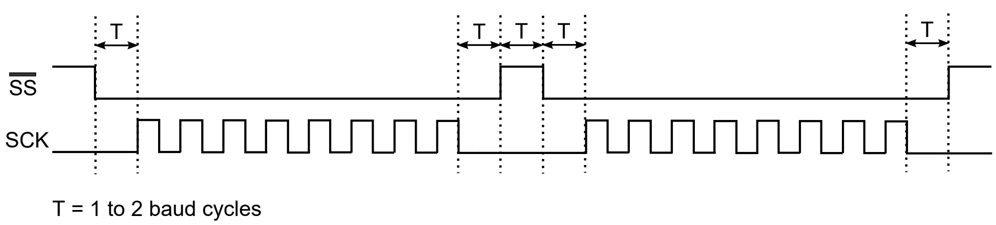
For Loop-back mode, configure the Data In Pinout (CTRLA.DIPO) and Data Out Pinout (CTRLA.DOPO) to use the same data pins for transmit and receive. The loop-back is through the pad, so the signal is also available externally.

35.7.4.3.5. Hardware Controlled $\overline{\text{SS}}$

In Host mode, a single $\overline{\text{SS}}$ chip select can be controlled by hardware by writing the Host Client Select Enable (CTRLB.MSSEN) bit to '1'. In this mode, the $\overline{\text{SS}}$ 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 $\overline{\text{SS}}$ pin will always be driven low between frames.

In Hardware Controlled $\overline{\text{SS}}$, the time T is between one and two baud cycles depending on the SPI Transfer mode.

Figure 35-37. Hardware Controlled SS



When CTRLB.MSSEN=0, the $\overline{\text{SS}}$ pin(s) is/are controlled by user software and normal GPIO.

35.7.4.3.6. Client Select Low Detection

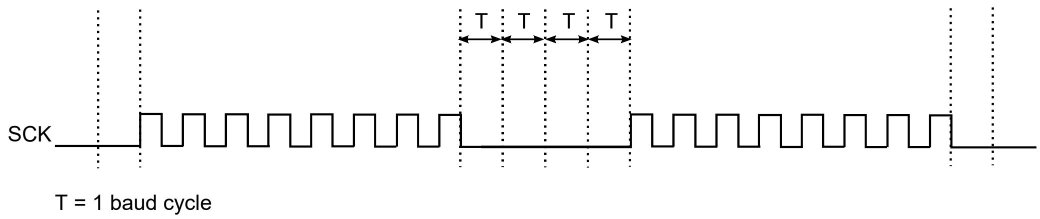
In Client mode, the SPI can wake the CPU when the client select ($\overline{\text{SS}}$) goes low. When the Client Select Low Detect is enabled (CTRLB.SSDE=1), a high-to-low transition will set the Client Select Low Interrupt flag (INTFLAG.SSL) and the device will wake-up if applicable.

35.7.4.3.7. 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 35-38. Four Cycle Inter-Character Spacing Example



35.7.4.3.8. 32-bit Extension

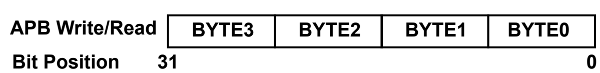
For better system bus utilization, 32-bit data receive and transmit can be enabled by writing to the Data 32-bit bit field in the Control C register (CTRLC.DATA32B=1). When enabled, write and read transaction to/from the DATA register are 32 bit in size.

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 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.

Only 8-bit character size is supported.

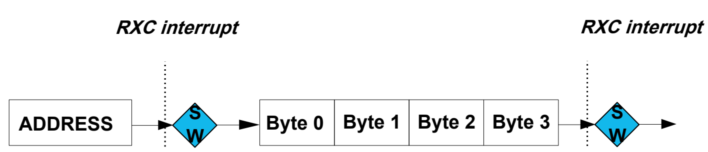
Figure 35-39. 32-bit Extension Byte Ordering



32-bit Extension Client Operation

The following figure 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 35-40. 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.

35.7.4.3.9. Framed SPI Operation

The SPI supports Framed SPI protocol while operating in either Host or Client mode. The CTRLC.FRMEN bit enables Framed SPI modes and causes the SS pin to be used as a frame synchronization (FSYNC) input or output pin. The state of CTRLB.MSSEN bit is then ignored.

Unlike in normal SPI mode, the serial clock is continuous (free-running) in Framed SPI mode rather than being generated only when there is data to be transmitted. The data transmission/reception starts only when the frame synchronization is generated at the FSYNC pin. The device can be either a frame host if it generates the frame sync or a frame client if it receives the frame sync at the FSYNC pin. In other words, only a frame host can generate the frame synchronization pulse.

Frame Host or Client mode is selected by clearing or setting the CTRLC.FMODE bit, respectively. The frame synchronization can be an active-high or an active-low pulse based on the CTRLC.FSPOL. It can have either one SCK clock duration, or frame duration, based on the CTRLC.FSLEN settings.

Irrespective of which device is a host and which is a client, a framed SPI data transfer begins as soon as the frame host generates the frame sync and writes the data to DATA. For full-duplex operation, the frame client should write to its buffer before the frame host does, in order to ensure that the data is ready at both ends when the data transfer begins.

Frame Host

The frame host mode is enabled by setting CTRLC.FMODE = HOST and CTRLC.FRMEN = 1.

If the module is enabled in host operating mode (CTRLA.MODE = 0x03), the serial clock is output at the SCK pin, regardless of whether the module is transmitting, and the FSYNC pin is driven high on the next transmit edge of the SCK clock when the Data buffer is written. Data will start transmitting on the subsequent transmit edge of the SPI clock. As long as data is available in the transmit buffer, a new frame sync is initiated after completing a transmit/receive sequence.

If the module is enabled in client operating mode (CTRLA.MODE = 0x2), the input clock at the SCK pin is continuous while the FSYNC pin and transmission are generated in the same way as in host operating mode.

If the 32-bit extension is disabled (CTRLC.DATA32B = 0), the transmit will continue as long as data are present in the internal buffer. The Transmit Complete interrupt flag is set when there are no more data to be transferred, and when the internal shift register is empty. The Data Register Empty flag is set each time a new data can be stored in the internal buffer.

If the 32-bit extension is enabled (CTRLC.DATA32B = 1), and the DATA Length Counter is disabled (LENTH.LENEN = 0), the behavior is similar to the previous mode, with the exception the DATA must be 32-bit format (word).

When the DATA Length Counter is enabled (LENTH.LENEN = 1), the length of the frame is controlled by the DATA Length field in LENGTH register (LENGTH.LEN). The Transmit Complete interrupt flag is set when there are no more data to be transferred, and when the internal shift register is empty. The Data Register Empty flag is set each time a new 32-bit data can be stored in the internal buffer.

A DMA trigger is generated each time there is 32-bit internal place to store a new data. As a consequence, the DMA descriptor transfer counter must be 32-bit aligned.

Figure 35-41. FMODE = HOST, FSPOL = HIGH, FSLEN = STROBE, FSES = 0

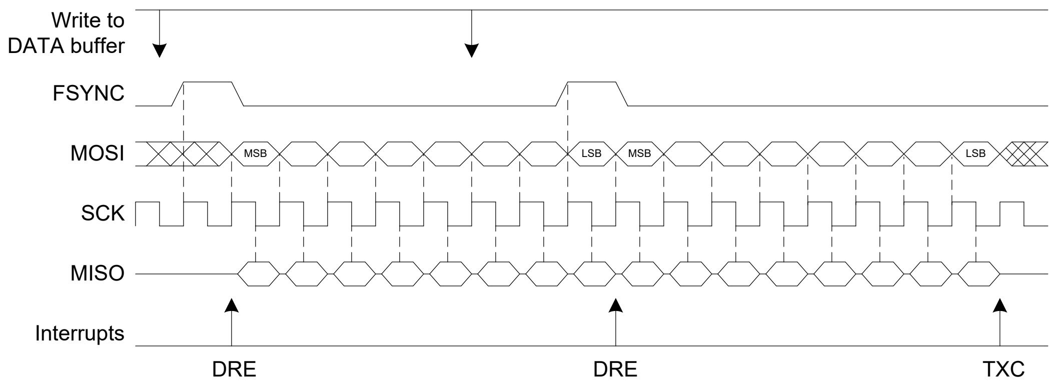


Figure 35-42. FMODE = HOST, FSPOL = HIGH, FSLEN = LEVEL, FSES = 0

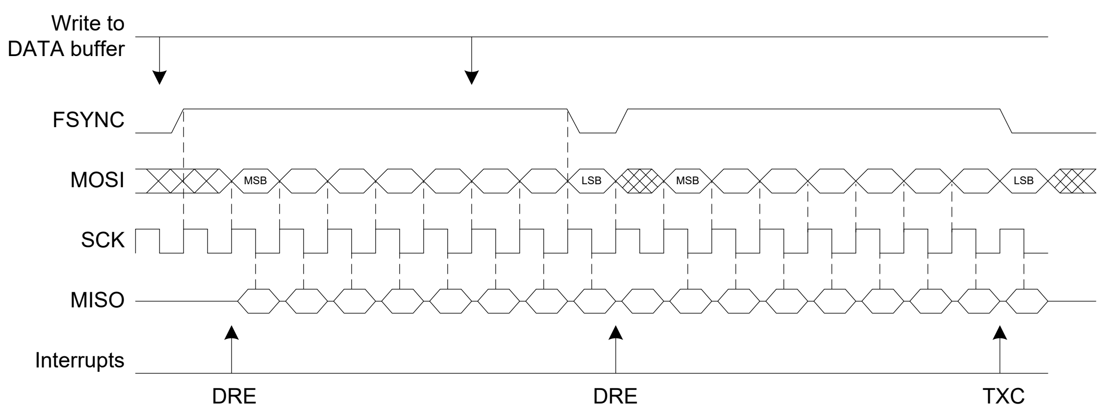


Figure 35-43. FMODE = HOST, FSPOL = HIGH, FSLEN = STROBE, FSES = 1

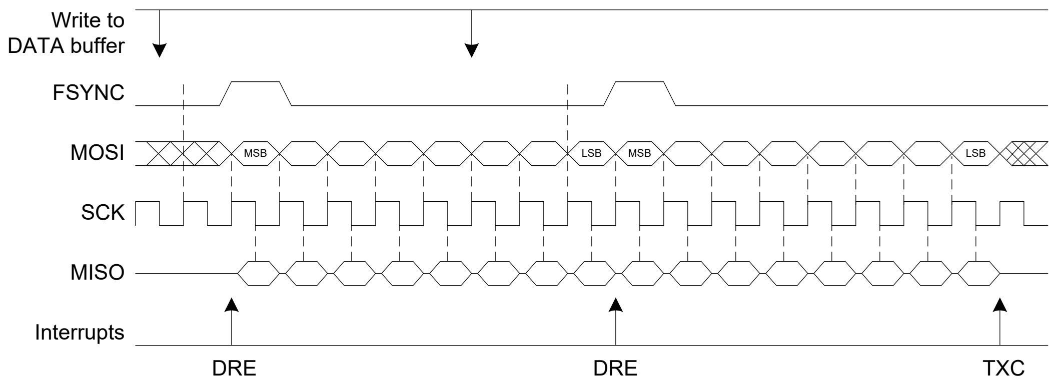
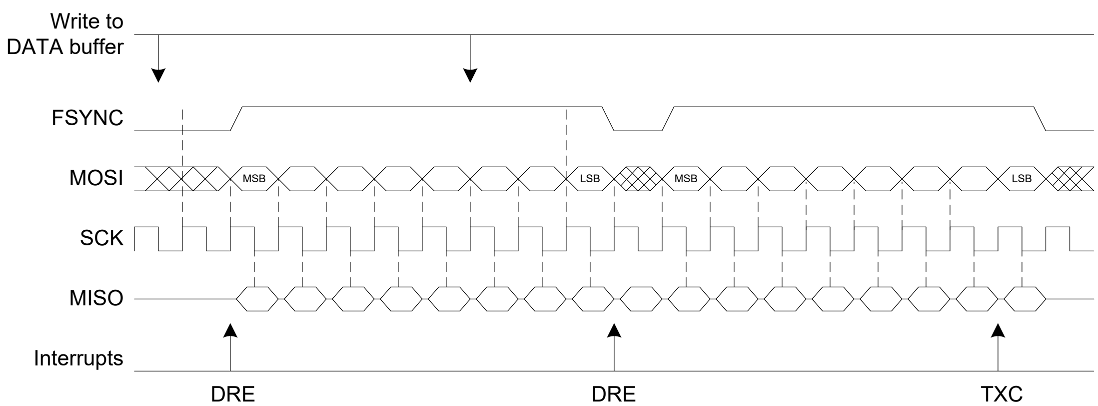


Figure 35-44. FMODE = HOST, FSPOL = HIGH, FSLEN = LEVEL, FSES = 1



Frame Client

The frame client mode is enabled by setting CTRLC.FMODE = CLIENT and CTRLC.FRMEN = 1. In both host (CTRLA.MODE = 0x03) and client (CTRLA.MODE = 0x02) operating mode, the FSYNC pin is an input. The serial clock is output at the SCK pin in host operating mode, and an input in client operating mode.

When the FSYNC pin is sampled high or low (CTRLC.FSPOL = 1/0), the data is transmitted on the subsequent transmit edge of the SPI clock. The user must make sure that the correct to-be-transmitted data is loaded into DATA buffer before the frame sync pulse is received at the FSYNC pin. If the FSYNC is detected and there is no data available in the buffer, the module will transmit zeroes until the end of the frame.

If the 32-bit extension is disabled (CTRLC.DATA32B = 0), one byte DATA will be sent, and a new FSYNC must be received before enabling a new transmission. The Transmit Complete interrupt flag is set when the internal shift register is empty. The Data Register Empty flag is set each time a new data can be stored in the internal buffer.

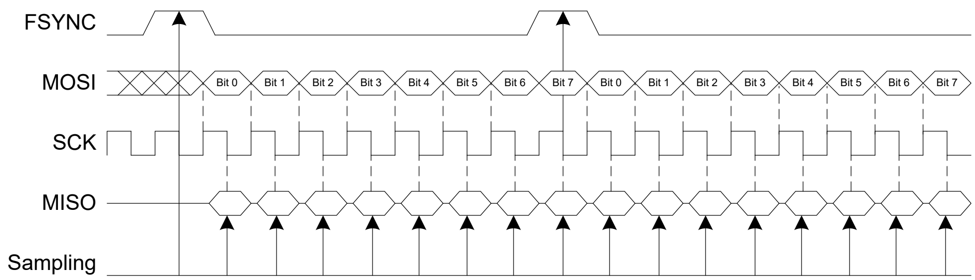
If the 32-bit extension is enabled (CTRLC.DATA32B = 1), and the DATA Length Counter is disabled (LENTH.LENEN = 0), the behavior is similar to the previous mode, with the exception the DATA will be 32-bit format (word).

When the DATA Length Counter is enabled (LENTH.LENEN = 1), the length of the frame is controlled by the DATA Length field in LENGTH register (LENGTH.LEN). The Transmit Complete interrupt flag is set when LENGTH DATA are transferred, and when the internal shift register is empty. The Data Register Empty flag is set each time a new 32-bit data can be stored in the internal buffer. A new FSYNC must be received before a new transaction starts.

If the address matching is enabled, the first data after FSYNC detection will be compared with the ADDR register. If a match occurs, the client continues to receive data. If the match fails, the data is discarded and the client waits for a new FSYNC.

A DMA trigger is generated each time there is 32-bit internal place to store a new data. As a consequence, the DMA descriptor transfer counter must be 32-bit aligned.

Figure 35-45. FMODE = CLIENT, FSPOL = HIGH, FSLEN = STROBE, FSES = 0



Framed SPI Errors

If the transmit buffer is empty when the SPI must load the shift register to start (in the case of Framed Client) or continue (for either Framed Host or Client) a transfer, the SPI immediately sets the transmit underrun status bit in the STATUS (STATUS.TUR) register. If enabled, an Error interrupt flag (INTFLAG.ERROR) will be also generated.

While the SPI is in an underrun condition, the SPI transmits all zeroes until the end of the transaction (byte, word or length counter). If the SPI is a frame client, another frame sync pulse can occur before the condition clears. In this case the SPI continues to transmit zeros. If the SPI is a frame host, it then waits for software to clear the under-run condition before initiating another FSYNC regardless of the state of the DATA buffer.

When the TUR status is cleared the data present in the DATA buffer is flushed. The SPI ignores writes to the DATA after clearing the STATUS.TUR bit, until a read of STATUS register, when STATUS.TUR=0. This behavior ensures that a data service routine that is interrupted long enough to cause STATUS.TUR, cannot inadvertently start a new framed transaction after the SPI error handler has cleared the error.

For cases when software does not care or need to know about the underrun condition, CTRLC.IGNTUR = 1 provides the serial engine the ability to ignore the underrun. When an underrun occurs, the SPI still sets the corresponding STATUS and INTFLAG bits. When the SPI is either a frame client or a frame host, an underrun event still causes the SPI to transmit zeros until the end of the frame. However, when the Ignore Transmit Underrun bit is set (CTRLC.IGNTUR = 1), the SPI can re-sample the underrun condition and continue to transmit data at each frame boundary. If the SPI is a frame host, new data written to the DATA buffer during a frame when an underrun condition exists does not get transmitted during that frame. But, the SPI evaluates the DATA buffer continuously after the last frame. If data is in it, the SPI generates a frame sync and transmits the data. If the SPI is a frame client, its transmit logic evaluates DATA buffer for underrun during the next FSYNC. If the buffer contains data when a valid FSYNC is detected, SPI transmits that data. If not, the SPI transmits zero data until the end of the frame.

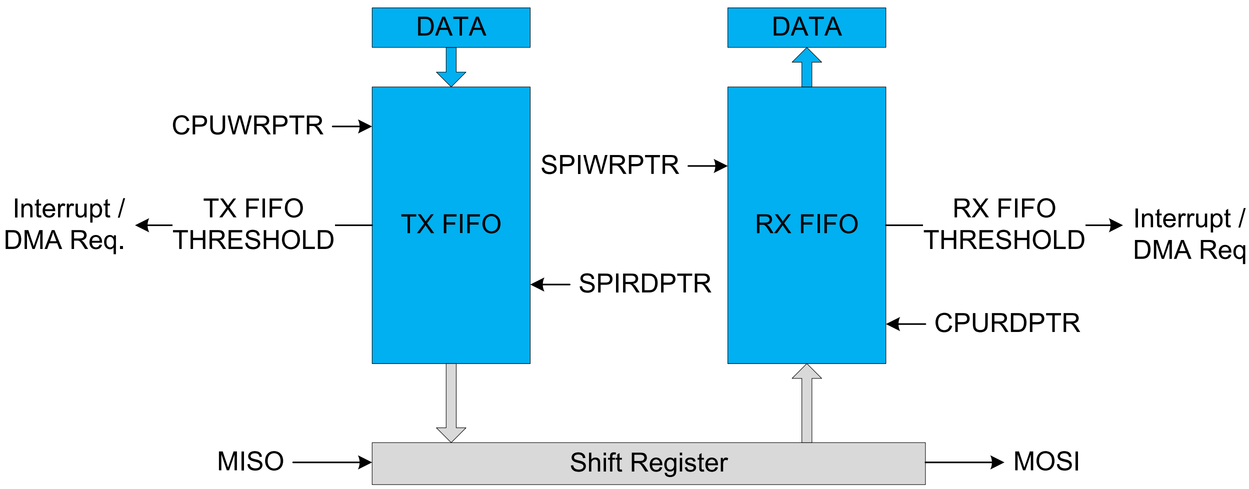
In the case of client frames, if a new FSYNC is received during an ongoing frame transaction, the transaction length error status bit is set, and the error interrupt flag is set. The module will continue to send data from the internal buffer.

35.7.4.3.10. 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 FIFO 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 go to TX FIFO. Reads to DATA register by the CPU will come from the 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 35-46. 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 respectively, 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 respectively, 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 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.

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.

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.

35.7.4.4. DMA and Interrupts

Table 35-25. Module Request for SERCOM SPI

Condition	Request DMA	Interrupt
Standard (DRE): Data Register Empty FIFO (DRE): at least TXTRHOLD locations in TX FIFO are empty	Yes (request cleared when data is written)	Yes
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	N/A	Yes
Client Select low (SSL)	N/A	Yes
Error (ERROR)	N/A	Yes

35.7.4.4.1. 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.

35.7.4.4.2. 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)
- Client 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 the corresponding interrupt is enabled. The interrupt request remains active until 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 the Nested Vector Interrupt Controller for details.

35.7.4.5. Sleep Mode Operation

The behavior in Sleep mode depends on the host/client configuration and the Run In Standby bit in the Control A register (CTRLA.RUNSTDBY):

- Host operation, CTRLA.RUNSTDBY=1: The peripheral clock GCLK_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_SERCOM_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

35.7.4.6. Synchronization

Due to asynchronicity between the main clock domain and the peripheral clock domains, some SERCOM registers need to be synchronized when written ("Write-Synchronized") or read ("Read-Synchronized").

The following bits are synchronized when written:

- Software Reset bit in the CTRLA register (CTRLA.SWRST)
- Enable bit in the CTRLA register (CTRLA.ENABLE)
- Receiver Enable bit in the CTRLB register (CTRLB.RXEN)
- Data Length Enable bit and Data Length bits in the LENGTH register (LENGTH.LENEN, LENGTH.LEN)

Note: CTRLB.RXEN is write-synchronized somewhat differently than other registers. See the CTRLB register for details.

Required write synchronization is denoted by the "Write-Synchronized" property in the register description. If a write-synchronized register is written while a synchronization is ongoing, a Bus Error exception will be generated.

35.7.5. Register Summary

For descriptions and definitions of both Register and bitfield properties, refer to Register Properties.

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							FMODE	FRMEN
		15:8					IGNTUR	FSPOL	FSLEN	FSES
		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					TUR	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]			

35.7.5.1. Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Table 35-26. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
		DORD	CPOL	CPHA	FORM[3:0]			
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

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

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

Bit	7	6	5	4	3	2	1	0
	RUNSTDBY			MODE[2:0]			ENABLE	SWRST
Access	R/W			R/W	R/W	R/W	R/W	R/W
Reset	0			0	0	0	0	0

Bit 30 – DORD Data Order

This bit selects the data order when a character is shifted out from the Shift register.
This bit is not synchronized.

Value	Description
0	MSB is transferred first.
1	LSB is transferred first.

Bit 29 – CPOL Clock Polarity

In combination with the Clock Phase bit (CPHA), this bit determines the SPI Transfer mode.
This bit is not synchronized.

Value	Description
0	SCK is low when idle. The leading edge of a clock cycle is a rising edge, while the trailing edge is a falling edge.
1	SCK is high when idle. The leading edge of a clock cycle is a falling edge, while the trailing edge is a rising edge.

Bit 28 – CPHA Clock Phase

In combination with the Clock Polarity bit (CPOL), this bit determines the SPI Transfer mode.
This bit is not synchronized.

Mode	CPOL	CPHA	Leading Edge	Trailing Edge
0x0	0	0	Rising, sample	Falling, change

CPHA (continued)

Mode	CPOL	CPHA	Leading Edge	Trailing Edge
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.
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.

FORM[3:0]	Name	Description
0x0	SPI	SPI frame
0x1	-	Reserved
0x2	SPI_ADDR	SPI frame with address
0x3-0xF	-	Reserved

Bits 21:20 – DIPO[1:0] Data In Pinout

These bits define the Data In (DI) pad configurations.
In host operation, DI is MISO.
In client operation, DI is MOSI.
These bits are not synchronized.

DIPO[1:0]	Name	Description
0x0	PAD[0]	SERCOM PAD[0] is used as data input
0x1	PAD[1]	SERCOM PAD[1] is used as data input
0x2	PAD[2]	SERCOM PAD[2] is used as data input
0x3	PAD[3]	SERCOM PAD[3] is used as data input

Bits 17:16 – DOPO[1:0] Data Out Pinout

This bit defines the available pad configurations for Data Out (DO) and the Serial Clock (SCK). In client operation, the Client Select ($\overline{SS}$) line is controlled by DOPO, while in host operation the $\overline{SS}$ line is controlled by the port configuration.
In host operation, DO is MOSI.
In client operation, DO is MISO.
These bits are not synchronized.

DOPO	DO	SCK	Client $\overline{SS}$	Host $\overline{SS}$
0x0	PAD[0]	PAD[1]	PAD[2]	PAD[2] Host $\overline{SS}$ pin when MSSEN = 1 otherwise System configuration
0x1	Reserved			
0x2	PAD[3]	PAD[1]	PAD[2]	PAD[2] Host $\overline{SS}$ pin when MSSEN = 1 otherwise System configuration
0x3	Reserved			

Bit 8 – IBON Immediate Buffer Overflow Notification

This bit controls when the Buffer Overflow Status bit (STATUS.BUFOVF) is set when a buffer overflow occurs.
This bit is not synchronized.

Value	Description
0	STATUS.BUFOVF is set when it occurs in the data stream.
1	STATUS.BUFOVF is set immediately upon buffer overflow.

Bit 7 – RUNSTDBY Run In Standby

This bit defines the functionality in Standby Sleep mode.

This bit is not synchronized.

RUNSTDBY	Client	Host
0x0	Disabled. All reception is dropped, including the ongoing transaction.	Generic clock is disabled when ongoing transaction is finished. All interrupts can wake-up the device.
0x1	Ongoing transaction continues, wake on Receive Complete interrupt.	Generic clock is enabled while in sleep modes. All interrupts can wake-up the device.

Bits 4:2 – MODE[2:0] Operating Mode

These bits must be written to 0x2 or 0x3 to select the SPI of the SERCOM.
0x2: SPI client operation
0x3: SPI host operation
These bits are not synchronized.

Bit 1 – ENABLE Enable

Due to synchronization, there is delay from writing CTRLA.ENABLE until the peripheral is enabled/disabled. The value written to CTRL.ENABLE will read back immediately and the Synchronization Enable Busy bit in the Synchronization Busy register (SYNCBUSY.ENABLE) will be set. SYNCBUSY.ENABLE is cleared when the operation is complete.
This bit is not enable-protected.

Value	Description
0	The peripheral is disabled or being disabled.
1	The peripheral is enabled or being enabled.

Bit 0 – SWRST Software Reset

Writing '0' to this bit has no effect.
Writing '1' to this bit resets all registers in the SERCOM, except DBGCTRL, to their initial state, and the SERCOM 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 a bus error. Reading any register will return the Reset value of the register.
Due to synchronization, there is a delay from writing CTRLA.SWRST until the Reset is complete. CTRLA.SWRST and SYNCBUSY. SWRST will both be cleared when the Reset is complete.
This bit is not enable-protected.

Notes:

1. When the CTRLA.SWRST is written, the user should poll the SYNCB.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 the hardware.

Value	Description
0	There is no Reset operation ongoing.
1	The Reset operation is ongoing.

35.7.5.2. Control B

Name: CTRLB
Offset: 0x04
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected, Write-Synchronized

Write to this register only when SYNCBUSY.CTRLB = 0, otherwise a bus error will result.

Table 35-27. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

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

Bit	15	14	13	12	11	10	9	8
	AMODE[1:0]		MSSEN				SSDE	
Access	R/W	R/W	R/W				R/W	
Reset	0	0	0				0	

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

Bits 23:22 – FIFOCLR[1:0] FIFO Clear
When these bits are set, the corresponding FIFO will be cleared. The bits will automatically clear when SYNCBUSY.CTRLB = 0.
These bits are not enable-protected.

FIFOCLR[1:0]	Name	Description
0x0	NONE	No action
0x1	TXFIFO	Clear TX FIFO
0x2	RXFIFO	Clear RX FIFO
0x3	BOTH	Clear both TX/RX FIFO

Bit 17 – RXEN Receiver Enable
Writing '0' to this bit 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 it immediately. When the SPI is enabled, RXEN will be cleared, SYNCBUSY.CTRLB will be set and remain set until the receiver is enabled. When the receiver is enabled RXEN will read back as '1'.
Writing '1' to RXEN when the SPI is enabled will set SYNCBUSY.CTRLB, which will remain set until the receiver is enabled, and RXEN will read back as '1'.

This bit is not enable-protected.

Value	Description
0	The receiver is disabled or being enabled.
1	The receiver is enabled or 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.

AMODE[1:0]	Name	Description
0x0	MASK	ADDRMASK is used as a mask to the ADDR register
0x1	2_ADDRS	The client responds to the two unique addresses in ADDR and ADDRMASK
0x2	RANGE	The client responds to the range of addresses : ADDRMASK < address < ADDR
0x3	-	Reserved

Bit 13 – MSSEN Host Client Select Enable

This bit enables hardware Client Select ($\overline{SS}$) control.

Value	Description
0	Hardware $\overline{SS}$ control is disabled.
1	Hardware $\overline{SS}$ control is enabled.

Bit 9 – SSDE Client Select Low Detect Enable

This bit enables wake-up when the Client Select ($\overline{SS}$) pin transitions from high to low.

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.

Bits 2:0 – CHSIZE[2:0] Character Size

CHSIZE[2:0]	Name	Description
0x0	8BIT	8 bits
0x1	9BIT	9 bits
0x2-0x7	-	Reserved

35.7.5.3. Control C

Name: CTRLC
Offset: 0x08
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Table 35-28. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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
							FMODE	FRMEN
Access							R/W	R/W
Reset							0	0

Bit	15	14	13	12	11	10	9	8
					IGNTUR	FSPOL	FSLEN	FSES
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

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.

TXTRHOLD	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.

RXTRHOLD	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 Enable 32-Bit Data

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
1	Transactions from and to DATA register are 32-bit

Bit 17 – FMODE Frame Mode

This bit defines the Frame Mode.

FMODE	Name	Description
0	HOST	Frame Host
1	CLIENT	Frame Client

Bit 16 – FRMEN Frame Mode Enable

This bit enables the SPI Frame Mode operation.

FRMEN	Name	Description
0	DISABLE	Frame Mode Disabled
1	ENABLE	Frame Mode Enabled

Bit 11 – IGNTUR Ignore Transmit Underrun

This bit controls when the underrun conditions in framed mode must be ignored.
This bit is not synchronized.

Value	Description
0	When a new FSYNC is detected, zero-bytes frames will be sent while underrun condition detection is not cleared.
1	When a new FSYNC is detected, DATA will be transmitted.

Bit 10 – FSPOL Frame Synch Polarity

This bit defines the valid Frame Synch polarity.

FSPOL	Name	Description
0	HIGH	VCC-level valid polarity
1	LOW	Ground-level valid polarity

Bit 9 – FSLEN Frame Synch Length

This bit defines the Frame Synch duration.

FSLEN	Name	Description
0	STROBE	One SCK pulse
1	LEVEL	One frame duration valid level

Bit 8 – FSES Frame Sync Edge Select

This bit controls when the frame sync pulse edge must be generated.
This bit is not synchronized.

Value	Description
0	Frame synchronization pulse (idle-to-active edge) precedes the first bit clock.
1	Frame synchronization pulse (idle-to-active edge) coincides with the first bit clock.

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

35.7.5.4. Baud Rate

Name: BAUD
Offset: 0x0C
Reset: 0x00
Property: PAC Write-Protection, Enable-Protected

Table 35-29. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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 Clock Generation – Baud-Rate Generator.

35.7.5.5. 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). On read, a bit value of zero indicates the associated interrupt is disabled while a bit value of one indicates the associated interrupt is enabled.

Table 35-30. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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 Disable
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.

Bit 3 – SSL Client Select Low Interrupt Disable
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the Client Select Low Interrupt Enable bit, which disables the Client Select Low interrupt.

Bit 2 – RXC Receive Complete Interrupt Disable
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.

Bit 1 – TXC Transmit Complete Interrupt Disable
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.

Bit 0 – DRE Data Register Empty Interrupt Disable
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.

35.7.5.6. 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). On read, a bit value of zero indicates the associated interrupt is disabled while a bit value of one indicates the associated interrupt is enabled.

Table 35-31. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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.

Bit 3 – SSL Client Select Low Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will set the Client Select Low Interrupt Enable bit, which enables the Client Select Low interrupt.

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.

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.

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.

35.7.5.7. Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x18
Reset: 0x00
Property: -

Note: Interrupt flags must be cleared and then read back to confirm the clear before exiting the ISR to avoid double interrupts.

Table 35-32. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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 flags in the STATUS register: TUR error, BUFOVF error, and the LENERR error.
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the flag.

Bit 3 – SSL Client 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 Client 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 (CTRLA.FORM = 0x2), 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 (CTRLA.FORM = 0x2), 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.

35.7.5.8. Status

Name: STATUS
Offset: 0x1A
Reset: 0x0000
Property: –

Table 35-33. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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

Bit	7	6	5	4	3	2	1	0
					TUR	BUFOVF		
Access					R/W	R/W		
Reset					0	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 3 – TUR Frame Transmit Underrun

This bit is cleared by writing '1' to the bit or by disabling the receiver. This bit is set when an underflow condition is detected in frame mode. Writing '0' to this bit has no effect. Writing '1' to this bit sends a request to clear it. It will be actually cleared after the DATA buffer is flushed.

Value	Description
0	No Underrun has occurred.
1	A Underrun 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 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.

35.7.5.9. Synchronization Busy

Name: SYNCBUSY
Offset: 0x1C
Reset: 0x00000000
Property: -

Table 35-34. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
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 asserted until synchronization is complete. If the LENGTH register is written to while SYNCBUSY.LENGTH is asserted, a bus 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, after which the bit will read zero. If CTRLB is written while SYNCBUSY.CTRLB=1, a bus 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, after which it will read zero.

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, after which it will read zero.

Value	Description
0	SWRST synchronization is not busy.
1	SWRST synchronization is busy.

35.7.5.10. Length

Name: LENGTH
Offset: 0x22
Reset: 0x0000
Property: Write-Synchronized

Write to this register only when SYNCBUSY.LENGTH = 0, otherwise a bus error will result.

Table 35-35. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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

35.7.5.11. Address

Name: ADDR
Offset: 0x24
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Table 35-36. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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 = 0x2, CTRLB.AMODE).

Bits 7:0 – ADDR[7:0] Address

These bits hold the address when the transaction format with address is used (CTRLA.FORM = 0x2, CTRLB.AMODE).

35.7.5.12. Data

Name: DATA
Offset: 0x28
Reset: 0x0000
Property: –

Table 35-37. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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.

35.7.5.13. Debug Control

Name: DBGCTRL
Offset: 0x30
Reset: 0x00
Property: PAC Write-Protection

Table 35-38. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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.

35.7.5.14. FIFO Space

Name: FIFOSPACE
Offset: 0x34
Reset: 0x0000

This register allows the user to identify the number of bytes present in each TX and RX FIFO.

Table 35-39. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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).
Note: When preloading is used, data written to shift register is not accessible in TXSPACE.

35.7.5.15. FIFO CPU Pointers

Name: FIFOPTR
Offset: 0x36
Reset: 0x0000

This register provides a copy of internal CPU TX and RX FIFO pointers.

Table 35-40. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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 Pointer
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 Pointer
These bits return the CPUWRPTR pointer value. These bits can be written only if the SERCOM is halted during debugging. When writing to DATA register, the DATA will be written to TXFIFO[CPUWRPTR] location.
Note: When preloading is used, data written to shift register is not accessible in CPUWRPTR.

35.8. SERCOM I^{2}C

35.8.1. Overview

The Inter-Integrated Circuit (I^{2}C) interface is one of the available modes in the Serial Communication Interface module (SERCOM).

The I^{2}C 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 I^{2}C host or an I^{2}C client. Both host and client have an interface containing a Shift register, a transmit buffer and a receive buffer. The I^{2}C host mode uses the SERCOM baud-rate generator, while the I^{2}C client mode uses the SERCOM address match logic.

35.8.2. Features

SERCOM I^{2}C includes the following features:

- Host or Client Operation
- Can be used with DMA

- Philips I²C Compatible
- SMBus Compatible
- PMBus™ Compatible
- Support of 100 kHz and 400 kHz, 1 MHz and 3.4 MHz I²C mode
- 32-bit Data Extension for better system bus utilization
- Up to 16-bytes internal 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

35.8.3. Block Diagram

Figure 35-47. I²C Single-Host Single-Client Interconnection

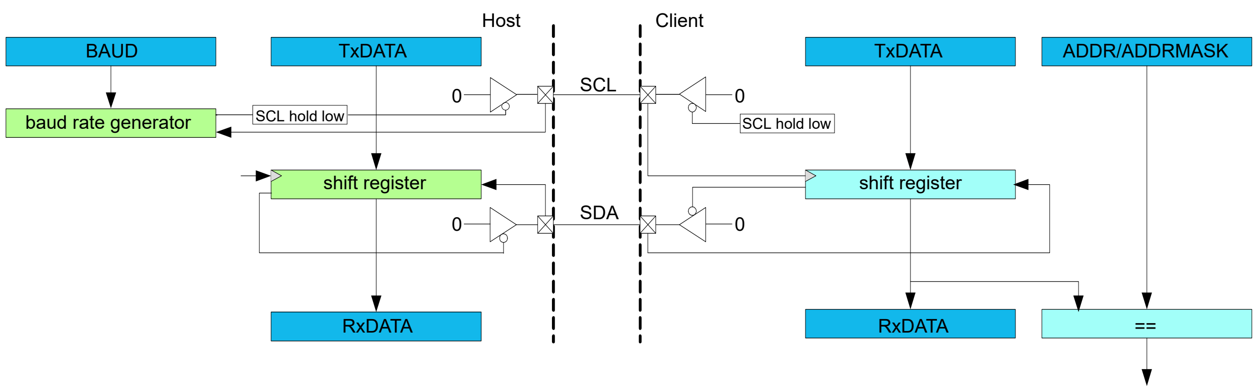


Table 35-41. I²C Pad Description

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)

35.8.4. Clocks

The SERCOM bus clock (CLK_SERCOM_APB) can be enabled and disabled in the Main Clock Controller. Refer to the Peripheral Clock Masking for details and default status of this clock.

Two generic clocks are used by SERCOM, GCLK_SERCOM_CORE and GCLK_SERCOM_SLOW. The core clock (GCLK_SERCOM_CORE) can clock the I²C when working as a host. The slow clock (GCLK_SERCOM_SLOW) is required only for certain functions, e.g. SMBus timing. These two clocks must be configured and enabled in the Generic Clock Controller (GCLK) before using the I²C.

These generic clocks are asynchronous to the bus clock (CLK_SERCOM_APB). Due to this asynchronicity, writes to certain registers will require synchronization between the clock domains.

35.8.5. Functional Description

35.8.5.1. Principle of Operation

The I²C 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 I²C host sending the Start condition, followed by a 7-bit address and a direction bit (read or write to/from the client).

The addressed I²C 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 I²C client or host, the I²C host takes action by either terminating the transaction by sending the Stop condition, or by sending a repeated start to transfer more data.

The following figure illustrates the possible transaction formats. These symbols will be used in the following descriptions.

Figure 35-48. Transaction Diagram Symbols

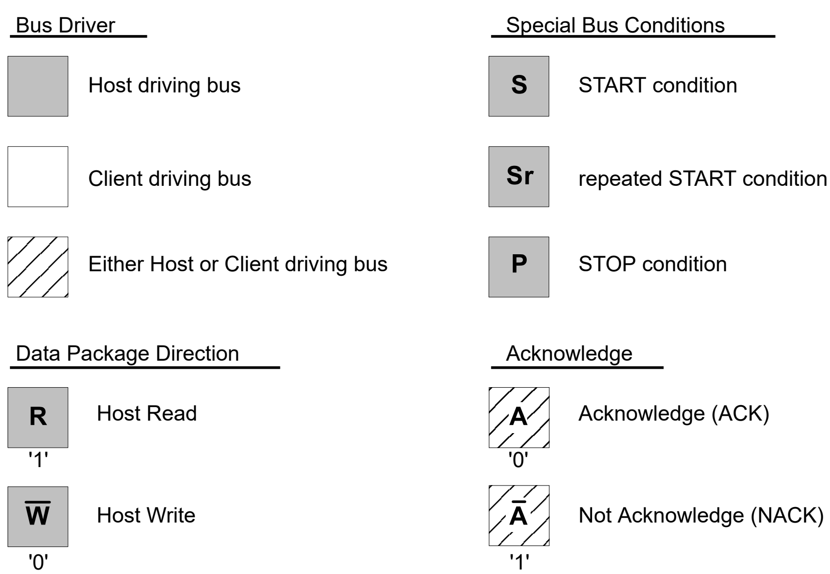
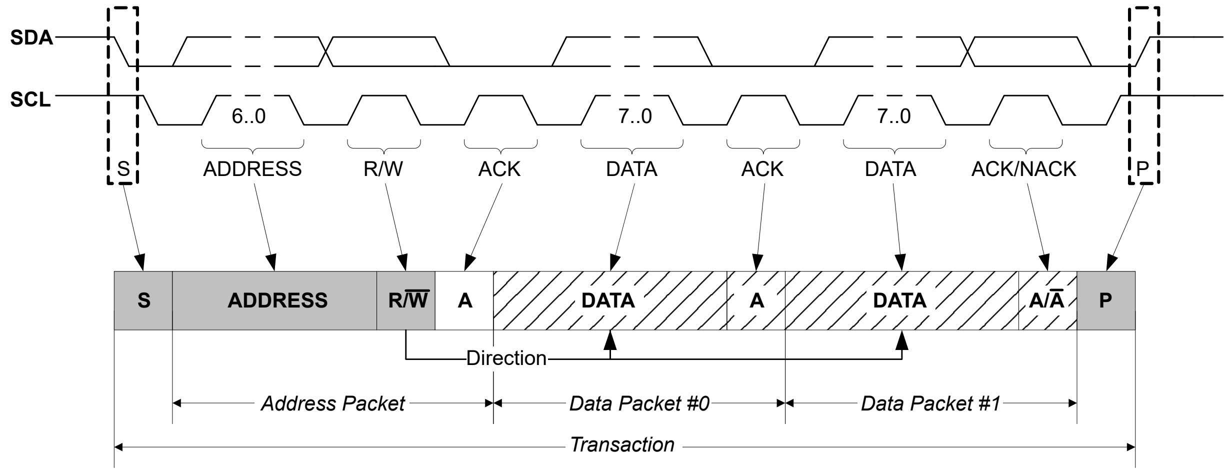


Figure 35-49. Basic I²C Transaction Diagram



35.8.5.2. Basic Operation

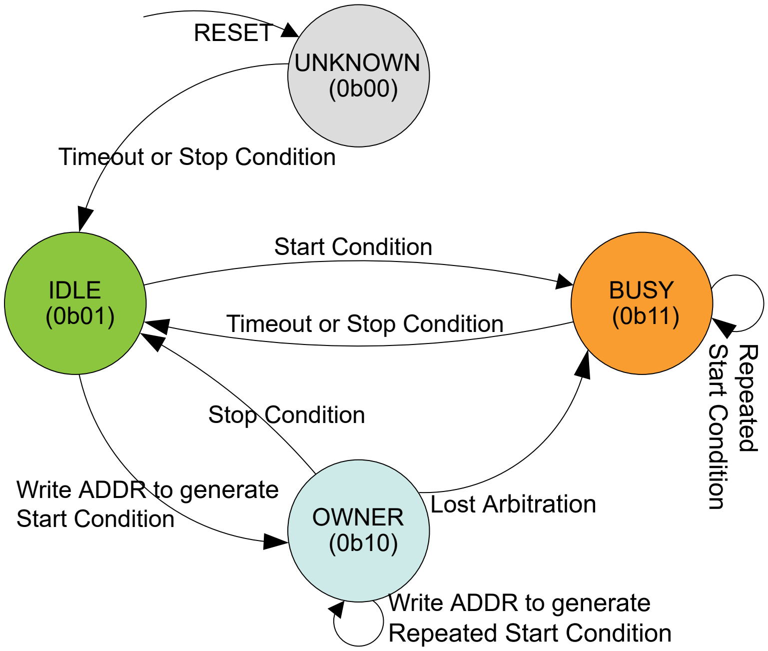
35.8.5.2.1. Initialization

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.LOWTOUT).
6. In Host mode:
   a. Select the inactive bus time-out in the Inactive Time-Out bit group in the CTRLA register (CTRLA.INACTOUT).
   b. Write the Baud Rate register (BAUD) to generate the desired baud rate.
7. 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.

35.8.5.2.2. 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 35-50. Bus State Diagram



The Bus state machine is active when the I^{2}C host is enabled.

After the I^{2}C 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 I^{2}C 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 I^{2}C 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 I^{2}C to hang. If this happens it is possible to recover from this state by a software Reset (CTRLA.SWRST='1').

35.8.5.2.3. I^{2}C Host Operation

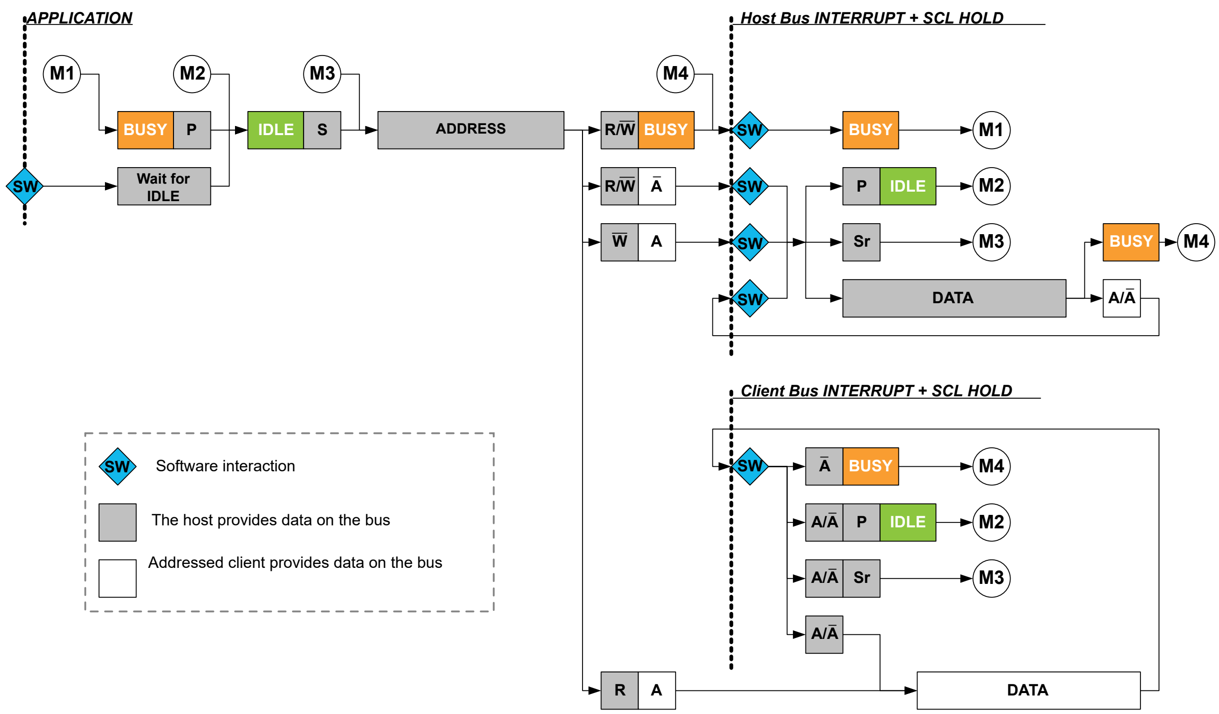
The I^{2}C host is byte-oriented and interrupt based. The number of interrupts generated is kept at a minimum by automatic handling of most incidents. The software driver complexity and code size are reduced by auto-triggering of operations, and a Special Smart mode, which can be enabled by the Smart Mode Enable bit in the Control A register (CTRLB.SMEN).

The I^{2}C 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 I^{2}C host operates according to the following diagram. 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 I^{2}C host operation throughout the document.

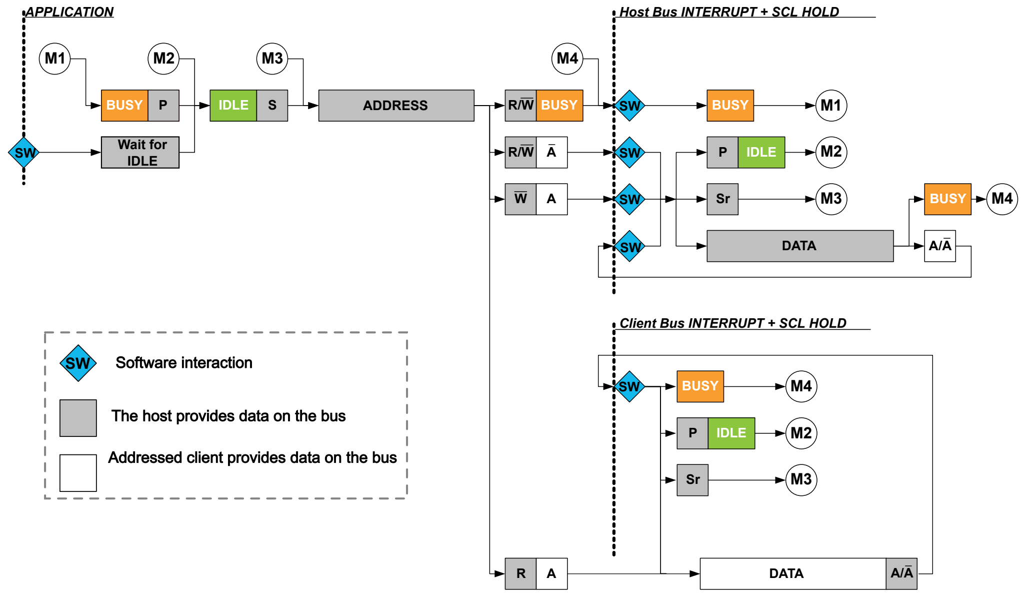
Figure 35-51. I^{2}C Host Behavioral Diagram (SCLSM=0)



In the second strategy (CTRLA.SCLSM=1), interrupts only occur after the ACK bit, as in the following diagram. This strategy can be used when it is not necessary to check DATA before acknowledging.

Note: I^{2}C High-speed (*Hs*) mode requires CTRLA.SCLSM=1.

Figure 35-52. I²C Host Behavioral Diagram (SCLSM=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

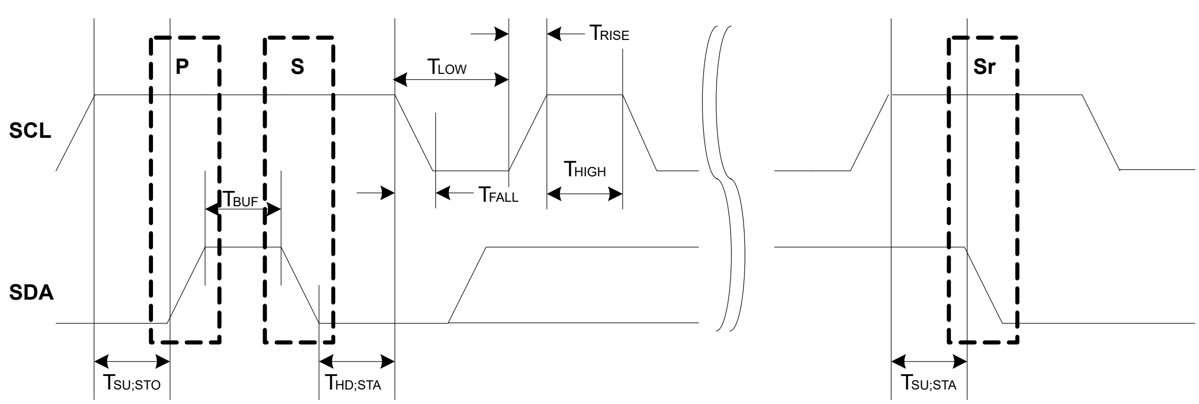
The Host clock configuration for *Sm, Fm,* and *Fm+* are described in *Clock Generation (Standard-Mode, Fast-Mode, and Fast-Mode Plus)*. For *Hs*, refer to *Host Clock Generation (High-Speed Mode)*.

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 (TLOW) and high (THIGH) times are determined by the Baud Rate register (BAUD), while the rise (TRISE) and fall (T FALL) times are determined by the bus topology. Because of the wired-AND logic of the bus, TFALL will be considered as part of TLOW. Likewise, TRISE will be in a state between TLOW and THIGH until a high state has been detected.

Figure 35-53. SCL Timing



The following parameters are timed using the SCL low time period TLOW. This comes from the Host Baud Rate Low bit group in the Baud Rate register (BAUD.BAUDLOW). When BAUD.BAUDLOW=0, the Host Baud Rate bit group in the Baud Rate register (BAUD.BAUD) determines it.

- **TLOW:** Low period of SCL clock
- **TSU;STO:** Set-up time for stop condition
- **TBUF:** Bus free time between stop and start conditions
- **THD;STA:** Hold time (repeated) start condition
- **TSU;STA:** Set-up time for repeated start condition
- **THIGH:** Timed using the SCL high time count from BAUD.BAUD
- **TRISE:** Determined by the bus impedance; for internal pull-ups
- **TFALL:** Determined by the open-drain current limit and bus impedance; can typically be regarded as zero

The SCL frequency is given by:

$$f_{SCL} = \frac{1}{T_{LOW} + T_{HIGH} + T_{RISE}}$$

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:

$$f_{SCL} = \frac{f_{GCLK}}{10 + 2BAUD + f_{GCLK} \cdot T_{RISE}}$$

When BAUD.BAUDLOW is non-zero, the following formula determines the SCL frequency:

$$f_{SCL} = \frac{f_{GCLK}}{10 + BAUD + BAUDLOW + f_{GCLK} \cdot T_{RISE}}$$

The following formulas can determine the SCL T_{LOW} and T_{HIGH} times:

$$T_{LOW} = \frac{BAUDLOW + 5}{f_{GCLK}}$$

$$T_{HIGH} = \frac{BAUD + 5}{f_{GCLK}}$$

Note: The I^{2}C 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.

Host Clock Generation (High-Speed Mode)

For I^{2}C *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 I^{2}C 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.

Transmitting Address Packets

The I^{2}C host starts a bus transaction by writing the I^{2}C client address to ADDR.ADDR and the direction bit, as described in 5.6.1 Principle of Operation. If the bus is busy, the I^{2}C host will wait until the bus becomes idle before continuing the operation. When the bus is idle, the I^{2}C host will issue a start condition on the bus. The I^{2}C host will then transmit an address packet using the address written to ADDR.ADDR. After the address packet has been transmitted by the I^{2}C 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 I^{2}C 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 I^{2}C 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 I^{2}C 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 I^{2}C host receives an ACK response from the I^{2}C 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 I^{2}C operation to continue:

- Initiate a data transmit operation by writing the data byte to be transmitted into DATA.DATA
- Transmit a new address packet by writing ADDR.ADDR. A repeated Start condition will automatically be inserted before the address packet.
- Issue a Stop condition, consequently terminating the transaction

Case 4: Address packet transmit complete – Read packet, Client on Bus set

If the I^{2}C host receives an ACK from the I^{2}C client, the I^{2}C host proceeds to receive the next byte of data from the I^{2}C client. When the first data byte is received, the Client on Bus bit in the Interrupt

Flag register (INTFLAG.SB) will be set and STATUS.RXNACK will be cleared. The clock hold is active at this point, preventing further activity on the bus.

In this case, the software implementation becomes highly protocol dependent. Three possible actions can enable the I^{2}C operation to continue:

- Let the I^{2}C 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.

Transmitting Data Packets

When an address packet with direction Host Write is transmitted successfully , INTFLAG.MB will be set. The I^{2}C host will start transmitting data via the I^{2}C bus by writing to DATA.DATA, and monitor continuously for packet collisions.

If a collision is detected, the I^{2}C host will lose arbitration and STATUS.ARBLOST will be set. If the transmit was successful, the I^{2}C host will receive an ACK bit from the I^{2}C client, and STATUS.RXNACK will be cleared. INTFLAG.MB will be set in both cases, regardless of arbitration outcome.

It is recommended to read STATUS.ARBLOST and handle the arbitration lost condition in the beginning of the I^{2}C Host on Bus interrupt. This can be done as there is no difference between handling address and data packet arbitration.

STATUS.RXNACK must be checked for each data packet transmitted before the next data packet transmission can commence. The I^{2}C host is not allowed to continue transmitting data packets if a NACK is received from the I^{2}C client.

Receiving Data Packets (SCLSM=0)

When INTFLAG.SB is set, the I^{2}C host will already have received one data packet. The I^{2}C 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.

Receiving Data Packets (SCLSM=1)

When INTFLAG.SB is set, the I^{2}C 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.

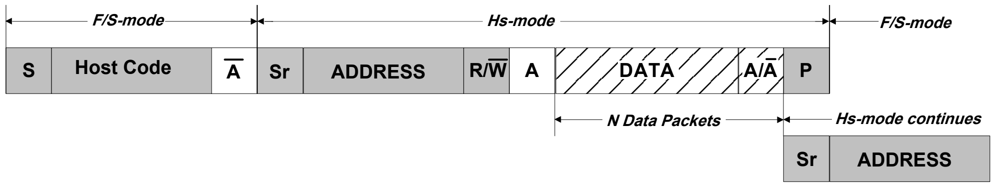
High-Speed Mode

High-speed transfers are a multi-step process, for further information, refer to the following figure.

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 meantime, 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 35-54. High Speed Transfer



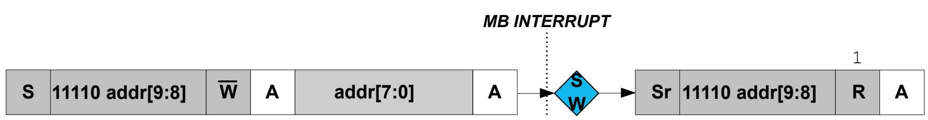
Transmitting in High-speed mode requires the I²C host to be configured in High-speed mode (CTRLA.SPEED=0x2) and the SCL Clock Stretch mode (CTRLA.SCLSM) bit set to '1'.

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 35-55. 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.

35.8.5.2.4. I²C Client Operation

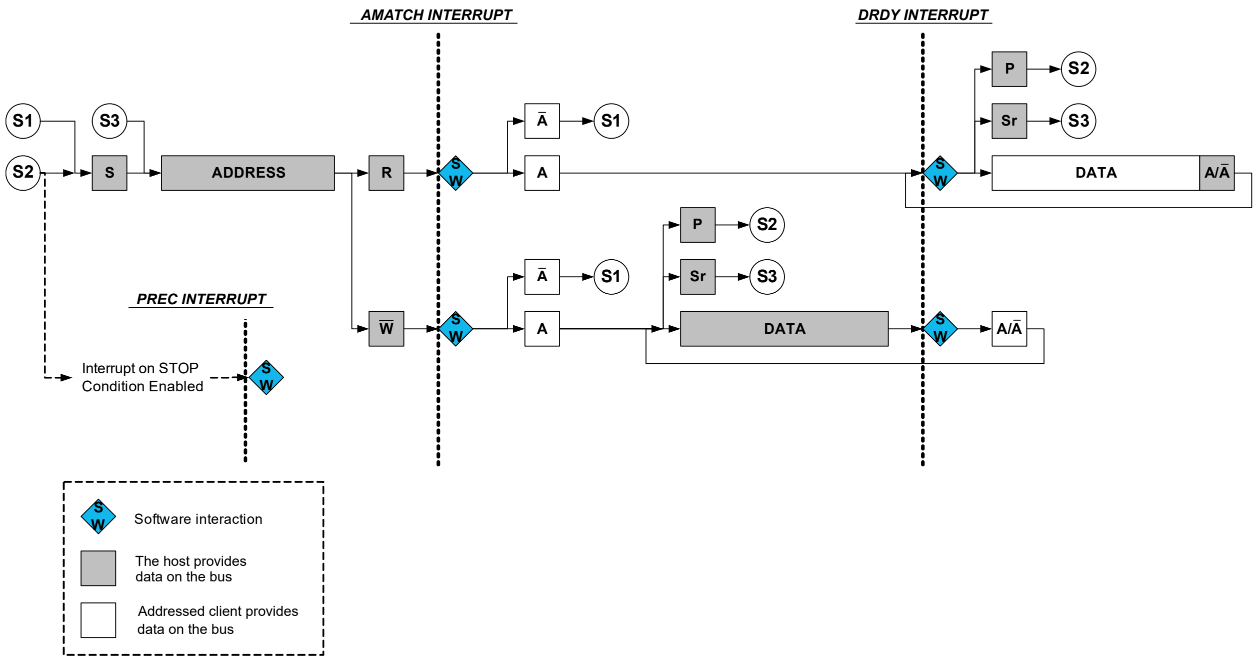
The I²C client is byte-oriented and interrupt-based. The number of interrupts generated is kept at a minimum by automatic handling of most events. The software driver complexity and code size are reduced by auto-triggering of operations, and a special smart mode, which can be enabled by the Smart Mode 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.

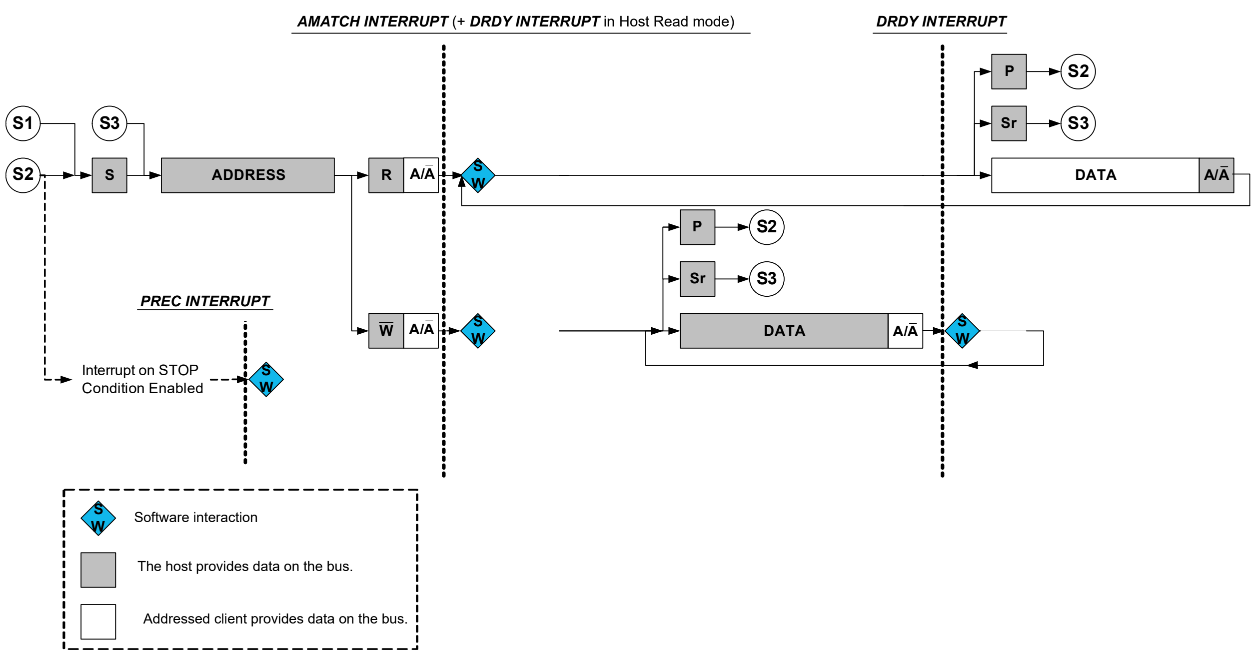
Figure 35-56. I^2C Behavioral Diagram (SCLSM=0)



In the second strategy (CTRLA.SCLSM=1), interrupts only occur after the ACK bit is sent as shown in Client Behavioral Diagram (SCLSM=1). 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 I^2C High-speed mode (*Hs*), SCLSM=1 is required.

Figure 35-57. I^2C Client Behavioral Diagram (SCLSM=1)



Receiving Address Packets (SCLSM=0)

When CTRLA.SCLSM=0, the I^{2}C client stretches the SCL line according to the I^{2}C Behavioral Diagram (SCLSM0). When the I^{2}C 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 I^{2}C 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 I^{2}C client clears INTFLAG.AMATCH. As the I^{2}C 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 I^{2}C 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 I^{2}C 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 I^{2}C host read operation. The SCL line is forced low, stretching the bus clock. If an ACK is sent, I^{2}C 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 I^{2}C 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 I^{2}C 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 I^{2}C host write operation. The SCL line is forced low, stretching the bus clock. If an ACK is sent, the I^{2}C client will wait for data to be received. Data, repeated start or stop can be received.

If a NACK is sent, the I^{2}C 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 I^{2}C 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.

Receiving Address Packets (SCLSM=1)

When SCLSM=1, the I^{2}C client will stretch the SCL line only after an ACK, see Client Behavioral Diagram(SCLSM=1). When the I^{2}C 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 I^{2}C 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.

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.

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.

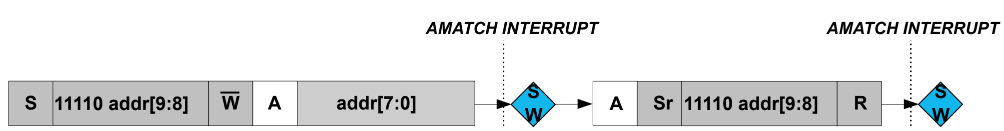
10-bit Addressing (Client)

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 35-58. 10-bit Addressing



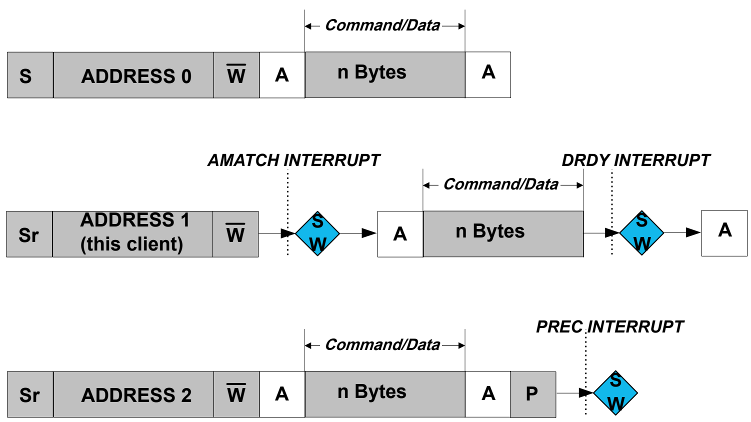
PMBus Group Command

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 35-59. PMBus Group Command Example



35.8.5.3. Additional Features

35.8.5.3.1. SMBus

The I^{2}C 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 KHz oscillator. The I^{2}C interface also allows for a SMBus compatible SDA hold time.

- **TTIMEOUT:** SCL low time of 25..35ms – Measured for a single SCL low period. It is enabled by CTRLA.LOWTOUTEN.
- **TLOW: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.
- **TLOW: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 writing the SMBus Input Buffer Enable bit in Control A register (CTRLA.SMBUFEN).

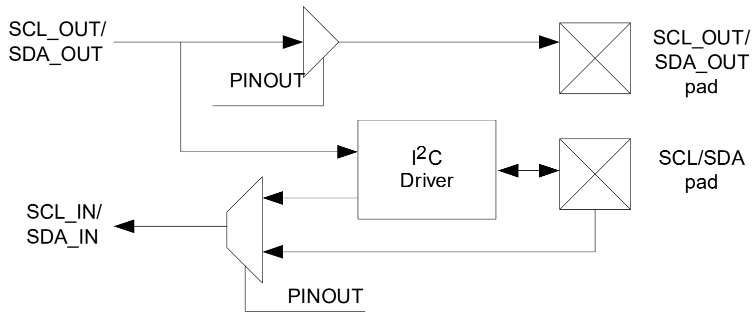
35.8.5.3.2. Smart Mode

The I^{2}C interface has a Smart mode that simplifies application code and minimizes the user interaction needed to adhere to the I^{2}C protocol. The Smart mode accomplishes this by automatically issuing an ACK or NACK (based on the content of CTRLB.ACKACT) as soon as DATA.DATA is read.

35.8.5.3.3. 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^{2}C tri-state drivers are bypassed, and an external I^{2}C compliant tri-state driver is needed when connecting to an I^{2}C bus.

Figure 35-60. I^{2}C Pad Interface



35.8.5.3.4. 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.

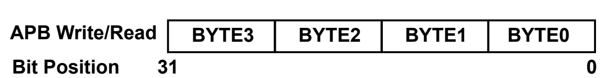
35.8.5.3.5. 32-bit Extension

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.

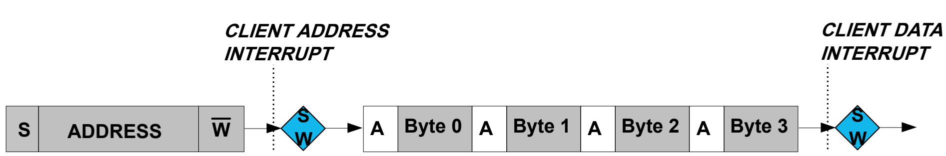
Figure 35-61. 32-bit Extension Byte Ordering



32-bit Extension Client Operation

The figure below 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 35-62. 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, so that 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 I^{2}C 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.

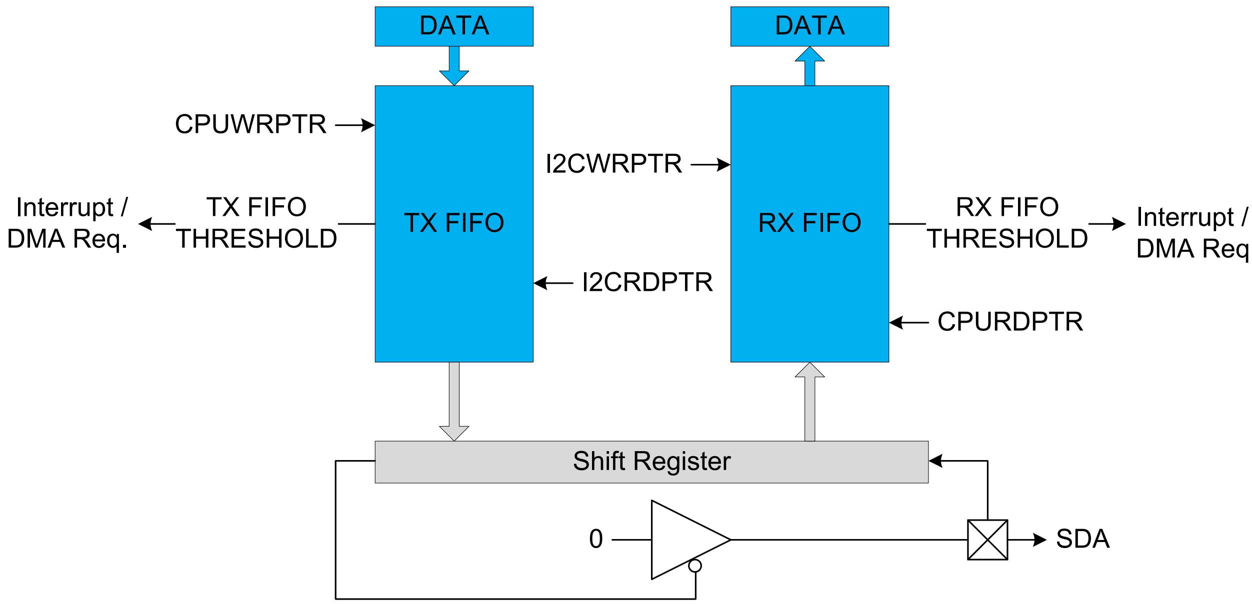
If a NACK is received by the client for a host write transaction before ADDR.LEN bytes, a STOP will be automatically generated, and the length error (STATUS.LENERR) is raised along with the INTFLAG.ERROR interrupt.

35.8.5.3.6. FIFO Operation

For better system bus utilization, the I^{2}C 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 I^{2}C Write pointer (I2CWRPTR) and the I^{2}C 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 I^{2}C 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 35-63. FIFO Overview



When using the I^{2}C 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 below.

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.

Hardware Actions in Host Mode

Table 35-42. Interrupts Request Conditions for Valid SERCOM I^{2}C 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 is 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	

Table 35-42. Interrupts Request Conditions for Valid SERCOM I²C Host Configurations (continued)

Direction	CTRLB.SMEN	CTRLC.DATA32B	LENGTH.LENEN	Action
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 35-43. 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

Hardware Actions in Client Mode

Table 35-44. 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 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 35-45. 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

Table 35-45. Bus Actions for Valid SERCOM I^2C Client Configurations (continued)

Direction	CTRLB.SMEN	CTRLC.DATA32B	LENGTH.LENEN	Actions
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

35.8.5.4. DMA and Interrupts

This chapter provides DMA and interrupt conditions when the 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 met. Each interrupt can be individually enabled by writing ‘1’ to the corresponding bit in the Interrupt Enable Set register (INTENSET), and disabled by writing ‘1’ to the corresponding bit in the Interrupt Enable Clear register (INTENCLR). An interrupt request is generated when the Interrupt flag is set and the corresponding interrupt is enabled. The interrupt request is active until the Interrupt flag is cleared, the interrupt is disabled or the I^2C is reset. See the INTFLAG (Client) or INTFLAG (Host) register for details on how to clear Interrupt flags.

Table 35-46. Module Request for SERCOM I^2C Client

Condition	Request DMA	Interrupt	Event
Data needed for transmit (TX) (Client Transmit mode)	Yes (request cleared when data is written)		N/A
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 35-47. Module Request for SERCOM I^2C Host

Condition	Request DMA	Interrupt	Event
Data needed for transmit (TX) (Host Transmit mode)	Yes (request cleared when data is written)		N/A
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	

35.8.5.4.1. DMA Operation

Smart mode must be enabled for DMA operation in the Control B register by writing CTRLB.SMEN=1.

Client DMA

When using the I^{2}C 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 I^{2}C 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.

Host DMA

When using the I^{2}C host with DMA, the ADDR register must be written with the desired address (ADDR.ADDR) and transaction length (ADDR.LEN), with 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^{2}C 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.

35.8.5.4.2. Interrupts

The I^{2}C 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^{2}C 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^{2}C 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.

35.8.5.5. Sleep Mode Operation

35.8.5.5.1. I^{2}C 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.

35.8.5.5.2. I^{2}C 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.

35.8.5.6. Synchronization

Due to asynchronicity between the main clock domain and the peripheral clock domains, some SERCOM registers need to be synchronized when written ("Write-Synchronized") or read ("Read-Synchronized").

The following bits are synchronized when written:

- Software Reset bit in the CTRLA register (CTRLA.SWRST)
- Enable bit in the CTRLA register (CTRLA.ENABLE)
- Command bits in CTRLB register (CTRLB.CMD)
- FIFO Clear bits in CTRLB register (CTRLB.FIFOCLR)
- Write to Bus State bits in the Status register (STATUS.BUSSTATE)
- Address bits in the Address register (ADDR.ADDR) when in host operation

The following registers are synchronized when written:

- Data (DATA) when in host operation
- Length (LENGTH) when in client operation

Required write synchronization is denoted by the "Write-Synchronized" property in the register description.

If a write-synchronized register is written while a synchronization is ongoing, a Bus Error exception will be generated.

35.8.6. Register Summary - I2C Client

For descriptions and definitions of both Register and bitfield properties, refer to Register Properties.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRLA	31:24		LOWTOUT			SCLSM		SPEED[1:0]	
		23:16	SEXTTOEN		SDAHOLD[1:0]				SMBUSEN	PINOUT
		15:8					SLEWRATE[1:0]		FILTSEL[1:0]	
		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]			

35.8.6.1. Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Table 35-48. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
		LOWTOUT			SCLSM		SPEED[1:0]	
Access		R/W			R/W		R/W	R/W
Reset		0			0		0	0

Bit	23	22	21	20	19	18	17	16
	SEXTTOEN		SDAHOLD[1:0]				SMBUSEN	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
					SLEWRATE[1:0]		FILTSEL[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
	RUNSTDBY			MODE[2:0]			ENABLE	SWRST
Access	R/W			R/W	R/W	R/W	R/W	R/W
Reset	0			0	0	0	0	0

Bit 30 – LOWTOUT SCL Low Time-Out 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.

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.
This bit is not synchronized.

Value	Description
0	SCL stretch according to I^2C Behavioral Diagram (SCLSM = 0)
1	SCL stretch only after ACK bit according to I^2C Client Behavioral Diagram SCLSM = 1

Bits 25:24 – SPEED[1:0] Transfer Speed

These bits define bus speed.
These bits are not synchronized.

Value	Description
0x0	Standard-mode (Sm) up to 100 kHz and Fast-mode (Fm) up to 400 kHz
0x1	Fast-mode Plus (Fm+) up to 1 MHz
0x2	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.

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.

These bits are not synchronized.

Value	Name	Description
0x0	DIS	Disabled
0x1	75	50-100ns hold time
0x2	450	300-600ns hold time
0x3	600	400-800ns hold time

Bit 17 – SMBUSEN SMBus Input Buffer Enable

This bit enables SMBus-compatible I/O logic level.

This bit is not synchronized.

Value	Description
0	SMBus input buffer is disabled.
1	SMBus input buffer is enabled.

Bit 16 – PINOUT Pin Usage

This bit sets the pin usage to either two- or four-wire operation:

This bit is not synchronized.

Value	Description
0	4-wire operation disabled
1	4-wire operation enabled

Bits 11:10 – SLEWRATE[1:0] Slew Rate Enable

This bit enables the I/O pins slew rate control.

These bits are not synchronized.

Note: If an I^2C function is enabled on a pin, the corresponding PINCFGn.SLEWLIM for that pin MUST = 0x00.

Value	Name	Description
0x0	SM	Standard Mode
0x1	FM	Fast Mode
0x2	FMP	Fast Mode Plus
0x3	HS	High-speed Mode

Bits 9:8 – FILTSEL[1:0] Input Filter Selection

These bits define filter length applied to the input signals.

These bits are not synchronized.

Value	Name	Description
0x0	DIS	Disabled
0x1	50F	Minimum 50ns filter (SCL fast mode)
0x2	50E	Minimum 50ns filter (SDA even mode)
0x3	10	Minimum 10ns filter

Bit 7 – RUNSTDBY Run in Standby

This bit defines the functionality in standby sleep mode.
This bit is not synchronized.

Value	Description
0	Disabled – All reception is dropped.
1	Wake on address match, if enabled.

Bits 4:2 – MODE[2:0] Operating Mode

These bits must be written to 0x04 to select the I^2C client serial communication interface of the SERCOM.
These bits are not synchronized.

Bit 1 – ENABLE Enable

Due to synchronization, there is delay from writing CTRLA.ENABLE until the peripheral is enabled/disabled. The value written to CTRL.ENABLE will read back immediately and the Enable Synchronization Busy bit in the Synchronization Busy register (SYNCBUSY.ENABLE) will be set. SYNCBUSY.ENABLE will be cleared when the operation is complete.
This bit is not enable-protected.

Value	Description
0	The peripheral is disabled or being disabled.
1	The peripheral is enabled.

Bit 0 – SWRST Software Reset

Writing '0' to this bit has no effect.
Writing '1' to this bit resets all registers in the SERCOM, except DBGCTRL, to their initial state, and the SERCOM 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 a bus error. Reading any register will return the reset value of the register.
Due to synchronization, there is a delay from writing CTRLA.SWRST until the reset is complete. CTRLA.SWRST and SYNCBUSY.SWRST will both be cleared when the reset is complete.
This bit is not enable-protected.

Notes:

1. When the CTRLA.SWRST is written, the user should poll the SYNCB.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 the hardware.

Value	Description
0	There is no reset operation ongoing.
1	The reset operation is ongoing.

35.8.6.2. Control B

Name: CTRLB
Offset: 0x04
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected, Write-Synchronized

Table 35-49. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	FIFOCLR[1:0]					ACKACT	CMD[1:0]	
Access	R/W	R/W				R/W	W	W
Reset	0	0				0	0	0

Bit	15	14	13	12	11	10	9	8
	AMODE[1:0]					AACKEN	GCMD	SMEN
Access	R/W	R/W				R/W	R/W	R/W
Reset	0	0				0	0	0

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bits 23:22 – FIFOCLR[1:0] FIFO Clear

When these bits are set, the corresponding FIFO will be cleared. The bits will automatically clear when SYNCBUSY.SYSOP = 0.
These bits are not enable-protected.

FIFOCLR[1:0]	Name	Description
0x0	NONE	No action
0x1	TXFIFO	Clear TX FIFO
0x2	RXFIFO	Clear RX FIFO
0x3	BOTH	Clear both TX/RX FIFO

Bit 18 – ACKACT Acknowledge Action

This bit defines the client's acknowledge behavior after an address or data byte is received from the host. The acknowledge action is executed when a command is written to the CMD bits. If smart mode is enabled (CTRLB.SMEN=1), the acknowledge action is performed when the DATA register is read.
ACKACT shall not be updated more than once between each peripheral interrupts request.
This bit is not enable-protected. This bit is not write-synchronized.

Note:
CTRLB.ACKACT shall not be updated more than once between each peripheral interrupt request.

Value	Description
0	Send ACK
1	Send NACK

Bits 17:16 – CMD[1:0] Command

This bit field triggers the client operation as shown 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.
This bit is not enable-protected. This bit is not write-synchronized.

Table 35-50. Command Description

CMD[1:0]	STATUS.DIR Value	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.
These bits are not write-synchronized.

Value	Name	Description
0x0	MASK	The client responds to the address written in ADDR.ADDR masked by the value in ADDR.ADDRMASK. See SERCOM – Serial Communication Interface for additional information.
0x1	2_ADDRS	The client responds to the two unique addresses in ADDR.ADDR and ADDR.ADDRMASK.
0x2	RANGE	The client responds to the range of addresses: ADDR.ADDRMASK < address < ADDR.ADDR
0x3	-	Reserved.

Bit 10 – AACKEN Automatic Acknowledge Enable

This bit enables the address to be automatically acknowledged if there is an address match.
This bit is not write-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 Received 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.
This bit is not write-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.
This bit is not write-synchronized.

Value	Description
0	Smart mode is disabled.
1	Smart mode is enabled.

35.8.6.3. Control C

Name: CTRLC
Offset: 0x08
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Table 35-51. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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.

TXTRHOLD	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.

RXTRHOLD	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:

$t_{SU:DAT}$ = (6 + 16 * SDASETUP) APB CLOCK_SERCOM periods.

35.8.6.4. 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) On read, a bit value of zero indicates the associated interrupt is disabled while a bit value of one indicates the associated interrupt is enabled.

Table 35-52. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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 Disable
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.

Bit 4 – RXFF RX FIFO Full Interrupt Disable
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.

Bit 3 – TXFE TX FIFO Empty Interrupt Disable
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.

Bit 2 – DRDY Data Ready Interrupt Disable
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.

Bit 1 – AMATCH Address Match Interrupt Disable
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.

Bit 0 – PREC Stop Received Interrupt Disable
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.

35.8.6.5. 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) On read, a bit value of zero indicates the associated interrupt is disabled while a bit value of one indicates the associated interrupt is enabled.

Table 35-53. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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.

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.

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.

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.

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.

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.

35.8.6.6. Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x18
Reset: 0x00
Property: -

Note: Interrupt flags must be cleared and then read back to confirm the clear before exiting the ISR to avoid double interrupts.

Table 35-54. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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: 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 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 2 – DRDY Data Ready
This flag is set when a I^2C client byte transmission is successfully completed.
The flag is cleared by hardware by either:

- Writing to the DATA register.
- Reading the DATA register with Smart mode enabled.
- Writing a valid command to the CMD register.

Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the Data Ready Interrupt flag.

Bit 1 – AMATCH Address Match
This flag is set when the I^2C client address match logic detects that a valid address has been received.
The flag is cleared by hardware when CTRL.CMD is written.
Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the Address Match Interrupt flag. When cleared, an ACK/NACK will be sent according to CTRLB.ACKACT.

Bit 0 – PREC Stop Received

This flag is set when a Stop condition is detected for a transaction being processed. A Stop condition detected between a bus host and another client will not set this flag, unless the PMBus Group Command is enabled in the Control B register (CTRLB.GCMD=1).
This flag is cleared by hardware after a command is issued on the next address match.
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the Stop Received Interrupt flag.

35.8.6.7. Status

Name: STATUS
Offset: 0x1A
Reset: 0x0000
Property: -

Table 35-55. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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 I^{2}C 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 0x3 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 immediately released the SDA and SCL lines and waited 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 this bit.

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.

35.8.6.8. Synchronization Busy

Name: SYNCBUSY
Offset: 0x1C
Reset: 0x00000000
Property: -

Table 35-56. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
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 the LENGTH register requires synchronization. When written, this bit will be set until synchronization is complete. If LENGTH is written while SYNCBUSY.LENGTH is asserted, a bus 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. In this case 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.

35.8.6.9. Length

Name: LENGTH
Offset: 0x22
Reset: 0x0000
Property: Write-Synchronized

Table 35-57. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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.

35.8.6.10. Address

Name: ADDR
Offset: 0x24
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Table 35-58. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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.

35.8.6.11. Data

Name: DATA
Offset: 0x28
Reset: 0x00000000
Property: -

Table 35-59. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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.

35.8.6.12. 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.

Table 35-60. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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).

35.8.6.13. FIFO CPU Pointers

Name: FIFOPTR
Offset: 0x36
Reset: 0x0000
Property: -

This register provides a copy of internal CPU TX and RX FIFO pointers.

Table 35-61. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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 Pointer
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 Pointer
These bits return the CPUWRPTR pointer value. These bits can be written only if the SERCOM is halted during debugging. When writing to DATA register, the DATA will be written to TXFIFO[CPUWRPTR] location.

35.8.7. Register Summary - I2C Host

For descriptions and definitions of both Register and bitfield properties, refer to Register Properties.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRLA	31:24		LOWTOUT	INACTOUT[1:0]		SCLSM		SPEED[1:0]	
		23:16	SEXTTOEN	MEXTTOEN	SDAHOLD[1:0]				SMBUSEN	PINOUT
		15:8					SLEWRATE[1:0]		FILTSEL[1:0]	
		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]			

35.8.7.1. Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Table 35-62. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
		LOWTOUT	INACTOUT[1:0]		SCLSM		SPEED[1:0]	
Access		R/W	R/W	R/W	R/W		R/W	R/W
Reset		0	0	0	0		0	0

Bit	23	22	21	20	19	18	17	16
	SEXTTOEN	MEXTTOEN	SDAHOLD[1:0]				SMBUSEN	PINOUT
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
					SLEWRATE[1:0]		FILTSEL[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
	RUNSTDBY			MODE[2:0]			ENABLE	SWRST
Access	R/W			R/W	R/W	R/W	R/W	R/W
Reset	0			0	0	0	0	0

Bit 30 – LOWTOUT SCL Low Time-Out Enable

This bit enables the SCL low time-out. If SCL is held low for 25 ms-35 ms, 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.

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 arises when either an I^{2}C 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 100 kHz baud rate.

These bits are 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.
This bit is not synchronized.

Value	Description
0	SCL stretch according to I²C Host Behavioral Diagram (SCLSM = 0).
1	SCL stretch only after ACK bit, I²C Host Behavioral Diagram (SCLSM = 1).

Bits 25:24 – SPEED[1:0] Transfer Speed

These bits define bus speed.
These bits are not synchronized.

Value	Description
0x0	Standard-mode (Sm) up to 100 kHz and Fast-mode (Fm) up to 400 kHz
0x1	Fast-mode Plus (Fm+) up to 1 MHz
0x2	High-speed mode (Hs-mode) up to 3.4 MHz
0x3	Reserved

Bit 23 – SEXTTOEN Client SCL Low Extend Time-Out Enable

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 released. The MEXTTOUT and BUSERR status bits will be set.
This bit is not synchronized.

Value	Description
0	Time-out disabled
1	Time-out enabled

Bit 22 – MEXTTOEN Host SCL Low Extend Time-Out Enable

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.
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.
These bits are 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 17 – SMBUSEN SMBus Input Buffer Enable

This bit enables SMBus-compatible I/O logic level.
This bit is not synchronized.

Value	Description
0	SMBus input buffer is disabled.
1	SMBus input buffer is enabled.

Bit 16 – PINOUT Pin Usage

This bit set the pin usage to either two- or four-wire operation:
This bit is not synchronized.

Value	Description
0	4-wire operation disabled.
1	4-wire operation enabled.

Bits 11:10 – SLEWRATE[1:0] Slew Rate Enable

This bit enables the I/O pins slew rate control.
This bit is not synchronized.

Value	Name	Description
0x0	SM	Standard Mode
0x1	FM	Fast Mode
0x2	FMP	Fast Mode Plus
0x3	HS	High-speed Mode

Bits 9:8 – FILTSEL[1:0] Input Filter Selection

These bits define filter length applied to the input signals.
These bits are not synchronized.

Value	Name	Description
0x0	DIS	Disabled
0x1	-	Reserved
0x2	50E	Minimum 50ns filter (SDA even mode)
0x3	10	Minimum 10ns filter

Bit 7 – RUNSTDBY Run in Standby

This bit defines the functionality in standby sleep mode.
This bit is not synchronized.

Value	Description
0	GCLK_SERCOMx_CORE is disabled and the I^{2}C 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^{2}C host serial communication interface of the SERCOM.
These bits are not synchronized.

Bit 1 – ENABLE Enable

Due to synchronization, there is delay from writing CTRLA.ENABLE until the peripheral is enabled/disabled. The value written to CTRL.ENABLE will read back immediately and the Synchronization Enable Busy bit in the Synchronization Busy register (SYNCBUSY.ENABLE) will be set. SYNCBUSY.ENABLE will be cleared when the operation is complete.
This bit is not enable-protected.

Value	Description
0	The peripheral is disabled or being disabled.
1	The peripheral is enabled.

Bit 0 – SWRST Software Reset

Writing '0' to this bit has no effect.

Writing '1' to this bit resets all registers in the SERCOM, except DBGCTRL, to their initial state, and the SERCOM 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 a bus error. Reading any register will return the reset value of the register.

Due to synchronization there is a delay from writing CTRLA.SWRST until the reset is complete. CTRLA.SWRST and SYNCBUSY.SWRST will both be cleared when the reset is complete.

This bit is not enable-protected.

Notes:

1. When the CTRLA.SWRST is written, the user should poll the SYNCB.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 the hardware.

Value	Description
0	There is no reset operation ongoing.
1	The reset operation is ongoing.

35.8.7.2. Control B

Name: CTRLB
Offset: 0x04
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected, Write-Synchronized

Table 35-63. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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

When these bits are set, the corresponding FIFO will be cleared. The bits will automatically clear when SYNCBUSY.SYSOP = 0.
These bits are not enable-protected.

FIFOCLR[1:0]	Name	Description
0x0	NONE	No action
0x1	TXFIFO	Clear TX FIFO
0x2	RXFIFO	Clear RX FIFO
0x3	BOTH	Clear both TX/RX FIFO

Bit 18 – ACKACT Acknowledge Action

This bit defines the I^{2}C host's acknowledge behavior after a data byte is received from the I^{2}C client.
The acknowledge action is executed when a command is written to CTRLB.CMD, or if Smart mode is enabled (CTRLB.SMEN is written to one), when DATA.DATA is read.
This bit is not enable-protected.
This bit is not write-synchronized.

Value	Description
0	Send ACK.
1	Send NACK.

Bits 17:16 – CMD[1:0] Command

Writing these bits triggers a host operation as described below. The CMD bits are strobe bits, and always read as 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.

Issuing a command will set the System Operation bit in the Synchronization Busy register (SYNCBUSY.SYSOP).

Table 35-64. Command Description

CMD[1:0]	Direction	Action
0x0	X	(No action)
0x1	X	Execute acknowledge action succeeded by repeated Start
0x2	0 (Write)	No operation
	1 (Read)	Execute acknowledge action succeeded by a byte read operation
0x3	X	Execute acknowledge action succeeded by issuing a Stop condition

These bits are not enable-protected.

Bit 9 – QCEN Quick Command Enable

This bit is not write-synchronized.

Value	Description
0	Quick Command is disabled.
1	Quick Command is enabled.

Bit 8 – SMEN Smart Mode Enable

When Smart mode is enabled, acknowledge action is sent when DATA.DATA is read.

This bit is not write-synchronized.

Value	Description
0	Smart mode is disabled.
1	Smart mode is enabled.

35.8.7.3. Control C

Name: CTRLC
Offset: 0x08
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Table 35-65. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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.

TXTRHOLD	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.

RXTRHOLD	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

35.8.7.4. Baud Rate

Name: BAUD
Offset: 0x0C
Reset: 0x0000
Property: PAC Write-Protection, Enable-Protected

Table 35-66. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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 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 Clock Generation - Baud-Rate Generator.

35.8.7.5. Interrupt Enable Clear

Name: INTENCLR
Offset: 0x14
Reset: 0x00
Property: PAC Write-Protection

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set register (INTENSET). On read, a bit value of zero indicates the associated interrupt is disabled while a bit value of one indicates the associated interrupt is enabled.

Table 35-67. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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 Disable
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.

Bit 4 – RXFF RX FIFO Full Interrupt Disable
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.

Bit 3 – TXFE TX FIFO Empty Interrupt Disable
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.

Bit 1 – SB Client on Bus Interrupt Disable
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 Disable
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.

35.8.7.6. 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). On read, a bit value of zero indicates the associated interrupt is disabled while a bit value of one indicates the associated interrupt is enabled.

Table 35-68. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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.

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.

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.

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.

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.

35.8.7.7. Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x18
Reset: 0x00
Property: -

Note: Interrupt flags must be cleared and then read back to confirm the clear before exiting the ISR to avoid double interrupts.

Table 35-69. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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: 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 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 1 – SB Client on Bus

The Client on Bus flag (SB) is set when a byte is successfully received in Host Read mode, for example, no arbitration lost or bus error occurred during the operation. When this flag is set, the host forces the SCL line low, stretching the I^2C clock period. The SCL line will be released and the SB bit 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.

35.8.7.8. Status

Name: STATUS
Offset: 0x1A
Reset: 0x0000
Property: Write-Synchronized

Table 35-70. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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.

Value	Description
0	No Error
1	Transaction Length Error has occurred.

Bit 9 – SEXTTOUT Client SCL Low Extend Time-Out

This bit is set if a client SCL low extend time-out occurs.
This bit is 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.

Value	Description
0	No Time-Out
1	Client SCL Low Time-Out has occurred.

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.

Value	Description
0	No Time-Out
1	Host SCL Low Extend Time-Out has occurred

Bit 7 – CLKHOLD Clock Hold

This bit is set when the host is holding the SCL line low, stretching the I²C 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.

Value	Description
0	No Time-Out
1	SCL Low Time-Out has occurred

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 I²C Bus state.
When in UNKNOWN state, writing 0x1 to BUSSTATE forces the bus state into the IDLE state. The bus state cannot be forced into any other state.
Writing BUSSTATE to idle will set SYNCBUSY.SYSOP.
These bits are write synchronized.

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.

Value	Description
0	No Error
1	Arbitration Lost

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.

Value	Description
0	No Error
1	Bus Error has occurred

35.8.7.9. Synchronization Busy

Name: SYNCBUSY
Offset: 0x1C
Reset: 0x00000000

Table 35-71. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						SYSOP	ENABLE	SWRST
Access						R	R	R
Reset						0	0	0

Bit 2 – SYSOP System Operation Synchronization Busy

Writing CTRLB.CMD, STATUS.BUSSTATE, ADDR, or DATA when the SERCOM is enabled requires synchronization.
Writing CTRLB.FIFOCLR when the SERCOM is enabled and Smart Mode 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.

35.8.7.10. Address

Name: ADDR
Offset: 0x24
Reset: 0x0000
Property: Write-Synchronized

Table 35-72. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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 Enable
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.

Note: 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.

35.8.7.11. Data

Name: DATA
Offset: 0x28
Reset: 0x00000000
Property: Read Synchronized, Write Synchronized

Table 35-73. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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 to this is reading the last data byte after the stop condition has been sent.

Accessing DATA.DATA auto-triggers I^{2}C 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.

35.8.7.12. Debug Control

Name: DBGCTRL
Offset: 0x30
Reset: 0x00
Property: PAC Write-Protection

Table 35-74. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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.

35.8.7.13. 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.

Table 35-75. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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).

35.8.7.14. FIFO CPU Pointers

Name: FIFOPTR
Offset: 0x36
Reset: 0x0000
Property: -

This register provides a copy of internal CPU TX and RX FIFO pointers.

Table 35-76. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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 Pointer
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 Pointer
These bits return the CPUWRPTR pointer value. These bits can be written only if the SERCOM is halted during debugging. When writing to DATA register, the DATA will be written to TXFIFO[CPUWRPTR] location.

36. Serial Quad Interface (SQI)

36.1. Overview

The SQI module is a synchronous serial interface that provides access to serial Flash memories and other serial devices. The SQI module supports Single Lane (identical to SPI), Dual Lane, and Quad Lane interface modes.

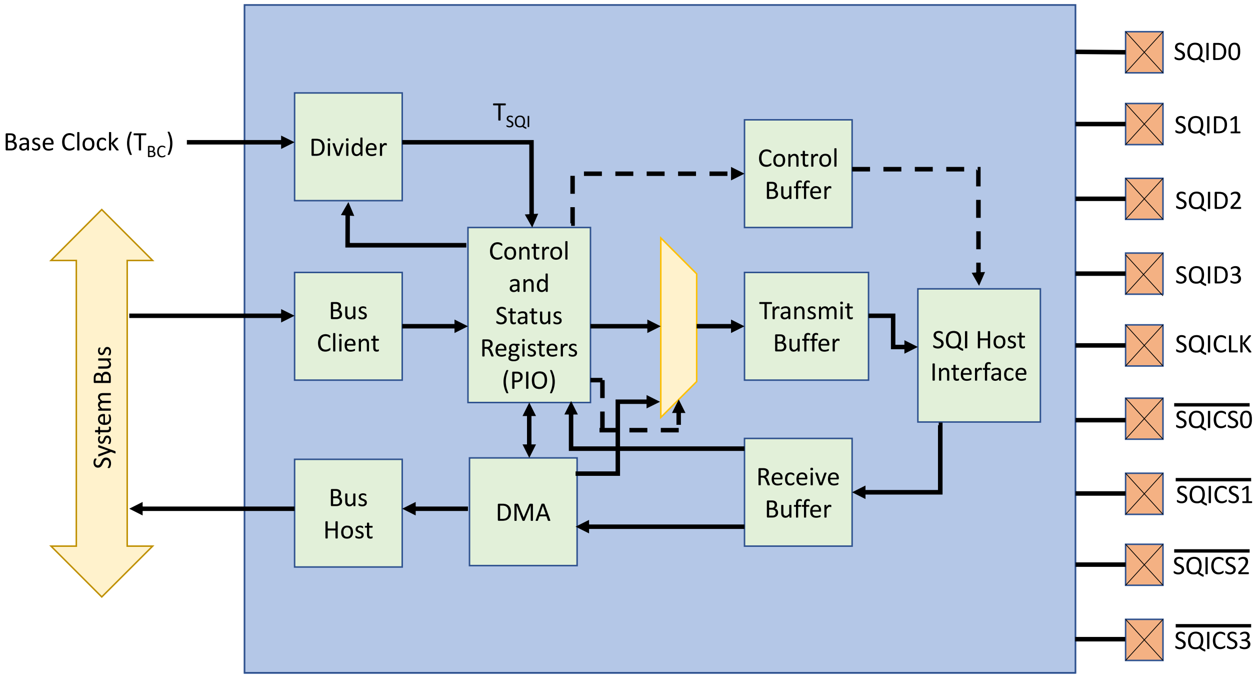
36.2. Features

The following are key features of the SQI module:

- Supports Single, Dual, and Quad Lane modes
- Supports Single Data Rate (SDR)
- Programmable command sequence
- Data transfer:
  - Programmed I/O mode(PIO)
  - Buffer descriptor DMA
  - Execute-In-Place (xIP)
- Supports SPI Mode 0 and Mode 3
- Programmable Clock Polarity (CPOL) and Clock Phase (CPHA) bits
- Supports up to four Chip Selects
- Supports up to four bytes of Flash address
- Programmable interrupt thresholds
- 256-byte transmit data buffer
- 256-byte receive data buffer
- 4-word controller buffer

36.3. Block Diagram

Figure 36-1. SQI Module Block Diagram



36.4. Peripheral Dependencies

Peripheral Name	Base Address	NVIC IRQ Index: Source	MCLK AHBx/APBx Clock Enable Mask Bit	GCLK Peripheral Channel Clock Name:Register	Power Domain
SQI	0x0FFF_F000	146 : SQI	MCLK.CLKMSK0[16]	CLK_SQI : GCLK.PCHCTRL[43]	VDDCORE_SW

36.5. Functional Description

The SQI module, which is an industry standard synchronous serial link, helps communicate with multiple SPI compatible devices such as serial EEPROMs and serial Flash devices.

The SQI module has three interfaces, one external to the device (SQI Bus Interface) that connects to the external Flash memories or other serial devices, and two internal (Bus Client interface for control register reads/writes and Bus Host for data transfers), as illustrated in the Block Diagram.

The SQI bus interface consists of four data lines (SQID3-SQID0), a clock line (SQICLK), and four select lines (SQICS0, SQICS1, SQICS2 and SQICS3). As mentioned earlier, the SQI module supports Single Lane (SPI mode), Dual Lane, and Quad Lane modes of operation.

The SQI module operates in both Single Data Rate (SDR) and Double Data Rate (DDR) modes. In DDR mode, the data transition occurs on both edges of the clock providing double the throughput.

Note: The SQI module is a half-duplex, synchronous serial interface when in Host mode of operation.

The SQI module has configurable transmit and receive buffers, programmable baud rates through the internal clock divider, clock phase, and clock polarity control for efficient data operations. Transmit and receive buffers can be accessed through SQI1TXDATA and SQI1RXDATA registers. Similarly, the control buffer can be accessed through the SQI1CON register and is mainly used to pipeline the operations. The SQI module operates in three transfer modes: DMA, PIO, and XIP. All three modes use the control buffer to pipeline the command/data sequences on the SQI bus.

The SQI module supports two data flow modes: SPI Mode 0 and Mode 3. Each transfer mode (XIP/PIO/DMA) can use any of the data flow modes as desired by the application.

DMA and PIO modes are typically used to transfer the data to and from external serial Flash memory, whereas, eXecute In Place (XIP) mode is used to execute the code out of the external serial Flash memory. DMA mode uses the internal DMA engine and buffer descriptors to transfer data between source and destination memory spaces off-loading the Host processor during which time, accessing SQI1TXDATA, SQI1RXDATA, and SQI1CON functionally will not yield expected results. However, PIO mode engages the Host processor to access the contents of the external serial Flash memory using a bit-band method through the transmit and receive data registers. Refer to *SQI Transfer Modes* for a detailed description of each transfer mode.

The SQI module supports automatic memory status check reducing software burden. For a block diagram of the SQI module, refer to the Block Diagram.

36.5.1. SQI Operation

The SQI peripheral is primarily used to communicate with Serial Flash memory devices. The Serial Flash devices support operations such as ERASE, READ, and WRITE through a set of command sequences, which are issued by a host controller, in this case the SQI peripheral. The SQI peripheral facilitates these command sequences through the following prominent interface features:

- Single, Dual, or Quad lane modes
- Single Data Rate (SDR) or Double Data Rate (DDR)speeds
- SPI Mode 0 or Mode3
- DMA, PIO, or XIP transfer modes
- Flash status check
- Tap delays at high interface speeds

36.5.1.1. Single, Dual or Quad Lane Modes

The lane modes (single/dual/quad), as the names imply set the interface to exercise transactions using single (SQID0), dual (SQID0, SQID1) or quad (SQID0-SQID3) data lanes. The majority of serial Flash devices provide commands specifically to exercise the transactions in a specific lane mode (i.e., JEDEC-ID to read device ID in single lane mode and QJID to read the same device ID in quad lane mode in case of SST26VF series devices).

36.5.1.2. Single Data Rate (SDR) Speeds

In SDR mode, the data transaction occurs only on the rising edge of the clock, whereas in DDR mode the transactions occur on both the rising and falling edge of the clock, providing double the throughput. Some Serial Flash devices support commands specifically aimed at DDR mode (i.e., the 4DDRQIOR command to read from Flash using quad I/O in DDR mode in case of Spansion memories supporting DDR mode).

The following figures show the high-speed read quad lane sequence in SDR mode and DDR mode, respectively.

Figure 36-2. High-Speed Read Quad Lane Sequence in SDR Mode

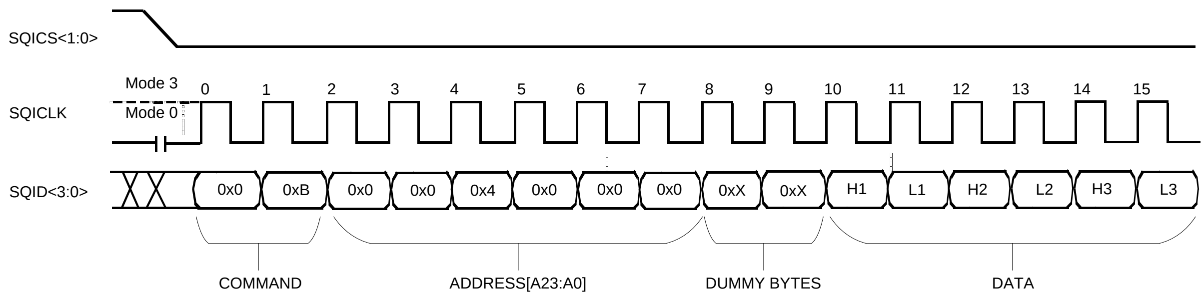
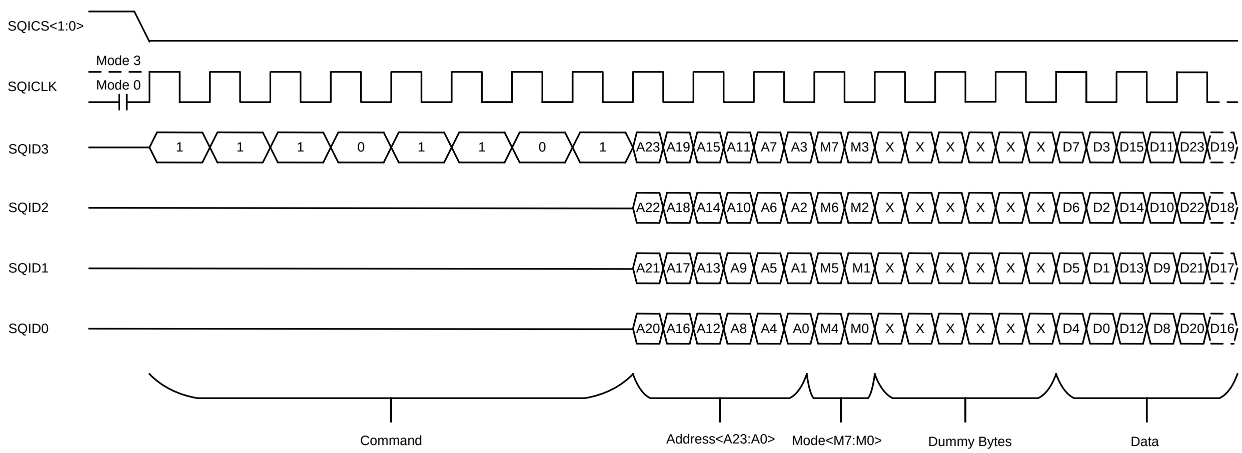


Figure 36-3. High-Speed Read Quad Lane Sequence in DDR Mode



36.5.2. SPI Mode 0 or Mode 3

The SQI peripheral supports the two most prominent SPI data flow modes, Mode 0 and Mode 3, which are controlled by the CPOL bit (CFG<4>) and the CPHA bit (CFG<3>). For additional details refer to SQI Data Flow Modes.

Mode 0 and Mode 3 are typical SPI modes of operation, which are differentiated by the CPOL and CPHA bit settings. When CPOL and CPHA are set to '0', the SQI module operates in Mode 0. When these two bits are set to '1', the SQI module operates in Mode 3.

As shown in the following figures, the primary difference between Mode 0 and Mode 3 concerns the state of SQI clock when the SQI controller is in Idle mode (i.e., no transfers are in progress). In Mode 0, the SQI clock stays low during Idle mode and in Mode 3, it stays high (provides a better clock edge entering active mode). In Mode 0, the SQI clock is held low at the start and the end of the SQI transfer cycle, whereas in Mode 3, the SQI clock is held high at the start and end of the transfer cycle. In both modes, the SQI data input in sampled on the rising edge of the SQI clock, and the SQI data output is clocked on the falling edge of the SQI clock.

Figure 36-4. Mode 0 (CPHA = 0, CPOL = 0)

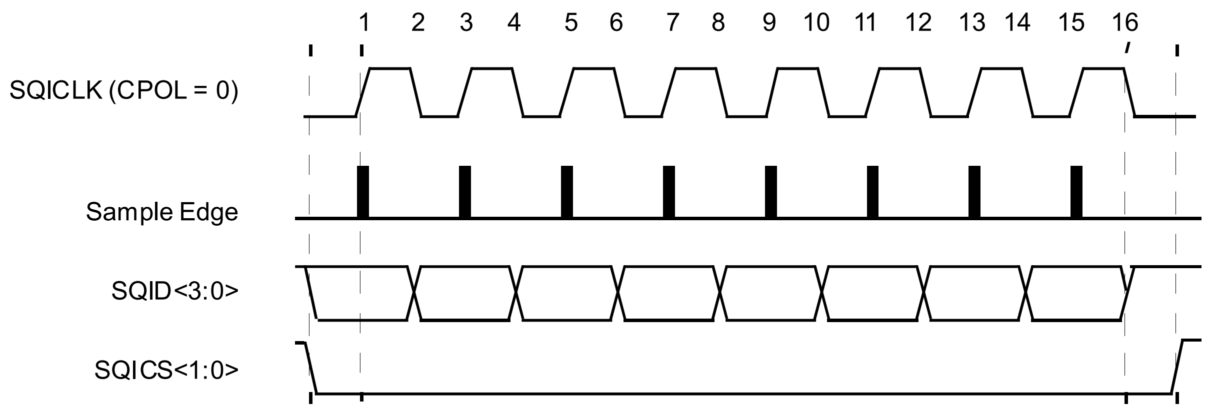
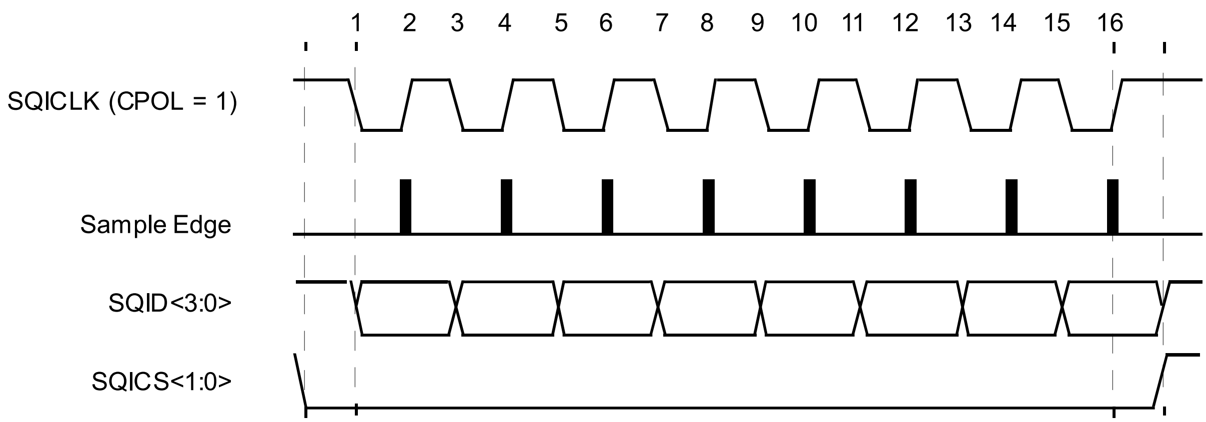


Figure 36-5. Mode 3 (CPHA = 1, CPOL = 1)



36.5.2.1. SQI Transfer Modes

The SQI module operates in three transfer modes: DMA, PIO, and XIP. As mentioned earlier, DMA and PIO modes are typically used to transfer data, whereas XIP mode is used to execute the code out of the attached serial Flash memory space. DMA mode uses the internal DMA engine and linked-list type of structures to transfer data between source and destination memory spaces, making it a little more automated, resulting in less software overhead and CPU intervention. DMA mode can be considered as a high throughput data transfer mode. In PIO mode, the CPU can access the contents of the attached serial memory device through the SQI transmit data and receive data registers with status and interrupt flag assistance.

36.5.2.2. SQI Data Flow Modes

The SQI module is a synchronous SPI-compatible serial port, which can operate in typical SPI modes 0 and 3 (specified by the CPOL bit (CFG<4>) and the CPHA bit (CFG<3>)).

36.5.2.3. DMA, PIO or XIP Transfer Modes

The SQI module operates in three transfer modes: DMA, PIO and XIP. For additional details on the modes of transfer, refer to SQI Transfer Modes.

36.5.2.4. Flash Status Check

The SQI module supports a hardware-based Flash status check in DMA, PIO, and XIP modes, thereby reducing the burden on software. The status check option is user-configurable, and checks the status of the Flash by automatically reading the Flash status register and checks the RDY/BUSY status flag. If a specific Flash command (i.e., programming or erase type of operations) requires a status check, the user can use this feature, and when enabled, the SQI module will not proceed with the next command in the queue until the status check on the current command returns the RDY state.

In DMA and PIO modes, the SQI module uses the SCHECK bit (CON<24>) in DMA mode and the DDRCMD bit (XCON1<24>) in PIO mode in combination with the MEMSTAT register to handle the Flash status check operation.

In XIP mode, the INIT1SCHECK (XCON3<28>) and INIT2SCHECK (XCON4<28>) bits in combination with different commands in the same registers facilitates the Flash status check operation.

Note: It is mandatory to set RX_CMD_THRES = 1" for automatic status check to work (even if not doing an actual read from external memory).

36.5.2.5. Tap Delays at High Interface Speeds

The SQI module provides a tap control register, TAPCON, which can be used to adjust the timing between the SQICLK and SQID0-SQID3 signals to compensate for the data delays control the setup and hold times at higher speeds. Each tap adds a certain delay on the signal that can be used to control the clock and data relationship, and may be useful in certain instances to compensate for the PCB routing delays. Refer to the *Electrical Characteristics* for the exact delay that each tap element adds.

36.6. Register Summary

For descriptions and definitions of both Register and bitfield properties, refer to Register Properties.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRLA	31:24								
		23:16								
		15:8								
		7:0		RUNSTDBY						
0x04 ... 0x0F	Reserved									
0x10	INTENCLR	31:24								
		23:16								
		15:8								
		7:0								SQI
0x14	INTENSET	31:24								
		23:16								
		15:8								
		7:0								SQI
0x18	INTFLAG	31:24								
		23:16								
		15:8								
		7:0								SQI
0x1C ... 0x1F	Reserved									
0x20	SYNCBUSY	31:24								
		23:16								
		15:8								
		7:0								SWRST
0x24 ... 0xFF	Reserved									
0x0100	XCON1	31:24			SDRCMD	DDRDATA	DDRDUMMY	DDRMODE	DDRADDR	DDRCMD
		23:16	DUMMYBYTES[2:0]			ADDRBYTES[2:0]			READOPCODE[7:6]	
		15:8	READOPCODE[5:0]						TYPEDATA[1:0]	
		7:0	TYPEDUMMY[1:0]		TYPEMODE[1:0]		TYPEADDR[1:0]		TYPECMD[1:0]	
0x0104	XCON2	31:24								
		23:16								
		15:8					DEVSEL[1:0]		MODEBYTES[1:0]	
		7:0	MODECODE[7:0]							
0x0108	CFG	31:24					CSEN[3:0]			
		23:16	SQIEN		DATAEN[1:0]		CONBUFRST	RXBUFRST	TXBUFRST	RESET
		15:8		Reserved[1:0]		BURSTEN	Reserved	HOLD	WP	Reserved[2]
		7:0	Reserved[1:0]		LSBF	CPOL	CPHA	MODE[2:0]		
0x010C	CON	31:24							Reserved	SCHECK
		23:16	DDRMODE	DASSERT	DEVSEL[1:0]		LANEMODE[1:0]		CMDINIT[1:0]	
		15:8	TXRXCOUNT[15:8]							
		7:0	TXRXCOUNT[7:0]							
0x0110	CLKCON	31:24								
		23:16						CLKDIV[10:8]		
		15:8	CLKDIV[7:0]							
		7:0							STABLE	EN
0x0114	CMDTHR	31:24								
		23:16								
		15:8			TXCMDTHR[5:0]					
		7:0			RXCMDTHR[5:0]					
0x0118	INTTHR	31:24								
		23:16								
		15:8			TXINTTHR[5:0]					
		7:0			RXINTTHR[5:0]					

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x011C	INTEN	31:24								
		23:16								
		15:8					DMAEIE	PKTCOMPIE	BDDONEIE	CONTHRIE
		7:0	CONEMPTYIE	CONFULLIE	RXTHRIE	RXFULLIE	RXEMPTYIE	TXTHRIE	TXFULLIE	TXEMPTYIE
0x0120	INTSTAT	31:24								
		23:16								
		15:8					DMAEIF	PKTCOMPIF	BDDONEIF	CONTHRIF
		7:0	CONEMPTYIF	CONFULLIF	RXTHRIF	RXFULLIF	RXEMPTYIF	TXTHRIF	TXFULLIF	TXEMPTYIF
0x0124	TXDATA	31:24	TXDATA[31:24]							
		23:16	TXDATA[23:16]							
		15:8	TXDATA[15:8]							
		7:0	TXDATA[7:0]							
0x0128	RXDATA	31:24	RXDATA[31:24]							
		23:16	RXDATA[23:16]							
		15:8	RXDATA[15:8]							
		7:0	RXDATA[7:0]							
0x012C	STAT1	31:24								
		23:16			TXBUFFREE[5:0]					
		15:8								
		7:0			RXBUFCNT[5:0]					
0x0130	STAT2	31:24								
		23:16							CMDSTAT[1:0]	
		15:8						CONAVAIL[3:1]		
		7:0	CONAVAIL[0]	SQID3	SQID2	SQID1	SQID0		RXUN	TXOV
0x0134	BDCON	31:24								
		23:16								
		15:8								
		7:0						START	POLLEN	DMAEN
0x0138	BDCURADD	31:24	BDCURRADDR[31:24]							
		23:16	BDCURRADDR[23:16]							
		15:8	BDCURRADDR[15:8]							
		7:0	BDCURRADDR[7:0]							
0x013C ... 0x013F	Reserved									
0x0140	BDBASEADD	31:24	BDADDR[31:24]							
		23:16	BDADDR[23:16]							
		15:8	BDADDR[15:8]							
		7:0	BDADDR[7:0]							
0x0144	BDSTAT	31:24								
		23:16			BDSTATE[3:0]				DMASTART	DMAACTV
		15:8	BDCON[15:8]							
		7:0	BDCON[7:0]							
0x0148	BDPOLLCON	31:24								
		23:16								
		15:8	POLLCON[15:8]							
		7:0	POLLCON[7:0]							
0x014C	BDTXDSTAT	31:24				TXSTATE[3:0]				
		23:16				TXBUFCNT[4:0]				
		15:8								
		7:0	TXCURBUFLEN[7:0]							
0x0150	BDRXDSTAT	31:24				RXSTATE[3:0]				
		23:16				RXBUFCNT[4:0]				
		15:8								
		7:0	RXCURBUFLEN[7:0]							
0x0154	THR	31:24								
		23:16								
		15:8								
		7:0					THRES[3:0]			

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0158	INTSIGEN	31:24								
		23:16								
		15:8					DMAEISE	PKTDONEISE	BDDONEISE	CONTHRISE
		7:0	CONEMPTYISE	CONFULLISE	RXTHRISE	RXFULLISE	RXEMPTYISE	TXTHRISE	TXFULLISE	TXEMPTYISE
0x015C	TAPCON	31:24			DDRCLKINDLY[5:0]					
		23:16	SDRDATINDLY[3:0]				DDRDATINDLY[3:0]			
		15:8			SDRCLKINDLY[5:0]					
		7:0	DATAOUTDLY[3:0]				CLKOUTDLY[3:0]			
0x0160	MEMSTAT	31:24								
		23:16				STATPOS	STATTYPE[1:0]		STATBYTES[1:0]	
		15:8	STATCMD[15:8]							
		7:0	STATCMD[7:0]							
0x0164	XCON3	31:24				INIT1SCHECK	INIT1COUNT[1:0]		INIT1TYPE[1:0]	
		23:16	INIT1CMD3[7:0]							
		15:8	INIT1CMD2[7:0]							
		7:0	INIT1CMD1[7:0]							
0x0168	XCON4	31:24				INIT2SCHECK	INIT2COUNT[1:0]		INIT2TYPE[1:0]	
		23:16	INIT2CMD3[7:0]							
		15:8	INIT2CMD2[7:0]							
		7:0	INIT2CMD1[7:0]							

36.6.1. Control A Register

Name: CTRLA
Offset: 0x000
Reset: 0x000
Property: PAC Write Protection

Table 36-1. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

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

Bit 6 – RUNSTDBY Run in Standby

Value	Description
0	Module is disabled in Standby Sleep mode
1	Module continues to run in Standby Sleep mode

36.6.2. Interrupt Enable Clear Register

Name: INTENCLR
Offset: 0x010
Reset: 0x000
Property: PAC Write Protection

Table 36-2. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								SQI
Access								R/K
Reset								0

Bit 0 – SQI SQI Interrupt Enable Clear
Writing a '1' to this field clears the interrupt enable.
When read, the value return reflects the state of the enable as denoted below.

Value	Description
0	Interrupt disabled
1	Interrupt enabled

36.6.3. Interrupt Enable Set Register

Name: INTENSET
Offset: 0x014
Reset: 0x000
Property: PAC Write Protection

Table 36-3. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								SQI
Access								R/S
Reset								0

Bit 0 – SQI SQI Interrupt Enable Set
Writing a '1' to this field clears the interrupt enable.
When read, the value return reflects the state of the enable as denoted below.

Value	Description
0	Interrupt disabled
1	Interrupt enabled

36.6.4. Interrupt Status Flag Register

Name: INTFLAG
Offset: 0x018
Reset: 0x000
Property: PAC Write Protection

Note: Interrupt flags must be cleared and then read back to confirm the clear before exiting the ISR to avoid double interrupts.

Table 36-4. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								SQI
Access								R/K
Reset								0

Bit 0 – SQI SQI Interrupt Status Flag

Read value reflects the state of the interrupt flag. Do not use the interrupt flag and associated mask registers if the EIP already provides similar controls for its interrupts. This feature is design for interrupts created in the SIB or EIP which do not have mask or flag bits.

Write to ‘1’ to clear the flag.

36.6.5. Synchronization Busy Register

Name: SYNCBUSY
Offset: 0x020
Reset: 0x000
Property: PAC Write Protection

Table 36-5. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

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

Bit 0 – SWRST Software Reset Synchronization Busy

Value	Description
0	SWRST synchronization is not busy
1	SWRST synchronization is busy

36.6.6. SQI XIP CONTROL REGISTER 1

Name: XCON1
Offset: 0x100
Reset: 0x0000

Table 36-6. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
			SDRCMD	DDRDATA	DDRDUMMY	DDRMODE	DDRADDR	DDRCMD
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
	DUMMYBYTES[2:0]			ADDRBYTES[2:0]			READOPCODE[7:6]	
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	15	14	13	12	11	10	9	8
	READOPCODE[5:0]						TYPEDATA[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
	TYPEDUMMY[1:0]		TYPEMODE[1:0]		TYPEADDR[1:0]		TYPECMD[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 29 – SDRCMD SQI Command in SDR Mode bit

Value	Description
1	SQI command is in SDR mode and SQI data is in DDR mode
0	SQI command is in DDR mode and SQI data is in DDR mode

Bit 28 – DDRDATA SQI Data DDR Mode bit

Value	Description
1	SQI data bytes are transferred in DDR mode
0	SQI data bytes are transferred in SDR mode

Bit 27 – DDRDUMMY SQI Dummy DDR Mode bit

Value	Description
1	SQI dummy bytes are transferred in DDR mode
0	SQI dummy bytes are transferred in SDR mode

Bit 26 – DDRMODE SQI DDR Mode bit

Value	Description
1	SQI mode bytes are transferred in DDR mode
0	SQI mode bytes are transferred in SDR mode

Bit 25 – DDRADDR SQI Address Mode bit

Value	Description
1	SQI address bytes are transferred in DDR mode
0	SQI address bytes are transferred in SDR mode

Bit 24 – DDRCMD SQI DDR Command Mode bit

Note: When DDRCMD is set to '0', the SQI module will ignore the value in the SDRCMD bit.

Value	Description
1	SQI command bytes are transferred in DDR mode
0	SQI command bytes are transferred in SDR mode

Bits 23:21 – DUMMYBYTES[2:0] Transmit Dummy Bytes bits <2:0>

Value	Description
111	Transmit seven dummy bytes after the address bytes
011	Transmit three dummy bytes after the address bytes
010	Transmit two dummy bytes after the address bytes
001	Transmit one dummy bytes after the address bytes
000	Transmit zero dummy bytes after the address bytes

Bits 20:18 – ADDRBYTES[2:0] Address Cycle bits <2:0>

Value	Description
111	Reserved
101	Reserved
100	Four address bytes
011	Three address bytes
010	Two address bytes
001	One address bytes
000	Zero address bytes

Bits 17:10 – READOPCODE[7:0] Op code Value for Read Operation bits <7:0>

Bits 9:8 – TYPEDATA[1:0] SQI Type Data Enable bits <1:0>

Value	Description
11	Reserved
10	Quad Lane mode data is enabled
01	Dual Lane mode data is enabled
00	Single Lane mode data is enabled

Bits 7:6 – TYPEDUMMY[1:0] SQI Type Dummy Enable bits <1:0>

Value	Description
11	Reserved
10	Quad Lane mode dummy is enabled
01	Dual Lane mode dummy is enabled
00	Single Lane mode dummy is enabled

Bits 5:4 – TYPEMODE[1:0] SQI Type Mode Enable bits <1:0>

Value	Description
11	Reserved
10	Quad Lane mode is enabled
01	Dual Lane mode is enabled
00	Single Lane mode is enabled

Bits 3:2 – TYPEADDR[1:0] SQI Type Address Enable bits <1:0>

Value	Description
11	Reserved
10	Quad Lane mode address is enabled
01	Dual Lane mode address is enabled
00	Single Lane mode address is enabled

Bits 1:0 – TYPECMD[1:0] SQI Type Command Enable bits <1:0>

Value	Description
11	Reserved
10	Quad Lane mode command is enabled
01	Dual Lane mode command is enabled
00	Single Lane mode command is enabled

36.6.7. SQI XIP CONTROL REGISTER 2

Name: XCON2
Offset: 0x104
Reset: 0x0000

These bits contain the 8-bit code value for the mode bits.

Table 36-7. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
					DEVSEL[1:0]		MODEBYTES[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
	MODECODE[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 11:10 – DEVSEL[1:0] Device Select bits <1:0>

Value	Description
11	Reserved
10	Reserved
01	Device 1 is selected
00	Device 0 is selected

Bits 9:8 – MODEBYTES[1:0] Mode Byte Cycle Enable bits <1:0>

Value	Description
11	Three cycles
10	Two cycles
01	One cycle
00	Zero cycles

Bits 7:0 – MODECODE[7:0] Mode Code Value bits <7:0>

36.6.8. SQI CONFIGURATION REGISTER

Name: CFG
Offset: 0x108
Reset: 0x0000
Property: -

Table 36-8. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
					CSEN[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
	SQIEN		DATAEN[1:0]		CONBUFRST	RXBUFRST	TXBUFRST	RESET
Access	R/W		R/W	R/W	R/W/HC	R/W/HC	R/W/HC	R/W/HC
Reset	0		0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
		Reserved[1:0]		BURSTEN	Reserved	HOLD	WP	Reserved[2]
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
	Reserved[1:0]		LSBF	CPOL	CPHA	MODE[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 27:24 – CSEN[3:0] Chip Select Output Enable bits <1:0>

Value	Description
1111	Chip Select 3, 2, 1 and 0 are used
1110	Chip Select 3, 2 and 1 are used
1101	Chip Select 3, 2 and 0 are used
1100	Chip Select 3 and 2 are used
1011	Chip Select 3, 1 and 0 are used
1010	Chip Select 3 and 1 are used
1001	Chip Select 3 and 0 are used
1000	Chip Select 3 is used
0111	Chip Select 2, 1 and 0 are used
0110	Chip Select 2 and 1 are used
0101	Chip Select 2 and 0 are used
0100	Chip Select 2 is used
0011	Chip Select 1 and 0 are used
0010	Chip Select 1 is used
0001	Chip Select 0 is used
0000	No chip is selected

Bit 23 – SQIEN SQI Enable bit

Value	Description
1	SQI module is enabled
0	SQI module is disabled

Bits 21:20 – DATAEN[1:0] Data Output Enable bits <1:0>

Value	Description
11	Reserved
10	SQID3-SQID0 outputs are enabled
01	SQID1 and SQID0 data outputs are enabled
00	SQID0 data output is enabled

Bit 19 – CONBUFRST Control Buffer Reset bit

Value	Description
1	A reset pulse is generated clearing the control buffer
0	A reset pulse is not generated

Bit 18 – RXBUFRST Receive Buffer Reset bit

Value	Description
1	A reset pulse is generated clearing the receive buffer
0	A reset pulse is not generated

Bit 17 – TXBUFRST Transmit Buffer Reset bit

Value	Description
1	A reset pulse is generated clearing the transmit buffer
0	A reset pulse is not generated

Bit 16 – RESET Software Reset Select bit

Value	Description
1	A reset pulse is generated
0	A reset pulse is not generated

Bits 14:13 – Reserved[1:0]
Must be programmed as ‘0’

Bit 12 – BURSTEN Burst Configuration bit
Note: This bit must be programmed as'1'.

Value	Description
1	Burst is enabled
0	Burst is not enabled

Bit 11 – Reserved
Must be programmed as ‘0’

Bit 10 – HOLD Hold bit
In Single Lane or Dual Lane mode, this bit is used to drive the SQID3 pin, which can be used for devices with a HOLD input pin. The meaning of the values for this bit will depend on the device to which SQID3 is connected.

Bit 9 – WP Write Protect bit

In Single Lane or Dual Lane mode, this bit is used to drive the SQID2 pin, which can be used with devices with a write-protect pin. The meaning of the values for this bit will depend on the device to which SQID2 is connected.

Bits 8:6 – Reserved[2:0]

Must be programmed as '0'

Bit 5 – LSBF Data Format Select bit

Value	Description
1	LSB is sent or received first
0	MSB is sent or received first

Bit 4 – CPOL Clock Polarity Select bit

Value	Description
1	Active-low SQICLK (SQICLK high is the Idle state)
0	Active-high SQICLK (SQICLK low is the Idle state)

Bit 3 – CPHA Clock Phase Select bit

Value	Description
1	SQICLK starts toggling at the start of the first data bit
0	SQICLK starts toggling at the middle of the first data bit

Bits 2:0 – MODE[2:0] Mode Select bits <2:0>

Value	Description
111	Reserved
100	Reserved
011	XIP mode is selected (when this mode is entered, the module behaves as if executing in place (XIP), but uses the register data to control timing)
010	DMA mode is selected
001	CPU mode is selected (the module is controlled by the CPU in PIO mode. This mode is entered when leaving Boot or XIP mode)
000	Reserved

36.6.9. SQI CONTROL REGISTER

Name: CON
Offset: 0x10C
Reset: 0x0000
Property: -

If it is Transmit, commands are initiated based on a write to the transmit register or the contents of TX buffer. If CMDINIT is Receive, commands are initiated based on reads to the read register or RX buffer availability.

These bits specify the total number of bytes to transmit or received (based on CMDINIT).

Table 36-9. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
							Reserved	SCHECK
Access							R	RW
Reset							0	0

Bit	23	22	21	20	19	18	17	16
	DDRMODE	DASSERT	DEVSEL[1:0]		LANEMODE[1:0]		CMDINIT[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	15	14	13	12	11	10	9	8
	TXRXCOUNT[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
	TXRXCOUNT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 25 – Reserved

Must be programmed as '0'

Bit 24 – SCHECK Flash Status Check bit

Note: When this bit is set to '1', the SQI module uses the MEMSTAT register to control the status check command process.

Value	Description
1	Check the status of the Flash
0	Do not check the status of the Flash

Bit 23 – DDRMODE Double Data Rate Mode bit

Value	Description
1	Set the SQI transfers to DDR mode
0	Set the SQI transfers to SDR mode

Bit 22 – DASSERT Chip Select Assert bit

Value	Description
1	Chip Select is deasserted after transmission or reception of the specified number of bytes
0	Chip Select is not deasserted after transmission or reception of the specified number of bytes

Bits 21:20 – DEVSEL[1:0] SQI Device Select bits <1:0>

Value	Description
11	Reserved
10	Reserved
01	Select Device 1
00	Select Device 0

Bits 19:18 – LANEMODE[1:0] SQI Lane Mode Select bits <1:0>

Value	Description
11	Reserved
10	Quad Lane mode
01	Dual Lane mode
00	Single Lane mode

Bits 17:16 – CMDINIT[1:0] Command Initiation Mode Select bits <1:0>

If it is Transmit, commands are initiated based on a write to the transmit register or the contents of Tx buffer. If CMDINIT is Receive, commands are initiated based on reads to the read register or Rx buffer availability.

Value	Description
11	Reserved
10	Receive
01	Transmit
00	Idle

Bits 15:0 – TXRXCOUNT[15:0] Transmit/Receive Count bits <15:0>

These bits specify the total number of bytes to transmit or received (based on CMDINIT).

36.6.10. SQI CLOCK CONTROL REGISTER

Name: CLKCON
Offset: 0x110
Reset: 0x0000
Property: -

Table 36-10. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
						CLKDIV[10:8]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	15	14	13	12	11	10	9	8
	CLKDIV[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
							STABLE	EN
Access							R	R/W
Reset							0	0

Bits 18:8 – CLKDIV[10:0] SQI Clock Tsqi Frequency Select bit <10:0>

Note: Refer to the *Electrical Characteristics* for the maximum clock frequency specifications.

Setting these bits to ‘00000000’ specifies the highest frequency of the SQI clock.

Value	Description
10000000000	Base clock Tbc is divided by 2048
01000000000	Base clock Tbc is divided by 1024
00100000000	Base clock Tbc is divided by 512
00010000000	Base clock Tbc is divided by 256
00001000000	Base clock Tbc is divided by 128
00000100000	Base clock Tbc is divided by 64
00000010000	Base clock Tbc is divided by 32
00000001000	Base clock Tbc is divided by 16

Value	Description
00000000100	Base clock Tbc is divided by 8
00000000010	Base clock Tbc is divided by 4
00000000001	Base clock Tbc is divided by 2
00000000000	Base clock Tbc

Bit 1 – STABLE Tsqi Clock Stable Select bit

This bit is set to '1' when the SQI clock, TSQI, is stable after writing a '1' to the EN bit.

Value	Description
1	Tsqi clock is stable
0	Tsqi clock is not stable

Bit 0 – EN Tsqi Clock Enable Select bit

When clock oscillation is stable, the SQI module will set the STABLE bit to '1'.

Value	Description
1	Enable the SQI clock (Tsqi) (when clock oscillation is stable, the SQI module sets the STABLE bit to ‘1’)
0	Disable the SQI clock (Tsqi) (the SQI module should stop its clock to enter a low power state); SFRs can still be accessed, as they use PBCLK5

36.6.11. SQI COMMAND THRESHOLD REGISTER

Name: CMDTHR
Offset: 0x114
Reset: 0x0000
Property: -

Table 36-11. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
			TXCMDTHR[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
			RXCMDTHR[5:0]					
Access			RW	RW	RW	RW	RW	RW
Reset			0	0	0	0	0	0

Bits 13:8 – TXCMDTHR[5:0] Transmit Command Threshold bits <5:0>

In transmit initiation mode, the SQI module performs a transmit operation when transmit command threshold bytes are present in the Tx buffer. These bits should usually be set to '1' for normal Flash commands, and set to a higher value for page programming. For 16-bit mode, the value should be a multiple of 2.

Bits 5:0 – RXCMDTHR[5:0] Receive Command Threshold bits <5:0>

In receive initiation mode, the SQI module attempts to perform receive operations to fetch the receive command threshold number of bytes in the receive buffer. If space for these bytes is not present in the buffer, the SQI will not initiate a transfer. For 16-bit mode, the value should be a multiple of 2.

If software performs any reads, thereby reducing the buffer count, hardware would initiate a receive transfer to make the buffer count equal to the value in these bits. If software would not like any more words latched into the buffer, command initiation mode needs to be changed to Idle before any buffer reads by software.

In the case of Boot/xIP mode, the SQI module will use the System Bus burst size, instead of the receive command threshold value.

Note:

These bits should only be programmed when a receive is not active (i.e., during Idle mode or a transmit).

36.6.12. SQI INTERRUPT THRESHOLD REGISTER

Name: INTTHR
Offset: 0x118
Reset: 0x0000
Property: -

Table 36-12. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
			TXINTTHR[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
			RXINTTHR[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bits 13:8 – TXINTTHR[5:0] Transmit Interrupt Threshold bits <5:0>
A transmit interrupt is set when the transmit buffer has more space than the set number of bytes. For 16-bit mode, the value should be a multiple of 2.

Bits 5:0 – RXINTTHR[5:0] Receive Interrupt Threshold bits <5:0>
A receive interrupt is set when the receive buffer count is larger than or equal to the set number of bytes. For 16-bit mode, the value should be multiple of 2.

36.6.13. SQI INTERRUPT ENABLE REGISTER

Name: INTEN
Offset: 0x11C
Reset: 0x0000
Property: -

Table 36-13. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
					DMAEIE	PKTCOMPIE	BDDONEIE	CONTHRIE
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CONEMPTYIE	CONFULLIE	RXTHRIE	RXFULLIE	RXEMPTYIE	TXTHRIE	TXFULLIE	TXEMPTYIE
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 11 – DMAEIE DMA Bus Error Interrupt Enable bit

Value	Description
1	Interrupt is enabled
0	Interrupt is disabled

Bit 10 – PKTCOMPIE DMA Buffer Descriptor Packet Complete Interrupt Enable bit

Value	Description
1	Interrupt is enabled
0	Interrupt is disabled

Bit 9 – BDDONEIE DMA Buffer Descriptor Done Interrupt Enable bit

Value	Description
1	Interrupt is enabled
0	Interrupt is disabled

Bit 8 – CONTHRIE Control Buffer Threshold Interrupt Enable bit

Value	Description
1	Interrupt is enabled

Value	Description
0	Interrupt is disabled

Bit 7 – CONEMPTYIE Control Buffer Empty Interrupt Enable bit

Value	Description
1	Interrupt is enabled
0	Interrupt is disabled

Bit 6 – CONFULLIE Control Buffer Full Interrupt Enable bit
This bit enables an interrupt when the receive buffer is full.

Value	Description
1	Interrupt is enabled
0	Interrupt is disabled

Bit 5 – RXTHRIE Receive Buffer Threshold Interrupt Enable bit

Value	Description
1	Interrupt is enabled
0	Interrupt is disabled

Bit 4 – RXFULLIE Receive Buffer Full Interrupt Enable bit

Value	Description
1	Interrupt is enabled
0	Interrupt is disabled

Bit 3 – RXEMPTYIE Receive Buffer Empty Interrupt Enable bit

Value	Description
1	Interrupt is enabled
0	Interrupt is disabled

Bit 2 – TXTHRIE Transmit Threshold Interrupt Enable bit

Value	Description
1	Interrupt is enabled
0	Interrupt is disabled

Bit 1 – TXFULLIE Transmit Buffer Full Interrupt Enable bit

Value	Description
1	Interrupt is enabled
0	Interrupt is disabled

Bit 0 – TXEMPTYIE Transmit Buffer Empty Interrupt Enable bit

Value	Description
1	Interrupt is enabled
0	Interrupt is disabled

36.6.14. SQI INTERRUPT STATUS REGISTER

Name: INTSTAT
Offset: 0x120
Reset: 0x0000
Property: -

Note: The bits in the register are cleared by writing a '1' to the corresponding bit position.

Table 36-14. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
					DMAEIF	PKTCOMPIF	BDDONEIF	CONTHRIF
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CONEMPTYIF	CONFULLIF	RXTHRIF	RXFULLIF	RXEMPTYIF	TXTHRIF	TXFULLIF	TXEMPTYIF
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	0	1	0	1	1	0	0

Bit 11 – DMAEIF DMA Bus Error Interrupt Flag bit

Value	Description
1	DMA bus error has occurred
0	DMA bus error has not occurred

Bit 10 – PKTCOMPIF DMA Buffer Descriptor Processor Packet Completion Interrupt Flag bit

Value	Description
1	DMA BD packet is complete
0	DMA BD packet is in progress

Bit 9 – BDDONEIF DMA Buffer Descriptor Done Interrupt Flag bit

Value	Description
1	DMA BD process is done
0	DMA BD process is in progress

Bit 8 – CONTHRIF Control Buffer Threshold Interrupt Flag bit

Value	Description
1	The control buffer has more than THRES words of space available
0	The control buffer has less than THRES words of space available

Bit 7 – CONEMPTYIF Control Buffer Empty Interrupt Flag bit

Value	Description
1	Control buffer is empty
0	Control buffer is not empty

Bit 6 – CONFULLIF Control Buffer Full Interrupt Flag bit

Value	Description
1	Control buffer is full
0	Control buffer is not full

Bit 5 – RXTHRIF Receive Buffer Threshold Interrupt Flag bit

Note: In the case of Boot/xIP mode, the POR value of the receive buffer threshold is zero. Therefore, this bit will be set to a '1',immediately after a POR until a read request on the System Bus bus is received.

Value	Description
1	Receive buffer has more than RXINTTHR words of space available
0	Receive buffer has less than RXINTTHR words of space available

Bit 4 – RXFULLIF Receive Buffer Full Interrupt Flag bit

Value	Description
1	Receive buffer is full
0	Receive buffer is not full

Bit 3 – RXEMPTYIF Receive Buffer Empty Interrupt Flag bit

Value	Description
1	Receive buffer is empty
0	Receive buffer is not empty

Bit 2 – TXTHRIF Transmit Buffer Threshold Interrupt Flag bit

Value	Description
1	Transmit buffer has more than TXINTTHR words of space available
0	Transmit buffer has less than TXINTTHR words of space available

Bit 1 – TXFULLIF Transmit Buffer Full Interrupt Flag bit

Value	Description
1	The transmit buffer is full
0	The transmit buffer is not full

Bit 0 – TXEMPTYIF Transmit Buffer Empty Interrupt Flag bit

Value	Description
1	The transmit buffer is empty
0	The transmit buffer has content

36.6.15. SQI TRANSMIT DATA BUFFER REGISTER

Name: TXDATA
Offset: 0x124
Reset: 0x0000
Property: -

Table 36-15. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	TXDATA[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
	TXDATA[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
	TXDATA[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
	TXDATA[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 – TXDATA[31:0] Transmit Command Data bits <31:0>

Data is loaded into this register before being transmitted. Just prior to the beginning of a data transfer, the data in TxDATA is loaded into the shift register (SFDR).

Multiple writes to TxDATA can occur even while a transfer is already in progress. There can be a maximum of eight commands that can be queued.

36.6.16. SQI RECEIVE DATA BUFFER REGISTER

Name: RXDATA
Offset: 0x128
Reset: 0x0000
Property: -

Table 36-16. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	RXDATA[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
	RXDATA[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
	RXDATA[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
	RXDATA[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – RXDATA[31:0] Receive Data Buffer bits <31:0>
At the end of a data transfer, the data in the shift register is loaded into the RxDATA register. This register works like a buffer. The depth of the receive buffer is eight words.

36.6.17. SQI STATUS REGISTER 1

Name: STAT1
Offset: 0x12C
Reset: 0x0000
Property: -

Table 36-17. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
			TXBUFFREE[5:0]					
Access			R	R	R	R	R	R
Reset			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
			RXBUFCNT[5:0]					
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bits 21:16 – TXBUFFREE[5:0] Transmit buffer Available Word Space bits <5:0>

Bits 5:0 – RXBUFCNT[5:0] Number of words of read data in the buffer <5:0>

36.6.18. SQI STATUS REGISTER 2

Name: STAT2
Offset: 0x130
Reset: 0x0000
Property: -

Table 36-18. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
							CMDSTAT[1:0]	
Access							R	R
Reset							0	0

Bit	15	14	13	12	11	10	9	8
						CONAVAIL[3:1]		
Access						R	R	R
Reset						0	0	0

Bit	7	6	5	4	3	2	1	0
	CONAVAIL[0]	SQID3	SQID2	SQID1	SQID0		RXUN	TXOV
Access	R	R	R	R	R		R	R
Reset	0	0	0	0	0		0	0

Bits 17:16 – CMDSTAT[1:0] Current Command Status bits <1:0>
These bits indicate the current command status.

Value	Description
`11`	Reserved
`10`	Receive
`01`	Transmit
`00`	Idle

Bits 10:7 – CONAVAIL[3:0] Control buffer Space Available bits <3:0>
These bits indicate the available control wordspace.

Value	Description
`1000`	8 words are available
`0111`	7 words are available
`0001`	1 word is available
`0000`	No words are available

Bit 6 – SQID3 SQID3 Status bit

Value	Description
`1`	Data is present on SQID3

Value	Description
0	Data is not present on SQID3

Bit 5 – SQID2 SQID2 Status bit

Value	Description
1	Data is present on SQID2
0	Data is not present on SQID2

Bit 4 – SQID1 SQID1 Status bit

Value	Description
1	Data is present on SQID1
0	Data is not present on SQID1

Bit 3 – SQID0 SQID0 Status bit

Value	Description
1	Data is present on SQID0
0	Data is not present on SQID0

Bit 1 – RXUN Receive buffer Underflow Status bit

Value	Description
1	Receive buffer Underflow has occurred
0	Receive buffer underflow has not occurred

Bit 0 – TXOV Transmit buffer Overflow Status bit

Value	Description
1	Transmit buffer overflow has occurred
0	Transmit buffer overflow has not occurred

36.6.19. SQI BUFFER DESCRIPTOR CONTROL REGISTER

Name: BDCON
Offset: 0x134
Reset: 0x0000
Property: -

Table 36-19. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						START	POLLEN	DMAEN
Access						R/W	R/W	R/W
Reset						0	0	0

Bit 2 – START Buffer Descriptor Processor Start bit

Value	Description
1	Start the buffer descriptor processor
0	Disable the buffer descriptor processor

Bit 1 – POLLEN Buffer Descriptor Poll Enable bit

Value	Description
1	BDP poll is enabled
0	BDP poll is not enabled

Bit 0 – DMAEN DMA Enable bit

Value	Description
1	DMA is enabled
0	DMA is disabled

36.6.20. SQI BUFFER DESCRIPTOR CURRENT ADDRESS REGISTER

Name: BDCURADD
Offset: 0x138
Reset: 0x0000
Property: -

Table 36-20. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	BDCURRADDR[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
	BDCURRADDR[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
	BDCURRADDR[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
	BDCURRADDR[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – BDCURRADDR[31:0] Current Buffer Descriptor Address bits <31:0>

36.6.21. SQI BUFFER DESCRIPTOR BASE ADDRESS REGISTER

Name: BDBASEADD
Offset: 0x140
Reset: 0x0000
Property: -

Table 36-21. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	BDADDR[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
	BDADDR[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
	BDADDR[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
	BDADDR[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – BDADDR[31:0] DMA Base Address bits <31:0>
These bits contain the physical address of the root buffer descriptor. This register should be updated only when the DMA is idle.

36.6.22. SQI BUFFER DESCRIPTOR STATUS REGISTER

Name: BDSTAT
Offset: 0x144
Reset: 0x0000
Property: -

Table 36-22. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
			BDSTATE[3:0]				DMASTART	DMAACTV
Access			R	R	R	R	R	RO
Reset			x	x	x	x	x	x

Bit	15	14	13	12	11	10	9	8
	BDCON[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	x	x	x	x	0	0	0	0

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

Bits 21:18 – BDSTATE[3:0] DMA Buffer Descriptor Processor State Status bits <3:0>
These bits return the current state of the buffer descriptor processor:

Value	Description
5	Fetched buffer descriptor is disabled
4	Descriptor is done
3	Data phase
2	Buffer descriptor is loading
1	Descriptor fetch request is pending
0	Idle

Bit 17 – DMASTART DMA Buffer Descriptor Processor Start Status bit

Value	Description
1	DMA has started
0	DMA has not started

Bit 16 – DMAACTV DMA Buffer Descriptor Processor Active Status bit

Value	Description
1	Buffer Descriptor Processor is active
0	Buffer Descriptor Processor is idle

Bits 15:0 – BDCON[15:0] DMA Buffer Descriptor Control Word bits <15:0>
These bits contain the current buffer descriptor control word.

36.6.23. SQI BUFFER DESCRIPTOR POLL CONTROL REGISTER

Name: BDPOLLCON
Offset: 0x148
Reset: 0x0000
Property: -

Table 36-23. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
	POLLCON[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
	POLLCON[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 – POLLCON[15:0] Buffer Descriptor Processor Poll Status bits <15:0>
These bits indicate the number of cycles the BDP would wait before refetching the descriptor control word if the previous descriptor fetched was disabled.

36.6.24. SQI BUFFER DESCRIPTOR DMA TRANSMIT STATUS REGISTER

Name: BDTXDSTAT
Offset: 0x14C
Reset: 0x0000
Property: -

Table 36-24. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
				TXSTATE[3:0]				
Access				R	R	R	R	
Reset				x	x	x	x	

Bit	23	22	21	20	19	18	17	16
				TXBUFCNT[4:0]				
Access				R	R	R	R	R
Reset				x	x	x	x	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	TXCURBUFLEN[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	x	x	x	x	0	0	0	0

Bits 28:25 – TXSTATE[3:0] Current DMA Transmit State Status bits <3:0>
These bits provide information on the current DMA receive states.

Bits 20:16 – TXBUFCNT[4:0] DMA Buffer Byte Count Status bits <4:0>
These bits provide information on the internal buffer space.

Bits 7:0 – TXCURBUFLEN[7:0] Current DMA Transmit Buffer Length Status bits <7:0>
These bits provide the length of the current DMA transmit buffer.

36.6.25. SQI BUFFER DESCRIPTOR DMA RECEIVE STATUS REGISTER

Name: BDRXDSTAT
Offset: 0x150
Reset: 0x0000
Property: -

Table 36-25. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
				RXSTATE[3:0]				
Access				R	R	R	R	
Reset				x	x	x	x	

Bit	23	22	21	20	19	18	17	16
				RXBUFCNT[4:0]				
Access				R	R	R	R	R
Reset				x	x	x	x	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	RXCURBUFLEN[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	x	x	x	x	0	0	0	0

Bits 28:25 – RXSTATE[3:0] Current DMA Receive State Status bits <3:0>
These bits provide information on the current DMA receive states.

Bits 20:16 – RXBUFCNT[4:0] DMA Buffer Byte Count Status bits <4:0>
These bits provide information on the internal buffer space.

Bits 7:0 – RXCURBUFLEN[7:0] Current DMA Receive Buffer Length Status bits <7:0>
These bits provide the length of the current DMA receive buffer.

36.6.26. SQI THRESHOLD CONTROL REGISTER

Name: THR
Offset: 0x154
Reset: 0x0000
Property: -

Table 36-26. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

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

Bits 3:0 – THRES[3:0] SQI Control Threshold Value bits <3:0>

36.6.27. SQI INTERRUPT SIGNAL ENABLE REGISTER

Name: INTSIGEN
Offset: 0x158
Reset: 0x0000
Property: -

Table 36-27. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
					DMAEISE	PKTDONEISE	BDDONEISE	CONTHRISE
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CONEMPTYISE	CONFULLISE	RXTHRISE	RXFULLISE	RXEMPTYISE	TXTHRISE	TXFULLISE	TXEMPTYISE
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 11 – DMAEISE DMA Bus Error Interrupt Signal Enable bit

Value	Description
1	Interrupt signal is enabled
0	Interrupt signal is disabled

Bit 10 – PKTDONEISE Receive Error Interrupt Signal Enable bit

Value	Description
1	Interrupt signal is enabled
0	Interrupt signal is disabled

Bit 9 – BDDONEISE Transmit Error Interrupt Signal Enable bit

Value	Description
1	Interrupt signal is enabled
0	Interrupt signal is disabled

Bit 8 – CONTHRISE Control Buffer Threshold Interrupt Signal Enable bit

Value	Description
1	Interrupt signal is enabled
0	Interrupt signal is disabled

Bit 7 – CONEMPTYISE Control Buffer Empty Interrupt Signal Enable bit

Value	Description
1	Interrupt signal is enabled
0	Interrupt signal is disabled

Bit 6 – CONFULLISE Control Buffer Full Interrupt Signal Enable bit

Value	Description
1	Interrupt signal is enabled
0	Interrupt signal is disabled

Bit 5 – RXTHRISE Receive Buffer Threshold Interrupt Signal Enable bit

Value	Description
1	Interrupt signal is enabled
0	Interrupt signal is disabled

Bit 4 – RXFULLISE Receive Buffer Full Interrupt Signal Enable bit

Value	Description
1	Interrupt signal is enabled
0	Interrupt signal is disabled

Bit 3 – RXEMPTYISE Receive Buffer Empty Interrupt Signal Enable bit

Value	Description
1	Interrupt signal is enabled
0	Interrupt signal is disabled

Bit 2 – TXTHRISE Transmit Buffer Threshold Interrupt Signal Enable bit

Value	Description
1	Interrupt signal is enabled
0	Interrupt signal is disabled

Bit 1 – TXFULLISE Transmit Buffer Full Interrupt Signal Enable bit

Value	Description
1	Interrupt signal is enabled
0	Interrupt signal is disabled

Bit 0 – TXEMPTYISE Transmit Buffer Empty Interrupt Signal Enable bit

Value	Description
1	Interrupt signal is enabled
0	Interrupt signal is disabled

36.6.28. SQI TAP CONTROL REGISTER

Name: TAPCON
Offset: 0x15C
Reset: 0x0000
Property: -

Table 36-28. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
			DDRCLKINDLY[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
	SDRDATINDLY[3:0]				DDRDATINDLY[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
			SDRCLKINDLY[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
	DATAOUTDLY[3:0]				CLKOUTDLY[3:0]			
Access	R/W	R/W	R/W	R/W	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 29:24 – DDRCLKINDLY[5:0] SQI Clock Input Delay in DDR Mode bits <5:0>

These bits are used to add fractional delays to SQI Clock Input while sampling the incoming data in DDR mode.

Value	Description
111111	64 taps added on clock input
111110	63 taps added on clock input
000001	2 taps added on clock input
000000	1 tap added on clock input

Bits 23:20 – SDRDATINDLY[3:0] SQI Data Input Delay in SDR Mode bits <3:0>

These bits are used to add fractional delays to SQI Data Output while writing the data to the Flash in SDR mode.

Value	Description
1111	16 taps added on data input
1110	15 taps added on data input
0001	2 taps added on data input
0000	1 tap added on data input

Bits 19:16 – DDRDATINDLY[3:0] SQI Data Output Delay in DDR Mode bits <3:0>

These bits are used to add fractional delays to SQI Data Output while writing the data to the Flash in DDR mode.

Value	Description
1111	16 taps added on data input
1110	15 taps added on data input
0001	2 taps added on data input
0000	1 tap added on data input

Bits 13:8 – SDRCLKINDLY[5:0] SQI Clock Input Delay in SDR Mode bits <5:0>

These bits are used to add fractional delays to SQI Clock Input while sampling the incoming data in DDR mode.

Value	Description
111111	64 taps added on clock input
111110	63 taps added on clock input
000001	2 taps added on clock input
000000	1 tap added on clock input

Bits 7:4 – DATAOUTDLY[3:0] SQI Data Output Delay bits <3:0>

These bits are used to add fractional delays to SQI Data Output while writing the data to the Flash in all modes of operation.

Value	Description
1111	16 taps added on data output
1110	15 taps added on data output
0001	2 taps added on data output
0000	1 tap added on data output

Bits 3:0 – CLKOUTDLY[3:0] SQI Clock Output Delay bits <3:0>

These bits are used to add fractional delays to SQI Clock Output while writing the data to the Flash in all modes of operation.

Value	Description
1111	16 taps added on clock output
1110	15 taps added on clock output
0001	2 taps added on clock output
0000	1 tap added on clock output

36.6.29. SQI MEMORY STATUS CONTROL REGISTER

Name: MEMSTAT
Offset: 0x160
Reset: 0x0000
Property: -

Table 36-29. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
				STATPOS	STATTYPE[1:0]		STATBYTES[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
	STATCMD[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
	STATCMD[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 20 – STATPOS Status Bit Position in Flash bit

Indicates the BUSY bit position in the Flash Status register. This bit is added to support all Flash types (with BUSY bit at 0 and at 7).

Value	Description
1	BUSY bit position is bit 7 in status register
0	BUSY bit position is bit 0 in status register

Bits 19:18 – STATTYPE[1:0] Status Command Lane Mode bits <1:0>

Value	Description
11	Reserved
10	Status command and read are executed in Quad Lane mode
01	Status command and read are executed in Dual Lane mode
00	Status command and read are executed in Single Lane mode

Bits 17:16 – STATBYTES[1:0] Number of Status Bytes bits <1:0>

Value	Description
11	Reserved
10	Status command is 2 bytes long
01	Status command is 1 byte long

Value	Description
00	Reserved

Bits 15:0 – STATCMD[15:0] Status Command bits <15:0>
The status check command is written into these bits

36.6.30. SQI XIP CONTROL REGISTER 3

Name: XCON3
Offset: 0x164
Reset: 0x0000
Property: -

Note: Some Flash devices require write enable and sector unprotect commands before write/read operations and this register is useful in working with those Flash types (xIP mode only)

Table 36-30. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
				INIT1SCHECK	INIT1COUNT[1:0]		INIT1TYPE[1:0]	
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
	INIT1CMD3[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
	INIT1CMD2[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
	INIT1CMD1[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 28 – INIT1SCHECK Flash Initialization 1 Command Status Check bit

Value	Description
1	Check the status after executing the INIT1 commands
0	Do not check the status

Bits 27:26 – INIT1COUNT[1:0] Flash Initialization 1 Command Count bits <1:0>

Value	Description
11	INIT1CMD1, INIT1CMD2, and INIT1CMD3 are sent
10	INIT1CMD1 and INIT1CMD2 are sent, but INIT1CMD3 is still pending
01	INIT1CMD1 is sent, but INIT1CMD2 and INIT1CMD3 are still pending
00	No commands are sent

Bits 25:24 – INIT1TYPE[1:0] Flash Initialization 1 Command Type bits <1:0>

Value	Description
11	Reserved
10	INIT1 commands are sent in Quad Lane mode

Value	Description
01	INIT1 commands are sent in Dual Lane mode
00	INIT1 commands are sent in Single Lane mode

Bits 24:16 – INIT1CMD3[8:0] Flash Initialization Command 3 bits <7:0>
Third command of the Flash initialization.

Bits 15:8 – INIT1CMD2[7:0] Flash Initialization Command 2 bits <7:0>
Second command of the Flash initialization.

Bits 7:0 – INIT1CMD1[7:0] Flash Initialization Command 1 bits <7:0>
First command of the Flash initialization.

36.6.31. SQI XIP CONTROL REGISTER 4

Name: XCON4
Offset: 0x168
Reset: 0x0000
Property: -

Table 36-31. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
				INIT2SCHECK	INIT2COUNT[1:0]		INIT2TYPE[1:0]	
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
	INIT2CMD3[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
	INIT2CMD2[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
	INIT2CMD1[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 28 – INIT2SCHECK Flash Initialization 2 Command Status Check bit

Value	Description
1	Check the status after executing the INIT2 commands
0	Do not check the status

Bits 27:26 – INIT2COUNT[1:0] Flash Initialization 2 Command Count bits <1:0>

Value	Description
11	INIT2CMD1, INIT2CMD2, and INIT2CMD3 are sent
10	INIT2CMD1 and INIT2CMD2 are sent, but INIT2CMD3 is still pending
01	INIT2CMD1 is sent, but INIT2CMD2 and INIT2CMD3 are still pending
00	No commands are sent

Bits 25:24 – INIT2TYPE[1:0] Flash Initialization 2 Command Type bits <1:0>

Value	Description
11	Reserved
10	INIT2 commands are sent in Quad Lane mode
01	INIT2 commands are sent in Dual Lane mode
00	INIT2 commands are sent in Single Lane mode

Bits 24:16 – INIT2CMD3[8:0] Flash Initialization Command 3 bits <7:0>
Third command of the Flash initialization.

Bits 15:8 – INIT2CMD2[7:0] Flash Initialization Command 2 bits <7:0>
Second command of the Flash initialization.

Bits 7:0 – INIT2CMD1[7:0] Flash Initialization Command 1 bits <7:0>
First command of the Flash initialization

37. Universal Serial Bus Hi-Speed (USBHS)

37.1. Overview

The Universal Serial Bus Hi-Speed (USBHS) module contains analog and digital components to provide a USB 2.0 embedded Host or device implementation with a minimum of external components. The PIC32CK SG/GC family of devices contains one High-Speed USB module.

The module supports Hi-Speed, Full-Speed, or Low- Speed (in Host Mode only). This module in Host mode is intended for use as an embedded Host and therefore does not implement a Universal Host Controller Interface (UHCI), or Open Host Controller Interface (OHCI) controller.

The USBHS module consists of the RAM controller, packet encode/decode, UTM synchronization, endpoint control, a dedicated USB DMA controller, pull-up and pull-down resistors, and the register interface. A diagram of the PIC32 USB module is illustrated in the Block Diagram.

37.2. Features

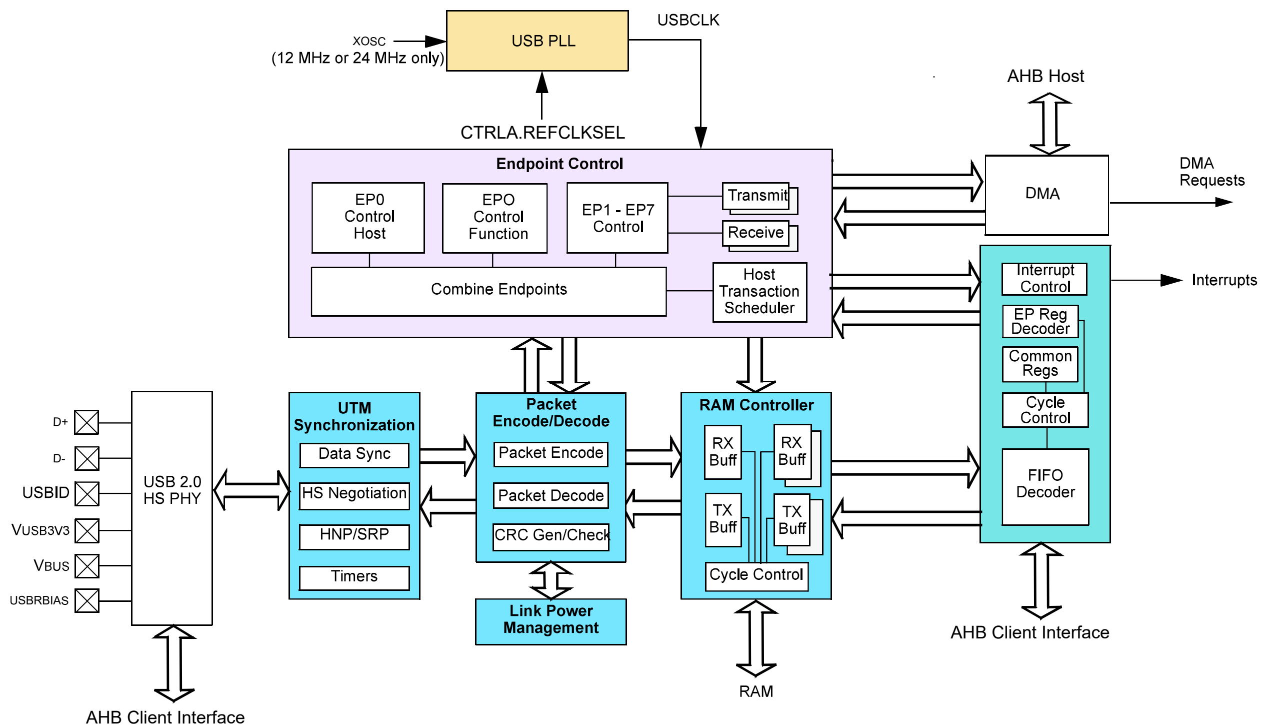
The USBHS module includes:

- USB Hi-Speed, Full-Speed, and Low-Speed (in Host mode only)
- USB support with one or more Hi-Speed, Full-Speed, or Low-Speed device
- Integrated internal USB signaling resistors
- Integrated analog comparators for VBUS monitoring
- Integrated USB transceiver
- Transaction handshaking performed by hardware
- Integrated 8-channel DMA to access system RAM and Flash
- Seven transmit endpoints and seven receive endpoints, in addition to Endpoint 0
- Session Request Protocol (SRP) and Host Negotiation Protocol (HNP) support
- Suspend and resume signaling support
- Dynamic FIFO sizing
- Integrated RAM for the FIFOs, eliminating the need for system RAM for the FIFOs
- Link-power management support

Notes:

1. The implementation and use of the USB specifications, as well as other third party specifications or technologies, may require licensing; including, but not limited to, USB Implementers Forum, Inc. (also referred to as USB-IF). The user is fully responsible for investigating and satisfying any applicable licensing obligations.
2. If the USB module is used, the Primary Oscillator (XOSC) is limited to either 12 MHz or 24 MHz.
3. External 270 (+/-1%) ohm Resister is required on the USBRBIASx pin to AVSS.

37.3. Block Diagram



37.4. Signal Description

Signal Name	Type	Description
D0-	Analog	USB0 D- differential data line
D0+	Analog	USB0 D+ differential data line
USBID0	Digital Input	USB0 ID Detect
VUSB3V3	Power	USB 3.3v internal transceiver supply. This pin should be grounded when the USB feature is not used.
VBUS0	Analog (VBUS Pwr Monitor)	USB0 Bus Power Monitor
USBRBIAS0	Analog	270Ω ±1% resistor to AGND

37.5. Peripheral Dependencies

Peripheral Name	Base Address	NVIC IRQ Index: Source	MCLK AHBx/APBx Clock Enable Mask Bit	Power Domain
USBHS	0x4600_0000	154: USBHS	MCLK.CLKMSK0[20]	VDDREG / VUSB3V3

37.6. Functional Description

Important: Before attempting to initialize and enable the USB, the user must first enable the internal USB regulator, SUPC.VREGCTRL.AVREGEN, and then wait a minimum of 55µs for internal power to stabilize before reading or writing any of the USB registers.

The universal serial bus (USB) is an industry standard serial bus used to facilitate communication between a host and devices within a hierarchical system. In the PIC32CK SG/GC family of MCUs, the Hi-Speed USB module is connected to the ARM M-33 core via the advanced high-performance bus (AHB). The Hi-Speed USB module has a dedicated DMA host to transfer data between the USB port and system memory.

The Hi-Speed USB module has two main modes of operation:

- Device mode
- Embedded Host mode

37.6.1. Device Mode

In Device mode, the Hi-Speed USB module encodes, decodes, checks, and directs all USB packets sent and received. IN transactions are handled through the device's TX FIFOs, OUT transactions are handled through its RX FIFOs. Control, Bulk, Isochronous and Interrupt transactions are also supported.

37.6.2. Embedded Host Mode

In Embedded Host mode, the way in which the Hi-Speed USB module behaves depends on whether it is linked up for point-to-point communications with another USB system or whether it is attached to a hub. When attached directly to a USB system operating as a Device, the module offers the range of capabilities needed to act as the host in point-to-point communications with this USB system. When attached through a USB hub, the Hi-Speed USB module can perform the functions required to act as the host for multiple USB Devices simultaneously.

When operating in Embedded Host mode and used for point-to-point communications with a single other USB device (which can be Hi-Speed, Full-Speed, or Low-Speed), the Hi-Speed USB module can support Control, Bulk, Isochronous or Interrupt transactions. IN transactions are handled through the RX FIFOs, OUT transactions are handled through the TX FIFOs. As well as encoding, decoding and checking the USB packets sent and received, the module will also automatically schedule Isochronous endpoints and Interrupt endpoints to perform one transaction every 'n' frames/micro-frames (or up to three transactions if the high-bandwidth option is selected), where 'n' represents the polling interval that has been programmed for the endpoint. The remaining bus bandwidth is shared equally between the Control and Bulk endpoints.

37.6.3. Embedded Host Mode Through a Hub

When attached to a single or multiple USB devices through a USB hub, the Hi-Speed USB module continues to offer the facilities previously mentioned, but it needs to be further configured for the following functional overlays:

- The function address of the target device
- The operating speed of the target device (so that the appropriate speed conversion can be carried out)
- If the target device is a Full-Speed or Low-Speed device that is accessed through a Hi-Speed hub, the endpoint additionally needs to be configured with the function address and port number of the hub

37.6.4. Operation

The initial operating role of the Hi-Speed USB module (Embedded Host mode or Device mode) depends on orientation of the Micro-AB cable. The orientation of the Micro-AB cable determines the logic state of USBID input pin. The USBID low state indicates the connection of the Micro-A side of the Micro-AB cable and initial operation in the Embedded Host role. The USBID high state indicates the connection of the Micro-B side of the Micro-AB cable and initial operation in the Device role. The cable orientation state of USBID can be superseded using the IDOVEN and IDVAL bits of the CTRLA register.

The HOSTREQ bit is provided in the DEVCTL register of the Hi-Speed USB module which can trigger a Host Negotiation, requesting that the system currently operating in the Device role swap to the Embedded Host role during the next USB bus idle period. Information about the current operating role and current bus connection speed can be found in the DEVCTL register.

37.6.5. FIFOs

The Hi-Speed USB module has 9kB of dedicated RAM that can be dynamically allocated between the endpoint FIFOs.

37.6.6. Dedicated DMA

The Hi-Speed USB module has a dedicated, 8-channel DMA controller that can be set up to load/unload the endpoint FIFOs. Each of the DMA channels can be configured to operate in one of two modes:

DMA Mode 0: Allows one packet to be automatically transferred to/from its respective endpoint FIFO

DMA Mode 1: Allows for a complete Bulk transfer to be setup and automatically transferred to/from its respective endpoint FIFO

37.6.7. Configuration

The Hi-Speed USB module has a set of common registers and a set of indexed registers for configuration of the individual endpoints. The indexed endpoint configuration registers are accessed for a specific endpoint by selecting that endpoint using in the INDEX register. For additional information on the configuration and operation of the Hi-Speed USB module, please refer to the documentation, firmware and application examples provided by Microchip and third-party solution providers for PIC32 products. For information about fully USB IF compliant middleware, please refer to available third-party solutions.

37.6.8. Operating Speed

The Hi-Speed USB module supports operation in Hi-Speed (480 Mb/s), Full Speed (12 Mb/s) and Low Speed (1.5 Mb/s).

- Hi-Speed operation can be enabled or disabled using the HSEN bit in the POWR register
- Operation in Full and Hi-Speed modes can be forced in the TESTMODE register for testing purposes
- The operating speed is set using the SPEED bits of the configuration registers for the individual endpoints

37.7. Register Summary: USB Common Registers

For descriptions and definitions of both Register and bitfield properties, refer to Register Properties.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRLA	31:24								
		23:16								
		15:8						REFCLKSEL	IDOVEN	IDVAL
		7:0							ENABLE	SWRST
0x04	CTRLB	31:24								
		23:16					BLANK[19:16]			
		15:8	BLANK[15:8]							
		7:0	BLANK[7:0]							
0x08	CTRLC	31:24								
		23:16								
		15:8								
		7:0				T1MSEN				
0x0C	INTENCLR	31:24								
		23:16								
		15:8								
		7:0			PHYRDY	T1MS	DMA	USB	RESUME	WAKEUP
0x10	INTENSET	31:24								
		23:16								
		15:8								
		7:0			PHYRDY	T1MS	DMA	USB	RESUME	WAKEUP
0x14	INTFLAG	31:24								
		23:16								
		15:8								
		7:0			PHYRDY	T1MS	DMA	USB	RESUME	WAKEUP
0x18	STATUS	31:24								
		23:16								
		15:8								
		7:0						VREGRDY	PHYON	PHYRDY
0x1C	SYNCBUSY	31:24								
		23:16								
		15:8								
		7:0						T1MSEN	ENABLE	SWRST
0x20 ... 0x1000	Reserved									
0x1001	POWR	7:0	ISOUPD	SOFTCONN	HSEN	HSMODE	RESET	RESUME	SUSPMODE	SUSPEN
0x1002	INTRTX	15:8								
		7:0	EP6TXIF	EP5TXIF	EP4TXIF	EP3TXIF	EP2TXIF	EP1TXIF	EP0TXIF	EP0IF
0x1004	INTRRX	15:8								
		7:0	EP6RXIF	EP5RXIF	EP4RXIF	EP3RXIF	EP2RXIF	EP1RXIF	EP0RXIF	
0x1006	INTRTXE	15:8								
		7:0	EP6TXEN	EP5TXEN	EP4TXEN	EP3TXEN	EP2TXEN	EP1TXEN	EP0TXEN	EP0EN
0x1008	INTRRXE	15:8								
		7:0	EP6RXEN	EP5RXEN	EP4RXEN	EP3RXEN	EP2RXEN	EP1RXEN	EP0RXEN	
0x100A	INTRUSB	7:0	VBUSERR	SESSREQ	DISCON	CONN	SOF	RESET	RESUME	SUSPEND
0x100B	INTRUSBE	7:0	VBUSERREN	SESSREQEN	DISCONEN	CONNEN	SOFEN	RESETEN	RESUMEEN	SUSPENDEN
0x100C	FRAME	15:8						FRMNUM[10:8]		
		7:0	FRMNUM[7:0]							
0x100E	INDEX	7:0					SELEP[3:0]			
0x100F	TESTMODE	7:0	FORCEHOST	FIFOACCESS	FORCEFS	FORCEHS	TESTPACKET	TESTK	TESTJ	TESTSE0NAK
0x1010	TXMAXP	15:8	MULT[4:0]					TXMAXP[10:8]		
		7:0	TXMAXP[7:0]							
0x1012 ... 0x1013	Reserved									
0x1014	RXMAXP	15:8	MULT[4:0]					RXMAXP[10:8]		
		7:0	RXMAXP[7:0]							

Register Summary: USB Common Registers (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x1016 ... 0x101E	Reserved									
0x101F	FIFOSIZE	7:0	RXFIFOSIZE[3:0]				TXFIFOSIZE[3:0]			
0x1020	FIFO0	31:24	DATA[31:24]							
		23:16	DATA[23:16]							
		15:8	DATA[15:8]							
		7:0	DATA[7:0]							
0x1024	FIFO1	31:24	DATA[31:24]							
		23:16	DATA[23:16]							
		15:8	DATA[15:8]							
		7:0	DATA[7:0]							
0x1028	FIFO2	31:24	DATA[31:24]							
		23:16	DATA[23:16]							
		15:8	DATA[15:8]							
		7:0	DATA[7:0]							
0x102C	FIFO3	31:24	DATA[31:24]							
		23:16	DATA[23:16]							
		15:8	DATA[15:8]							
		7:0	DATA[7:0]							
0x1030	FIFO4	31:24	DATA[31:24]							
		23:16	DATA[23:16]							
		15:8	DATA[15:8]							
		7:0	DATA[7:0]							
0x1034	FIFO5	31:24	DATA[31:24]							
		23:16	DATA[23:16]							
		15:8	DATA[15:8]							
		7:0	DATA[7:0]							
0x1038	FIFO6	31:24	DATA[31:24]							
		23:16	DATA[23:16]							
		15:8	DATA[15:8]							
		7:0	DATA[7:0]							
0x103C	FIFO7	31:24	DATA[31:24]							
		23:16	DATA[23:16]							
		15:8	DATA[15:8]							
		7:0	DATA[7:0]							
0x1040 ... 0x105F	Reserved									
0x1060	DEVCTL	7:0	BDEVICE	FSDEV	LSDEV	VBUS[1:0]		HOSTMODE	HOSTREQ	SESSION
0x1061	MISC	7:0							TXEDMA	RXEDMA
0x1062	TXFIFOSZ	7:0				DPB	FIFOSZ[3:0]			
0x1063	RXFIFOSZ	7:0				DPB	FIFOSZ[3:0]			
0x1064	TXFIFOADD	15:8				ADDR[12:8]				
		7:0	ADDR[7:0]							
0x1066	RXFIFOADD	15:8				ADDR[12:8]				
		7:0	ADDR[7:0]							
0x1068 ... 0x1077	Reserved									
0x1078	EPIINFO	7:0	RXENDPOINTS[3:0]				TXENDPOINTS[3:0]			
0x1079	Reserved									
0x107A	LINKINFO	7:0	WTCON[3:0]				WTID[3:0]			
0x107B	VPLEN	7:0	VPLEN[7:0]							
0x107C	HSEOF1	7:0	HSEOF1[7:0]							
0x107D	FSEOF1	7:0	FSEOF1[7:0]							
0x107E	LSEOF1	7:0	LSEOF1[7:0]							
0x107F	SOFTRST	7:0							NRSTX	NRST
0x1080 ... 0x11FF	Reserved									

Register Summary: USB Common Registers (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x1200	DMAINTR	31:24								
		23:16								
		15:8								
		7:0	DMA7IF	DMA6IF	DMA5IF	DMA4IF	DMA3IF	DMA2IF	DMA1IF	DMA0IF
0x1204	DMA0CTRL	31:24								
		23:16								
		15:8						DMABRSTM[1:0]		DMAERR
		7:0	DMAEP[3:0]				DMAIE	DMAMODE	DMADIR	DMAEN
0x1208	DMA0ADDR	31:24	DMAADDR[31:24]							
		23:16	DMAADDR[23:16]							
		15:8	DMAADDR[15:8]							
		7:0	DMAADDR[7:0]							
0x120C	DMA0NCOUNT	31:24	DMACOUNT[31:24]							
		23:16	DMACOUNT[23:16]							
		15:8	DMACOUNT[15:8]							
		7:0	DMACOUNT[7:0]							
0x120E	DMA1CTRL	31:24								
		23:16								
		15:8						DMABRSTM[1:0]		DMAERR
		7:0	DMAEP[3:0]				DMAIE	DMAMODE	DMADIR	DMAEN
0x1212	DMA1ADDR	31:24	DMAADDR[31:24]							
		23:16	DMAADDR[23:16]							
		15:8	DMAADDR[15:8]							
		7:0	DMAADDR[7:0]							
0x1216	DMA1NCOUNT	31:24	DMACOUNT[31:24]							
		23:16	DMACOUNT[23:16]							
		15:8	DMACOUNT[15:8]							
		7:0	DMACOUNT[7:0]							
0x1218	DMA2CTRL	31:24								
		23:16								
		15:8						DMABRSTM[1:0]		DMAERR
		7:0	DMAEP[3:0]				DMAIE	DMAMODE	DMADIR	DMAEN
0x121C	DMA2ADDR	31:24	DMAADDR[31:24]							
		23:16	DMAADDR[23:16]							
		15:8	DMAADDR[15:8]							
		7:0	DMAADDR[7:0]							
0x1220	DMA2NCOUNT	31:24	DMACOUNT[31:24]							
		23:16	DMACOUNT[23:16]							
		15:8	DMACOUNT[15:8]							
		7:0	DMACOUNT[7:0]							
0x1222	DMA3CTRL	31:24								
		23:16								
		15:8						DMABRSTM[1:0]		DMAERR
		7:0	DMAEP[3:0]				DMAIE	DMAMODE	DMADIR	DMAEN
0x1226	DMA3ADDR	31:24	DMAADDR[31:24]							
		23:16	DMAADDR[23:16]							
		15:8	DMAADDR[15:8]							
		7:0	DMAADDR[7:0]							
0x122A	DMA3NCOUNT	31:24	DMACOUNT[31:24]							
		23:16	DMACOUNT[23:16]							
		15:8	DMACOUNT[15:8]							
		7:0	DMACOUNT[7:0]							
0x122C	DMA4CTRL	31:24								
		23:16								
		15:8						DMABRSTM[1:0]		DMAERR
		7:0	DMAEP[3:0]				DMAIE	DMAMODE	DMADIR	DMAEN
0x1230	DMA4ADDR	31:24	DMAADDR[31:24]							
		23:16	DMAADDR[23:16]							
		15:8	DMAADDR[15:8]							
		7:0	DMAADDR[7:0]							

Register Summary: USB Common Registers (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x1234	DMA4NCOUNT	31:24	DMACOUNT[31:24]							
		23:16	DMACOUNT[23:16]							
		15:8	DMACOUNT[15:8]							
		7:0	DMACOUNT[7:0]							
0x1236	DMA5CTRL	31:24								
		23:16								
		15:8						DMABRSTM[1:0]		DMAERR
		7:0	DMAEP[3:0]				DMAIE	DMAMODE	DMADIR	DMAEN
0x123A	DMA5ADDR	31:24	DMAADDR[31:24]							
		23:16	DMAADDR[23:16]							
		15:8	DMAADDR[15:8]							
		7:0	DMAADDR[7:0]							
0x123E	DMA5NCOUNT	31:24	DMACOUNT[31:24]							
		23:16	DMACOUNT[23:16]							
		15:8	DMACOUNT[15:8]							
		7:0	DMACOUNT[7:0]							
0x1240	DMA6CTRL	31:24								
		23:16								
		15:8						DMABRSTM[1:0]		DMAERR
		7:0	DMAEP[3:0]				DMAIE	DMAMODE	DMADIR	DMAEN
0x1244	DMA6ADDR	31:24	DMAADDR[31:24]							
		23:16	DMAADDR[23:16]							
		15:8	DMAADDR[15:8]							
		7:0	DMAADDR[7:0]							
0x1248	DMA6NCOUNT	31:24	DMACOUNT[31:24]							
		23:16	DMACOUNT[23:16]							
		15:8	DMACOUNT[15:8]							
		7:0	DMACOUNT[7:0]							
0x124A	DMA7CTRL	31:24								
		23:16								
		15:8						DMABRSTM[1:0]		DMAERR
		7:0	DMAEP[3:0]				DMAIE	DMAMODE	DMADIR	DMAEN
0x124E	DMA7ADDR	31:24	DMAADDR[31:24]							
		23:16	DMAADDR[23:16]							
		15:8	DMAADDR[15:8]							
		7:0	DMAADDR[7:0]							
0x1252	DMA7NCOUNT	31:24	DMACOUNT[31:24]							
		23:16	DMACOUNT[23:16]							
		15:8	DMACOUNT[15:8]							
		7:0	DMACOUNT[7:0]							
0x1256 ... 0x133F	Reserved									
0x1340	RXDPKTBUFDIS	15:8								
		7:0	EP6RXD	EP5RXD	EP4RXD	EP3RXD	EP2RXD	EP1RXD	EP0RXD	
0x1342	TXDPKTBUFDIS	15:8								
		7:0	EP6TXD	EP5TXD	EP4TXD	EP3TXD	EP2TXD	EP1TXD	EP0TXD	
0x1344	CTUCH	15:8	TUCH[15:8]							
		7:0	TUCH[7:0]							
0x1346	CTHHSRTN	15:8	THHSRTN[15:8]							
		7:0	THHSRTN[7:0]							
0x1348	CTHSBT	15:8								
		7:0					HSTMEOUTADD[3:0]			
0x134A ... 0x135F	Reserved									
0x1360	LPMATTR	15:8	ENDPOINT[3:0]							RMTWAK
		7:0	HIRD[3:0]				LNKSTATE[3:0]			
0x1362	LPMCNTRL	7:0				LPMNAK	LPMEN[1:0]		LPMRES	LPMXMT
0x1363	LPMINTREN	7:0			LPMERREN	LPMRESEN	LPMNCEN	LPMACKEN	LPMNYEN	LPMSTEN

Register Summary: USB Common Registers (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x1364	LPMINTR	7:0			LPMERR	LPMRES	LPMNC	LPMACK	LPMNY	LPMST
0x1365 ... 0x14FF	Reserved									
0x1500	PHY00	31:24								
		23:16								
		15:8								
		7:0	RXPHSSEL[2:0]			SLEWRATE[1:0]		PREEMP[2:0]		
0x1504	PHY04	31:24								
		23:16								
		15:8								
		7:0	SQUELCH[2:0]			HIZ	Reserved	TXPHSSEL[2:0]		
0x1508	PHY08	31:24								
		23:16								
		15:8								
		7:0								SQUELCH
0x150C	PHY0C	31:24								
		23:16								
		15:8								
		7:0	TUNE[2:0]							
0x1510	PHY10	31:24								
		23:16								
		15:8								
		7:0	DRVTUNE[2:0]			TUNE[4:0]				
0x1514	PHY14	31:24								
		23:16								
		15:8								
		7:0	ODT			BYPSSSQUELCH		COMPBYPSS[2:0]		
0x1518	PHY18	31:24								
		23:16								
		15:8								
		7:0							ODT[1:0]	
0x151C	PHY1C	31:24								
		23:16								
		15:8								
		7:0	FSLSDIFF						ODTBYPASS	
0x1520	PHY20	31:24								
		23:16								
		15:8								
		7:0	HSSLEW[1:0]							
0x1524	PHY24	31:24								
		23:16								
		15:8								
		7:0	HSDRIVST[1:0]		HSPREEMPST[2:0]			PREEMPHEN	OTGPDN	HSSLEW
0x1528	PHY28	31:24								
		23:16								
		15:8								
		7:0	HSDRVCOMP[2:0]			DISCONDET[3:0]				HSDRIVST
0x152C ... 0x1543	Reserved									
0x1544	PHY44	31:24								
		23:16								
		15:8								
		7:0	FRCSESSEND				FRCVBUSVAL	DIGDBG	PLLDAMP	
0x1548	PHY48	31:24								
		23:16								
		15:8								
		7:0	SESSENDTUNE[2:0]					VBUSCHRGE	FRCBSESSVAL	FRCASESSVAL

Register Summary: USB Common Registers (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x154C	PHY4C	31:24								
		23:16								
		15:8								
		7:0	BSESSVALIDTUNE[1:0]					VBUSVALTUNE[2:0]		
0x1550	PHY50	31:24								
		23:16								
		15:8								
		7:0	COMPCURREF[2:0]				ASESSVALIDTUNE[2:0]			BSESSVALIDTUNE

37.7.1. Control A Register

Name: CTRLA
Offset: 0x0000
Reset: 0x0000000000
Property: PAC Write-Protection

Note: All bits in CTRLA are enable protected except for ENABLE and SWRST.

Table 37-1. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
						REFCLKSEL	IDOVEN	IDVAL
Access						R/W	R/W	R/W
Reset						0	0	0

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

Bit 10 – REFCLKSEL Select USB PLL Reference Clock Speed

Value	Description
0	12 MHz clock input
1	24 MHz clock input

Bit 9 – IDOVEN ID Source Select
Note: Affects the value of DEVCTL.B Device.

Value	Description
0	IDDIG value from PHY is the source of ID
1	IDVAL is the source of ID

Bit 8 – IDVAL Override value of ID

Value	Description
0	ID override value is 0 (A plug) bit
1	ID override value is 1 (B plug)

Bit 1 – ENABLE Enable

Notes:

1. Due to synchronization, there is delay from writing CTRLA.ENABLE until the operation completes. The value written to CTRL.ENABLE reads back immediately and the SYNCBUSY.ENABLE bit is set. SYNC- BUSY.ENABLE is cleared when the operation completes.
2. Before clearing ENABLE ensure the USBCORE Controller has entered Suspend mode.
3. It is necessary to make all other configuration settings in this register first before setting the ENABLE bit.

Value	Description
0	Disable module: Only SFR reads/writes
1	Enable module: Requests Reference Clock

Bit 0 – SWRST Software initiated Reset for USB System

Notes:

1. Writing'1' to CTRLA.SWRST take precedence over other bit updates in the same write-operation. Any register write during the ongoing reset results in a bus error. Reading any register returns the reset value of the register.
2. Due to synchronization there is a delay from writing CTRLA.SWRST until the reset completes. CTRLA.SWRST and SYNCBUSY.SWRST are both cleared when the reset is completes.
3. During a SWRST, access to registers/bits without SWRST are disallowed until SYNCBUSY.SWRST is cleared by hardware.

Value	Description
0	This module remains in its current state. Writing a 0 to this field has no effect.
1	Reset all logic and registers in the USB system, except SYNCBUSY.SWRST, and disable the module (USBCORE, PHY, and VREG).

37.7.2. Control B Register

Name: CTRLB
Offset: 0x0004
Reset: 0x0000000000
Property: PAC Write-Protection

Table 37-2. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

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

Bit	15	14	13	12	11	10	9	8
	BLANK[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
	BLANK[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 19:0 – BLANK[19:0] Blank the Resume/Wakeup Monitoring for XCLK

Blank the Resume/Wakeup Monitoring for XCLK counts when entering Suspend When entering Suspend from HS as a Host, Blank allows time for the downstream Device to detect Suspend and switch from HS to FS (HS term to FS Pull-up) and enter its suspend state, before the Host Resume/Wakeup monitor becomes active.

Notes:

1. The amount of time required to blank instability (HS term to FS pull-up) in LineState depends upon when the downstream device activates its pullup resistors in response to detecting suspend, the transmission line distance, and other factors. The user may adjust this blanking time as required but should consult the USB spec for detection of suspend/resume requirements.
2. Zero is a valid value for LS or FS modes.
3. This field can be safely updated when CTRLA.ENABLE=0 or when POWER.SUSPEND=.

Value	Description
`FFFFF`	Maximum time to ignore Resume/Wakeup detection (220-1)*16.6ns or about 17.4ms.
`00010`	Ignore Resume/Wakeup detector for2*16.6ns
`00001`	Ignore Resume/Wakeup detector for1*16.6ns
`00000`	Do not ignore Resume/Wakeup detector

37.7.3. Control C Register

Name: CTRLC
Offset: 0x0008
Reset: 0x0000000000
Property: PAC Write-Protection

Table 37-3. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
				T1MSEN				
Access				R/W				
Reset				0				

Bit 4 – T1MSEN 1ms Timer Tick Enable

Value	Description
0	Disabled
1	Enabled

37.7.4. Interrupt Enable Clear

Name: INTENCLR
Offset: 0x000C
Reset: 0x0000
Property: PAC Write-Protection

Table 37-4. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			PHYRDY	T1MS	DMA	USB	RESUME	WAKEUP
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit 5 – PHYRDY Physical Ready Bit Interrupt Clear
Writing a ‘1’ to this field clears the interrupt enable.

Value	Description
0	Interrupt disabled
1	Interrupt enabled

Bit 4 – T1MS
Writing a ‘1’ to this field clears the interrupt enable.

Value	Description
0	Interrupt disabled
1	Interrupt enabled

Bit 3 – DMA DMA Interrupt Enable Clear
Writing a ‘1’ to this field clears the interrupt enable.

Value	Description
0	Interrupt disabled
1	Interrupt enabled

Bit 2 – USB USB Interrupt Enable Clear
Writing a '1' to this field clears the interrupt enable.

Value	Description
0	Interrupt disabled
1	Interrupt enabled

Bit 1 – RESUME Resume Interrupt Enable Clear
Writing a '1' to this field clears the interrupt enable.

Value	Description
0	Interrupt disabled
1	Interrupt enabled

Bit 0 – WAKEUP Wake-Up Interrupt Enable Clear
Writing a '1' to this field clears the interrupt enable.

Value	Description
0	Interrupt disabled
1	Interrupt enabled

37.7.5. Interrupt Enable Set

Name: INTENSET
Offset: 0x0010
Reset: 0x0000
Property: PAC Write-Protection

Table 37-5. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			PHYRDY	T1MS	DMA	USB	RESUME	WAKEUP
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit 5 – PHYRDY Physical Ready Bit Interrupt Enable Set
Writing a '1' to this field sets the interrupt enable.

Value	Description
0	Interrupt disabled
1	Interrupt enabled

Bit 4 – T1MS
Writing a '1' to this field sets the interrupt enable.

Value	Description
0	Interrupt disabled
1	Interrupt enabled

Bit 3 – DMA DMA Interrupt Enable Set
Writing a '1' to this field sets the interrupt enable.

Value	Description
0	Interrupt disabled
1	Interrupt enabled

Bit 2 – USB USB Interrupt Enable Set
Writing a '1' to this field sets the interrupt enable.

Value	Description
0	Interrupt disabled
1	Interrupt enabled

Bit 1 – RESUME Resume Interrupt Enable Set
Writing a '1' to this field sets the interrupt enable.

Value	Description
0	Interrupt disabled
1	Interrupt enabled

Bit 0 – WAKEUP Wake-Up Interrupt Enable Set
Writing a '1' to this field sets the interrupt enable.

Value	Description
0	Interrupt disabled
1	Interrupt enabled

37.7.6. Interrupt Flag Register

Name: INTFLAG
Offset: 0x0014
Reset: 0x0000
Property: PAC Write-Protection

Note: Interrupt flags must be cleared and then read back to confirm the clear before exiting the ISR to avoid double interrupts.

Table 37-6. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			PHYRDY	T1MS	DMA	USB	RESUME	WAKEUP
Access			R/W/HS	R/W/HS	R/W/HS	R/W/HS	R/W/HS	R/W/HS
Reset			0	0	0	0	0	0

Bit 5 – PHYRDY RHY Ready Interrupt Flag

Value	Description
0	No Change in STATUS.PHYRDY state
1	Change in STATUS.PHYRDY state

Bit 4 – T1MS Timer 1ms Tick Interrupt Flag

Value	Description
0	1ms Timer has Not Expired
1	1ms Timer has Expired

Bit 3 – DMA DMAINTR Interrupt Flag

Value	Description
0	No interrupt input is present
1	DMA Interrupt

Bit 2 – USB USBCORE General Interrupts Flag

Value	Description
0	No interrupt input is present status
1	General Interrupt

Bit 1 – RESUME Resume Detected Flag

Note: Set to one when USB is in Suspend Mode and a Remote Device asserts a “K” state on the USB bus.

Value	Description
0	No Resume Activity Detected or not in Suspend State
1	Resume from Suspend Detected

Bit 0 – WAKEUP USB Activity Detected while in USB Suspend

Note: This bit can be used to wake the device from Standby.

Value	Description
0	No Activity Detected or not in Suspend State
1	Activity Detected while USB in Suspend

37.7.7. Status Register

Name: STATUS
Offset: 0x0018
Reset: 0x0000
Property: PAC Write-Protection

Table 37-7. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						VREGRDY	PHYON	PHYRDY
Access						R/HC/HS	R/HC/HS	R/HC/HS
Reset						0	0	0

Bit 2 – VREGRDY USB Voltage Regulator status

Value	Description
0	Voltage Regulator output is off
1	Voltage Regulator output is on

Bit 1 – PHYON PHY Power State

Value	Description
0	PHY is in off (low power state)
1	PHY is in on (operational power state)

Bit 0 – PHYRDY PHY Ready

Value	Description
0	PHY is NOT Ready for USBCORE activity
1	PHY is ready for USBCORE activity

37.7.8. Synchronization Busy Register

Name: SYNCBUSY
Offset: 0x001C
Reset: 0x0000
Property: PAC Write-Protection

Table 37-8. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						T1MSEN	ENABLE	SWRST
Access						R/HC/HS	R/HC/HS	R/HC/HS
Reset						0	0	0

Bit 2 – T1MSEN T1MS Enable busy bit

Note: Software must poll this bit to know when the operation completes.

Value	Description
0	T1MSEN synchronization is NOT busy
1	T1MSEN synchronization is busy

Bit 1 – ENABLE Enable Busy bit

Note: Software must poll this bit to know when the operation completes.

Value	Description
0	ENABLE synchronization is NOT busy
1	ENABLE synchronization is busy

Bit 0 – SWRST Software Reset Busy bit

Note: Software must poll this bit to know when the operation completes.

Value	Description
0	SWRST synchronization is NOT busy
1	SWRST synchronization is busy

37.7.9. Power Register

Name: POWR
Offset: 0x1001
Reset: 0x0000
Property: PAC Write-Protection

Table 37-9. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
	ISOUPD	SOFTCONN	HSEN	HSMODE	RESET	RESUME	SUSPMODE	SUSPEN
Access	R/W	R/W	R/W	R/HS	R	R/W	R/HC	R/W
Reset	0	0	1	0	0	0	0	0

Bit 7 – ISOUPD ISO Update Bit
(Device mode only; unimplemented in Hostmode)
This bit only affects endpoints performing isochronous transfers when in *Device mode*. This bit is unimplemented in *Host mode*.

Value	Description
0	No change in behavior
1	USB module will wait for a SOF token from the time TXPKTRDY is set before sending the packet

Bit 6 – SOFTCONN Soft Connect/Disconnect Feature Selection bit
This bit is only available in *Device mode*.

Value	Description
0	The USB D+/D- lines are disabled and are tri-stated
1	The USB D+/D- lines are enabled and active

Bit 5 – HSEN Hi-Speed Enable bit

Value	Description
0	Module only operates in Full-Speed mode
1	The USB module will negotiate for Hi-Speed mode when the device is reset by the hub

Bit 4 – HSMODE Hi-Speed Mode Status bit
In *Device mode*, this bit becomes valid when a USB reset completes. In *Host mode*, it becomes valid when the RESET bit is cleared.

Value	Description
0	Module is not in Hi-Speed mode
1	Hi-Speed mode successfully negotiated during USB reset

Bit 3 – RESET Module Reset Status bit
In *Device mode*, this bit is read-only. In *Host mode*, this bit is read/write.

Value	Description
0	Normal module operation
1	Reset signaling is present on the bus

Bit 2 – RESUME Resume from Suspend Control bit

In *Devicemode*, the software should clear this bit after 10 ms (a maximum of 15 ms) to end Resume signaling. In *Host mode*, the software should clear this bit after 20 ms.

Value	Description
0	Stop Resume signaling
1	Generate Resume signaling when the device is in Suspend mode

Bit 1 – SUSPMODE Suspend Mode Status bit

This bit is read-only in Device mode. In Host mode, it can be set by software, and is cleared by hardware.

Value	Description
0	The USB module is in Normal operations
1	The USB module is in Suspend mode

Bit 0 – SUSPEN Suspend Mode Enable bit

Value	Description
0	Suspend mode is not enabled
1	Suspend mode is enabled

37.7.10. USB TX Interrupt Flag Register

Name: INTRTX
Offset: 0x1002
Reset: 0x0000
Property: PAC Write-Protection

Note: Interrupt flags must be cleared and then read back to confirm the clear before exiting the ISR to avoid double interrupts.

All EPnTX and EP0 bits are cleared when this register is read. Therefore, each bit must be read independently from the remaining bits in this register to avoid accidental clearing.

Table 37-10. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	EP6TXIF	EP5TXIF	EP4TXIF	EP3TXIF	EP2TXIF	EP1TXIF	EP0TXIF	EP0IF
Access	R/HS	R/HS	R/HS	R/HS	R/HS	R/HS	R/HS	R/HS
Reset	0	0	0	0	0	0	0	0

Bits 1, 2, 3, 4, 5, 6, 7 – EPnTXIF Endpoint ‘n’ TX Interrupt Flag bit

Value	Description
0	No interrupt event
1	Endpoint has a transmit interrupt to be serviced

Bit 0 – EP0IF Endpoint 0 Interrupt bit

Value	Description
0	No interrupt event
1	Endpoint 0 has an interrupt to be serviced

37.7.11. USB RX Interrupt Flag Register

Name: INTRRX
Offset: 0x1004
Reset: 0x0000
Property: PAC Write-Protection

Note: Interrupt flags must be cleared and then read back to confirm the clear before exiting the ISR to avoid double interrupts.

All EPnRX and are cleared when this register is read. Therefore, each bit must be read independently from the remaining bits in this register to avoid accidental clearing.

Table 37-11. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	EP6RXIF	EP5RXIF	EP4RXIF	EP3RXIF	EP2RXIF	EP1RXIF	EP0RXIF	
Access	R/HS	R/HS	R/HS	R/HS	R/HS	R/HS	R/HS	
Reset	0	0	0	0	0	0	0	

Bits 1, 2, 3, 4, 5, 6, 7 – EPnRXIF Endpoint ‘n’ RX Interrupt Flag bit

Value	Description
0	No interrupt event
1	Endpoint has a transmit interrupt to be serviced

37.7.12. USB TX Interrupt Enable Register

Name: INTRTXE
Offset: 0x1006
Reset: 0x0000
Property: PAC Write-Protection

Table 37-12. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	EP6TXEN	EP5TXEN	EP4TXEN	EP3TXEN	EP2TXEN	EP1TXEN	EP0TXEN	EP0EN
Access	R/HS	R/HS	R/HS	R/HS	R/HS	R/HS	R/HS	R/HS
Reset	0	0	0	0	0	0	0	0

Bits 1, 2, 3, 4, 5, 6, 7 – EPnTXEN Endpoint ‘n’ Transmit Interrupt Enable bits

Value	Description
0	Endpoint Transmit interrupt events are not enabled
1	Endpoint Transmit interrupt events are enabled

Bit 0 – EP0EN Endpoint 0 Interrupt Enable bit

Value	Description
0	Endpoint 0 interrupt events are not enabled
1	Endpoint 0 interrupt events are enabled

37.7.13. USB RX Interrupt Enable Register

Name: INTRRXE
Offset: 0x1008
Reset: 0x0000
Property: PAC Write-Protection

Table 37-13. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	EP6RXEN	EP5RXEN	EP4RXEN	EP3RXEN	EP2RXEN	EP1RXEN	EP0RXEN	
Access	R/HS	R/HS	R/HS	R/HS	R/HS	R/HS	R/HS	
Reset	0	0	0	0	0	0	0	

Bits 1, 2, 3, 4, 5, 6, 7 – EPnRXEN Endpoint ‘n’ Receive Interrupt Enable bits

Value	Description
0	Endpoint Receive interrupt events are not enabled
1	Endpoint Receive interrupt events are enabled

37.7.14. USB Interrupt Status Register

Name: INTRUSB
Offset: 0x100A
Reset: 0x0000
Property: PAC Write-Protection

Table 37-14. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
	VBUSERR	SESSREQ	DISCON	CONN	SOF	RESET	RESUME	SUSPEND
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 7 – VBUSERR VBUS Error Interrupt Bit

Value	Description
0	No interrupt
1	V_{BUS} has dropped below the VBUS valid threshold during a session

Bit 6 – SESSREQ Session Request Interrupt bit

Value	Description
0	No session request detected
1	Session request signaling has been detected

Bit 5 – DISCON Device Disconnect Interrupt bit

Value	Description
0	No device disconnect detected
1	In *Host mode*, indicates when a device disconnect is detected. In *Device mode*, indicates when a session ends.

Bit 4 – CONN Device Connection Interrupt bit

Value	Description
0	No device connection detected
1	In *Host mode*, indicates when a device connection is detected

Bit 3 – SOF Start of Frame Interrupt bit

Value	Description
0	No start of frame detected
1	A new frame has started

Bit 2 – RESET Reset/Babble Interrupt bit

Value	Description
0	No reset/babble detected
1	In *Host mode*, indicates babble is detected. In *Device mode*, indicates reset signaling is detected on the bus.

Bit 1 – RESUME Resume Interrupt bit

Value	Description
0	No Resume signaling detected
1	Resume signaling is detected on the bus while USB module is in Suspend mode

Bit 0 – SUSPEND Suspend Interrupt bit

Value	Description
0	No suspend signaling detected
1	Suspend signaling is detected on the bus *(Device mode)*

37.7.15. USB Interrupt Enable Register

Name: INTRUSBE
Offset: 0x100B
Reset: 0x0000
Property: PAC Write-Protection

Table 37-15. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
	VBUSERREN	SESSREQEN	DISCONEN	CONNEN	SOFEN	RESETEN	RESUMEEN	SUSPENDEN
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	1	0

Bit 7 – VBUSERREN VBUS Error Interrupt Enable Bit

Value	Description
0	V_{BUS} error interrupt is disabled
1	V_{BUS} error interrupt is enabled

Bit 6 – SESSREQEN Session Request Interrupt Enable bit

Value	Description
0	Session request interrupt is disabled
1	Session request interrupt is enabled

Bit 5 – DISCONEN Device Disconnect Interrupt Enable bit

Value	Description
0	Device disconnect interrupt is disabled
1	Device disconnect interrupt is enabled

Bit 4 – CONNEN Device Connection Interrupt Enable bit

Value	Description
0	Device connection interrupt is disabled
1	Device connection interrupt is enabled

Bit 3 – SOFEN Start of Frame Interrupt Enable bit

Value	Description
0	Start of Frame event interrupt is disabled
1	Start of Frame event interrupt is enabled

Bit 2 – RESETEN Reset/Babble Interrupt Enable bit

Value	Description
0	Reset/Babble interrupt is disabled
1	Interrupt when reset (Device mode) or Babble (Host mode) is enabled

Bit 1 – RESUMEEN Resume Interrupt Enable bit

Value	Description
0	Resume signaling interrupt is disabled
1	Resume signaling interrupt is enabled

Bit 0 – SUSPENDEN Suspend Interrupt Enable bit

Value	Description
0	Suspend signaling interrupt is disabled
1	Suspend signaling interrupt is enabled

37.7.16. USB RX Frame Number Register

Name: FRAME
Offset: 0x100C
Reset: 0x0000
Property: PAC Write-Protection

Table 37-16. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	15	14	13	12	11	10	9	8
						FRMNUM[10:8]		
Access						R	R	R
Reset						0	0	0

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

Bits 10:0 – FRMNUM[10:0] Last Received Frame Number bits

37.7.17. USB Endpoint Select Register

Name: INDEX
Offset: 0x100E
Reset: 0x0000
Property: PAC Write-Protection

Table 37-17. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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

Bits 3:0 – SELEP[3:0] Endpoint Registers Select bits

37.7.18. USB Test Mode Register

Name: TESTMODE
Offset: 0x100F
Reset: 0x0000
Property: PAC Write-Protection

Table 37-18. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
	FORCEHOST	FIFOACCESS	FORCEFS	FORCEHS	TESTPACKET	TESTK	TESTJ	TESTSE0NAK
Access	R/W	R/W/HC	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 7 – FORCEHOST Test Mode Force Host Select bit

Value	Description
0	Normal operation
1	Forces USB module into *Host mode*, regardless of whether it is connected to any peripheral

Bit 6 – FIFOACCESS Test Mode Endpoint 0 FIFO Transfer Force bit

Value	Description
0	No transfer
1	Transfers the packet in the Endpoint 0 TX FIFO to the Endpoint 0 RX FIFO

Bit 5 – FORCEFS Test Mode Force Full-Speed Mode Select bit
This bit is only active if FORCEHST = 1.

Value	Description
0	If FORCEHS = 0, places USB module into Low-Speed mode.
1	Forces USB module into Full-Speed mode. Undefined behavior if FORCEHS = 1.

Bit 4 – FORCEHS Test Mode Force Hi-Speed Mode Select bit
This bit is only active if FORCEHST = 1.

Value	Description
0	If FORCEFS = 0, places USB module into Low-Speed mode.
1	Forces USB module into Hi-Speed mode. Undefined behavior if FORCEFS = 1.

Bit 3 – TESTPACKET Test_Packet Test Mode Select bit
This bit is only active if module is in Hi-Speed mode.

Value	Description
0	Normal operation
1	The USB module repetitively transmits on the bus a 53-byte test packet. Test packet must be loaded into the Endpoint 0 FIFO before the test mode is entered.

Bit 2 – TESTK Test_K Test Mode Select bit
This bit is only active if the USB module is in Hi-Speed mode.

Value	Description
0	Normal operation
1	Enters Test_K test mode. The USB module transmits a continuous K on the bus.

Bit 1 – TESTJ Test_J Test Mode Select bit
This bit is only active if the USB module is in Hi-Speed mode.

Value	Description
0	Normal operation
1	Enters Test_J test mode. The USB module transmits a continuous J on the bus.

Bit 0 – TESTSE0NAK Test_SE0_NAK Test Mode Select bit
This mode is only active if module is in Hi-Speed mode

Value	Description
0	Normal operation
1	Enter Test_SE0_NAK test mode. The USB module remains in Hi-Speed mode but responds to any valid IN token with a NAK

37.7.19. USB Max Data TX Register

Name: TXMAXP
Offset: 0x1010
Reset: 0x0000
Property: PAC Write-Protection

Table 37-19. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	15	14	13	12	11	10	9	8
	MULT[4:0]					TXMAXP[10:8]		
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
	TXMAXP[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 15:11 – MULT[4:0] Multiplier Control bits

For Isochronous/Interrupt endpoints or of packet splitting on Bulk endpoints, multiplies TXMAXP by MULT+1 for the payload size.

For Bulk endpoints, MULT can be up to 32 and defines the number of "USB" packets of the specified payload into which a single data packet placed in the FIFO should be split, prior to transfer. The data packet is required to be an exact multiple of the payload specified by TXMAXP.

For Isochronous/Interrupts endpoints operating in Hi-Speed mode, MULT may be either 2 or 3 and specifies the maximum number of such transactions that can take place in a single microframe.

Bits 10:0 – TXMAXP[10:0] Maximum TX Payload per Transaction Control bits

This field sets the maximum payload (in bytes) transmitted in a single transaction. The value is subject to the constraints placed by the USB Specification on packet sizes for Bulk, Interrupt and Isochronous transfers in Full-Speed and Hi-Speed operations.

TXMAXP must be set to an even number of bytes for proper interrupt generation in DMA Mode 1.

37.7.20. USB Max Data RX Register

Name: RXMAXP
Offset: 0x1014
Reset: 0x0000
Property: PAC Write-Protection

Table 37-20. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	15	14	13	12	11	10	9	8
	MULT[4:0]					RXMAXP[10:8]		
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
	RXMAXP[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 15:11 – MULT[4:0] Multiplier Control bits

ForIsochronous/Interrupt endpoints or of packet splitting on Bulk endpoints, multiplies RXMAXP by MULT+1 for the payload size.
For Bulk endpoints, MULT can be up to 32 and defines the number of "USB" packets of the specified payload into which a single data packet placed in the FIFO should be split, prior to transfer. The data packet is required to be an exact multiple of the payload specified by RXMAXP.
For Isochronous/Interrupts endpoints operating in Hi-Speed mode, MULT may be either 2 or 3 and specifies the maximum number of such transactions that can take place in a single microframe.

Bits 10:0 – RXMAXP[10:0] Maximum RX Payload per Transaction Control bits

This field sets the maximum payload (in bytes) transmitted in a single transaction. The value is subject to the constraints placed by the USB Specification on packet sizes for Bulk, Interrupt and Isochronous transfers in Full-Speed and Hi-Speed operations.
RXMAXP must be set to an even number of bytes for proper interrupt generation in DMA Mode 1.
Note: Transfer size greater than RxMaxP is handled by DMA Mode 1 only.

37.7.21. USB TX/RX FIFO Size Register

Name: FIFOSIZE
Offset: 0x101F
Reset: 0x0000
Property: PAC Write-Protection

FIFOSIZE is a Read-Only register that returns the sizes of the FIFOs associated with the selected additional TX/Rx endpoints. The lower nibble encodes the size of the selected TX endpoint FIFO; the upper nibble encodes the size of the selected Rx endpoint FIFO. Values of 3 – 13 correspond to a FIFO size of 2n bytes (8 – 8192 bytes). If an endpoint has not been configured, a value of 0 will be displayed. Where the TX and Rx endpoints share the same FIFO, the Rx FIFO size will be encoded as 0xF.

Table 37-21. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
	RXFIFOSIZE[3:0]				TXFIFOSIZE[3:0]			
Access	R	R	R	R	R	R	R	R
Reset	x	x	x	x	x	x	x	x

Bits 7:4 – RXFIFOSIZE[3:0] Receive FIFO Size bits
This register only has this interpretation when dynamic sizing is not selected. It is not valid where dynamic FIFO sizing is used.

Value	Description
1111	Reserved
1110	Reserved
1101	8192 bytes
1100	4096 bytes
0011	8 bytes
0010	Reserved
0001	Reserved
0000	Reserved or endpoint has not been configured

Bits 3:0 – TXFIFOSIZE[3:0] Transmit FIFO Size bits
This register only has this interpretation when dynamic sizing is not selected. It is not valid where dynamic FIFO sizing is used.

Value	Description
1111	Reserved
1110	Reserved
1101	8192 bytes
1100	4096 bytes
0011	8 bytes
0010	Reserved
0001	Reserved
0000	Reserved or endpoint has not been configured

37.7.22. USB RX End Point Double Packet Buffer Disable Register

Name: RXDPKTBUFDIS
Offset: 0x1340
Reset: 0x0000
Property: PAC Write-Protection

Table 37-22. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	EP6RXD	EP5RXD	EP4RXD	EP3RXD	EP2RXD	EP1RXD	EP0RXD	
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	0	0	0	0	0	0	0	

Bits 1, 2, 3, 4, 5, 6, 7 – EPnRXD RX Endpoint 'x' Double Packet Buffer Disable bits

Value	Description
0	RX double packet buffering is enabled for endpoint 'x'
1	RX double packet buffering is disabled for endpoint 'x'

37.7.23. USB TX End Point Double Packet Buffer Disable Register

Name: TXDPKTBUFDIS
Offset: 0x1342
Reset: 0x0000
Property: PAC Write-Protection

Table 37-23. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	EP6TXD	EP5TXD	EP4TXD	EP3TXD	EP2TXD	EP1TXD	EP0TXD	
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	0	0	0	0	0	0	0	

Bits 1, 2, 3, 4, 5, 6, 7 – EPnTXD TX Endpoint 'n' Double Packet Buffer Disable bits

Value	Description
0	TX double packet buffering is enabled for endpoint 'n'
1	TX double packet buffering is disabled for endpoint 'n'

37.7.24. USB FIFO Data Register ‘x’

Name: FIFOx
Offset: 0x1020 + x*0x04 [x=0..7]
Reset: 0x0000000000
Property: PAC Write-Protection

Table 37-24. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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] USB Transmit/Receive FIFO Data bits
Writes to this register loads data into the TxFIFO for the corresponding endpoint. Reading from this register unloads data from the RxFIFO for the corresponding endpoint.
Transfers may be 8-bit, 16-bit or 32-bit as required, and any combination of access is allowed provided the data accessed is contiguous. However, all transfers associated with one packet must be of the same width so that data is consistently byte-, word- or double-word aligned. The last transfer may contain fewer bytes than the previous transfers in order to complete an odd-byte or odd-word transfer.

37.7.25. USB Device Control Register

Name: DEVCTL
Offset: 0x1060
Reset: 0x0000
Property: PAC Write-Protection

Table 37-25. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
	BDEVICE	FSDEV	LSDEV	VBUS[1:0]		HOSTMODE	HOSTREQ	SESSION
Access	R	R	R/W	R/W	R/W	R	R/W/HC	R/W
Reset	1	0	1	0	0	0	0	0

Bit 7 – BDEVICE USB Device Type bit

Value	Description
0	USB is operating as an 'A' device
1	USB is operating as a 'B' device

Bit 6 – FSDEV Full-Speed/Hi-Speed Device Detection bit (Host mode)

Value	Description
0	No Full-Speed or Hi-Speed device detected
1	A Full-Speed or Hi-Speed device has been detected being connected to the port

Bit 5 – LSDEV Low-Speed Device Detection bit (Host mode)

Value	Description
0	No Low-Speed device detected
1	A Low-Speed device has been detected being connected to the port

Bits 4:3 – VBUS[1:0] VBUS Level Detection bits

Value	Description
11	Above V_{BUS} Valid
10	Above AValid, below V_{BUS} Valid
01	Above Session End, below AValid
00	Below Session End

Bit 2 – HOSTMODE Host Mode bit

Value	Description
0	USB module is not acting as aHost
1	USB module is acting as a Host

Bit 1 – HOSTREQ Host Request Control bit
'B' device only:

Value	Description
0	Host Negotiation is not taking place

Value	Description
1	USB module initiates the Host Negotiation when Suspend mode is entered. This bit is cleared when Host Negotiation is completed.

Bit 0 – SESSION Active Session Control/Status

'A'device:
1= Start a session
0= End a session
'B' device:
1= (Read) Session has started or is in progress, (Write) Initiate the Session Request Protocol
0= When USB module is in Suspend mode, clearing this bit will cause a software disconnect
Note: Clearing this bit when the USB module is not suspended will result in undefined behavior.

37.7.26. USB Miscellaneous Register

Name: MISC
Offset: 0x1061
Reset: 0x0000
Property: PAC Write-Protection

Table 37-26. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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

Bit 1 – TXEDMA TX Endpoint DMA Assertion Control bit

Value	Description
0	DMA_REQ signal for all IN endpoints will be deasserted when MAXP bytes have been written to an endpoint. This is Late mode.
1	DMA_REQ signal for all IN endpoints will be deasserted when MAXP-8 bytes have been written to an endpoint. This is Early mode.

Bit 0 – RXEDMA RX Endpoint DMA Assertion Control bit

Value	Description
0	DMA_REQ signal for all IN endpoints will be deasserted when MAXP bytes have been written to an endpoint. This is Late mode.
1	DMA_REQ signal for all IN endpoints will be deasserted when MAXP-8 bytes have been written to an endpoint. This is Early mode.

37.7.27. USB TX FIFO Size Register

Name: TXFIFOSZ
Offset: 0x1062
Reset: 0x0000
Property: PAC Write-Protection

TXFIFOSZ controls the size of the selected TX endpoint FIFO.

Table 37-27. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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

Bit 4 – DPB TX Endpoint Double-packet Buffering Control bit

Value	Description
0	Double-packet buffer is not supported
1	Double-packet buffer is supported. This doubles the size set in TXFIFOSZ.

Bits 3:0 – FIFOSZ[3:0] TX Endpoint FIFO packet size bits
The maximum packet size to allowed for (before any splitting within the FIFO of Bulk/High-Bandwidth packets prior to transmission)

Value	Description
1111	Reserved
1010	Reserved
1001	4096 bytes
1000	2048 bytes
0111	1024 bytes
0110	512 bytes
0101	256 bytes
0100	128 bytes
0011	64 bytes
0010	32 bytes
0001	16 bytes
0000	8 bytes

37.7.28. USB RX FIFO Size Register

Name: RXFIFOSZ
Offset: 0x1063
Reset: 0x0000
Property: PAC Write-Protection

RXFIFOSZ controls the size of the selected RX endpoint FIFO.

Table 37-28. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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

Bit 4 – DPB RX Endpoint Double-packet Buffering Control bit

Value	Description
0	Double-packet buffer is not supported
1	Double-packet buffer is supported. This doubles the size set in RXFIFOSZ.

Bits 3:0 – FIFOSZ[3:0] RX Endpoint FIFO packet size bits
The maximum packet size to allowed for (before any splitting within the FIFO of Bulk/High-Bandwidth packets prior to transmission)

Value	Description
1111	Reserved
1010	Reserved
1001	4096 bytes
1000	2048 bytes
0111	1024 bytes
0110	512 bytes
0101	256 bytes
0100	128 bytes
0011	64 bytes
0010	32 bytes
0001	16 bytes
0000	8 bytes

37.7.29. USB TX FIFO Address Register

Name: TXFIFOADD
Offset: 0x1064
Reset: 0x0000
Property: PAC Write-Protection

TXFIFOADD controls the start address of the selected TX endpoint FIFO.

Table 37-29. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	15	14	13	12	11	10	9	8
				ADDR[12:8]				
Access				R	R	R	R	R
Reset				0	0	0	0	0

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

Bits 12:0 – ADDR[12:0] Transmit Endpoint FIFO Address bits
Start address of the endpoint FIFO in units of 8 bytes as follows:

Value	Description
1111111111111	0xFFF8
0000000000010	0x0010
0000000000001	0x0008
0000000000000	0x0000

37.7.30. USB RX FIFO Address Register

Name: RXFIFOADD
Offset: 0x1066
Reset: 0x0000
Property: PAC Write-Protection

RXFIFOADD controls the start address of the selected RX endpoint FIFO.

Table 37-30. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	15	14	13	12	11	10	9	8
				ADDR[12:8]				
Access				R	R	R	R	R
Reset				0	0	0	0	0

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

Bits 12:0 – ADDR[12:0] Receive Endpoint FIFO Address bits
Start address of the endpoint FIFO in units of 8 bytes as follows:

Value	Description
1111111111111	0xFFF8
0000000000010	0x0010
0000000000001	0x0008
0000000000000	0x0000

37.7.31. USB Endpoint Info Register

Name: EPIINFO
Offset: 0x1078
Reset: 0x0000
Property: PAC Write-Protection

Table 37-31. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
	RXENDPOINTS[3:0]				TXENDPOINTS[3:0]			
Access	R	R	R	R	R	R	R	R
Reset	0	1	1	1	0	1	1	1

Bits 7:4 – RXENDPOINTS[3:0] Included RX Endpoint bits
This read-only register gives the number of RX endpoints in the design. For the PIC32CK SG/GC family, this number is 7.

Bits 3:0 – TXENDPOINTS[3:0] Included TX Endpoint bits
These read-only bits provide the number of TX endpoints in the design. For the PIC32CK SG/GC family, this number is 7.

37.7.32. USB Link Info Delay Register

Name: LINKINFO
Offset: 0x107A
Reset: 0x0000
Property: PAC Write-Protection

Table 37-32. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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

Bits 7:4 – WTCON[3:0] Connect/Disconnect Filter Control bits
Sets the wait to be applied to allow for the connect/disconnect filter in units of 533.3 ns. The default setting corresponds to 2.667 µs.

Bits 3:0 – WTID[3:0] ID Delay Valid Control bits
Sets the delay to be applied from ID Pull up resistor being asserted to ID input Signal being considered valid in units of 4.369ms. The default setting corresponds to 52.43ms.

37.7.33. USB VBUS Pulse Charge Duration Register

Name: VPLEN
Offset: 0x107B
Reset: 0x0000
Property: PAC Write-Protection

Table 37-33. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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

Bits 7:0 – VPLEN[7:0] VBUS Pulsing Charge Length bits.
Sets the duration of the VBUS pulsing charge in units of 546.1 μs. (The default setting corresponds to 32.77 ms.)

37.7.34. USB HS EOF Register

Name: HSEOF1
Offset: 0x107C
Reset: 0x0000
Property: PAC Write-Protection

Table 37-34. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
	HSEOF1[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 – HSEOF1[7:0] High-Speed EOF bits.
These bits set the Hi-Speed transaction in units of 133.3 μs (default setting is 17.07μs) prior to the EOF to stop new transactions from beginning.

37.7.35. USB FS EOF Register

Name: FSEOF1
Offset: 0x107D
Reset: 0x0000
Property: PAC Write-Protection

Table 37-35. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
	FSEOF1[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	0	1	1	1	0

Bits 7:0 – FSEOF1[7:0] Full-Speed EOF bits.
These bits set the Full-Speed transaction in units of 533.3 μs (default setting is 63.46 μs) prior to the EOF to stop new transactions from beginning.

37.7.36. USB LS EOF Register

Name: LSEOF1
Offset: 0x107E
Reset: 0x0000
Property: PAC Write-Protection

Table 37-36. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
	LSEOF1[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	1	0	0	1	1	1	0

Bits 7:0 – LSEOF1[7:0] Low-Speed EOF bits.
These bits set the Low-Speed transaction in units of 1.067 μs (default setting is 121.6 μs) prior to the EOF to stop new transactions from beginning.

37.7.37. USB Soft Reset Register

Name: SOFTRST
Offset: 0x107F
Reset: 0x0000
Property: PAC Write-Protection

Table 37-37. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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

Bit 1 – NRSTX Reset of XCLK Domain bit

Value	Description
0	Normal operation
1	Reset the XCLK domain, which is clock recovered from the received data by the PHY

Bit 0 – NRST Reset of CLK Domain bit

Value	Description
0	Normal operation
1	Reset the CLK domain, which is clock recovered from the peripheral bus

37.7.38. USB DMA Interrupt Register

Name: DMAINTR
Offset: 0x1200
Reset: 0x0000
Property: PAC Write-Protection

All bits are cleared on a read of the register.

Table 37-38. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	DMA7IF	DMA6IF	DMA5IF	DMA4IF	DMA3IF	DMA2IF	DMA1IF	DMA0IF
Access	R/W/HS	R/W/HS	R/W/HS	R/W/HS	R/W/HS	R/W/HS	R/W/HS	R/W/HS
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7 – DMAxIF DMA Channel x Interrupt bit

Value	Description
0	No interrupt event
1	The DMA channel has an interrupt event

37.7.39. USB DMA Channel x Control Register

Name: DMAxCTRL
Offset: 0x1204 + x*0x0A [x=0..7]
Reset: 0x0000000000
Property: PAC Write-Protection

Table 37-39. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
						DMABRSTM[1:0]		DMAERR
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	7	6	5	4	3	2	1	0
	DMAEP[3:0]				DMAIE	DMAMODE	DMADIR	DMAEN
Access	R/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 10:9 – DMABRSTM[1:0] DMA Burst Mode Selection bit

Value	Description
11	Burst Mode 3: INCR16, INCR8, INCR4 or unspecified length
10	Burst Mode 2: INCR8, INCR4 or unspecified length
01	Burst Mode 1: INCR4 or unspecified length
00	Burst Mode 0: Bursts of unspecified length

Bit 8 – DMAERR DMA Bus Error bit

Value	Description
0	The software writes this to clear the error
1	A bus error has been observed on the input

Bits 7:4 – DMAEP[3:0] DMA Endpoint Assignment bits
These bits hold the endpoint that the DMA channel is assigned to. Valid values are 0-7.

Bit 3 – DMAIE DMA Interrupt Enable bit

Value	Description
0	Interrupt is disabled for this channel
1	Interrupt is enabled for this channel

Bit 2 – DMAMODE DMA Transfer Mode bit

Value	Description
0	DMA Mode0 Transfers
1	DMA Mode1 Transfers

Bit 1 – DMADIR DMA Transfer Direction bit

Value	Description
0	DMA Write (RX endpoint)
1	DMA Read (TX endpoint)

Bit 0 – DMAEN DMA Enable bit

Value	Description
0	Disable the DMA transfer
1	Enable the DMA transfer and start the transfer

37.7.40. USB DMA Channel x Memory Address Register

Name: DMAxADDR
Offset: 0x1208 + x*0x0A [x=0..7]
Reset: 0x0000000000
Property: PAC Write-Protection

This register identifies the current memory address of the corresponding DMA channel. The initial memory address written to this register during initialization must have a value such that its modulo 4 value is equal to '0'. The lower two bits of this register are read only and cannot be set by software. As the DMA transfer progresses, the memory address will increment as bytes are transferred.

Table 37-40. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	DMAADDR[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
	DMAADDR[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
	DMAADDR[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
	DMAADDR[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 – DMAADDR[31:0] DMA Memory Address bits

37.7.41. USB DMA Channel x Count Register

Name: DMAxNCOUNT
Offset: 0x120C + x*0x0A [x=0..7]
Reset: 0x0000000000
Property: PAC Write-Protection

This register identifies the current DMA count of the transfer. Software will set the initial count of the transfer which identifies the entire transfer length. As the count progresses this count is decremented as bytes are transferred.

Table 37-41. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	DMACOUNT[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
	DMACOUNT[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
	DMACOUNT[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
	DMACOUNT[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 – DMACOUNT[31:0] DMA Transfer Count bits

37.7.42. USB High Chirp Time-out Register

Name: CTUCH
Offset: 0x1344
Reset: 0x0000
Property: PAC Write-Protection

Note: Use of this register will allow the Hi-Speed time-out to be set to values that are greater than the maximum specified in the USB 2.0 specification, making the USB module non-compliant.

Table 37-42. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	15	14	13	12	11	10	9	8
	TUCH[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	1	1	1	0	1	0	0

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

Bits 15:0 – TUCH[15:0] Chirp Time-out bits.
These bits set the chirp time-out. This number, when multiplied by 4, represents the number of USB module clock cycles before the time-out occurs.

37.7.43. USB High Speed Resume Signal Delay Register

Name: CTHHSRTN
Offset: 0x1346
Reset: 0x0000
Property: PAC Write-Protection

Note: Use of this register will allow the Hi-Speed time-out to be set to values that are greater than the maximum specified in the USB 2.0 specification, making the USB module non-compliant.

Table 37-43. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	15	14	13	12	11	10	9	8
	THHSRTN[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	1	1	0	0	1	1	1

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

Bits 15:0 – THHSRTN[15:0] Hi-Speed Resume Signaling Delay bits.
These bits set the delay from the end of Hi-Speed resume signaling (acting as a Host) to enable the UTM normal operating mode.

37.7.44. USB High Speed Time-out Adder Register

Name: CTHSBT
Offset: 0x1348
Reset: 0x0000
Property: PAC Write-Protection

Note: Use of this register will allow the Hi-Speed time-out to be set to values that are greater than the maximum specified in the USB 2.0 specification, making the USB module non-compliant.

Table 37-44. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

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

Bits 3:0 – HSTMEOUTADD[3:0] Hi-Speed Time-out Adder bits.
These bits represent the value to be added to the minimum high speed time-out period of 736 bit times. The time-out period can be increased in increments of 64 Hi-Speed bit times (133 ns).

37.7.45. USB LPM Attribute Register for LPM Transaction and Sleep Cycle

Name: LPMATTR
Offset: 0x1360
Reset: 0x0000
Property: PAC Write-Protection

This register is used to define the attributes of an LPM transaction and sleep cycle. In both the Host mode and the Device mode, the meaning of this register is the same however the source of the data is different for Host and Device as follows:

In Device mode:

In Device mode, the values in this register will contain the equivalent attributes that were received in the last LPM transaction that was accepted. This register is updated with the LPM packet contents if the response to the LPM transaction was an ACK. This register can be update via software. In all other cases, this register will hold its current value.

In Host mode:

In Host mode software will set-up the values in this register to define the next LPM transaction that will be transmitted. These values will be inserted in the payload of the next LPM Transaction.

Table 37-45. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	15	14	13	12	11	10	9	8
	ENDPOINT[3:0]							RMTWAK
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
	HIRD[3:0]				LNKSTATE[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

Bits 15:12 – ENDPOINT[3:0] LPM Token Packet Endpoint bits
This is the endpoint in the token packet of the LPM transaction.

Bit 8 – RMTWAK Remote Wake-up Enable bit
This bit is applied on a temporary basis only and is only applied to the current suspend state. After the suspend cycle, the remote wakeup capability that was negotiated upon enumeration applies.

Value	Description
0	Remote wake-up is disabled
1	Remote wake-up is enabled

Bits 7:4 – HIRD[3:0] Host Initiated Resume Duration bits
The minimum time the host will drive resume on the bus. The value in this register corresponds to an actual resume time of:
Resume Time = 50 μs + HIRD * 75 μs. The resulting range is 50 μs to 1200 μs.

Bits 3:0 – LNKSTATE[3:0] Link State bits

This value is provided by the host to the peripheral to indicate what state the peripheral must transition to after the receipt and acceptance of an LPM transaction. The only valid value for this register is '1' for Sleep State (L1). All other values are reserved.

37.7.46. USB LPM Control Register

Name: LPMCNTRL
Offset: 0x1362
Reset: 0x0000
Property: PAC Write-Protection

Table 37-46. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
				LPMNAK	LPMEN[1:0]		LPMRES	LPMXMT
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bit 4 – LPMNAK LPM-only Transaction Setting bit

Device Mode only bit. Unimplemented in Host mode.
This bit is used to place all end points in a state such that the response to all transactions other than an LPM transaction will be a NAK.
Setting this bit to '1'will only take effect after the USB module has been LPM suspended. In this case, the USB device will continue to NAK until this bit has been cleared by software.

Value	Description
0	Normal transaction operation
1	All endpoints will respond to all transactions other than an LPM transaction with a NAK

Bits 3:2 – LPMEN[1:0] LPM Enable bits

Device Mode only bit. Unimplemented in Host mode.

Value	Description
11	LPM Extended transactions are supported
10	LPM and Extended transactions are not supported
01	LPM mode is not supported but Extended transactions are supported
00	LPM Extended transactions are supported

Bit 1 – LPMRES LPM Resume bit

When in Device mode:
This bit is used by software to initiate resume (remote wakeup). This bit differs from the classic RESUME bit in the POWER register (address offset 0x0001) in that the RESUME signal timing is controlled by hardware. When software writes this bit, resume signaling will be asserted for 50us. This bit is self-clearing.
1 = Initiate resume (remote wake-up). Resume signaling is asserted for 50 μs.
0 = No resume operation
This bit is self-clearing.
When in Host mode:
This bit is used by software to initiate a RESUME from the L1 State. This bit differs from the classic RESUME bit in the POWER register (address offset 0x0001) in that the RESUME signal timing is controlled by hardware. When software writes this bit, resume signaling will be asserted for a time specified by the HIRD field in the LPMATTR register.
1 = Initiate resume
0 = No resume operation

This bit is self-clearing.

Bit 0 – LPMXMT LPM Transition to the L1 State bit

When in Device mode:
1 = USB module will transition to the L1 state upon the receipt of the next LPM transaction. LPMEN must be set to `'0b11`. Both LPMXMT and LPMEN must be set in the same cycle.
0 = Maintain current state
When LPMXMT and LPMEN are set, the USB module can respond in the following ways:

- If no data is pending (all TX FIFOs are empty), the USB module will respond with an ACK. The bit will self-clear and a software interrupt will be generated.
- If data is pending (data resides in at least one TX FIFO), the USB module will respond with a NYET. In this case, the bit will not self-clear however a software interrupt will be generated.

When in Host mode:
Software should set this bit to transmit an LPM transaction.
1 = USB module will transmit an LPM transaction. This bit is self-clearing and will be immediately cleared upon receipt of any Token or three time-outs have occurred.
0 = Maintain current state

37.7.47. USB LPM Interrupts Enable Register

Name: LPMINTREN
Offset: 0x1363
Reset: 0x0000
Property: PAC Write-Protection

Table 37-47. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
			LPMERREN	LPMRESEN	LPMNCEN	LPMACKEN	LPMNYEN	LPMSTEN
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit 5 – LPMERREN LPM Error Interrupt Enable bit
Device Mode only bit. Unimplemented in Host mode.

Value	Description
0	LPMERR interrupt is disabled
1	LPMERR interrupt is enabled

Bit 4 – LPMRESEN LPM Resume Interrupt Enable bit

Value	Description
0	LPMRES interrupt is disabled
1	LPMRES interrupt is enabled

Bit 3 – LPMNCEN LPM Not Complete Interrupt Enable bit

Value	Description
0	Disable the LPMNC Interrupt
1	Enable the LPMNC Interrupt

Bit 2 – LPMACKEN LPM ACK Interrupt Enable bit

Value	Description
0	Disable the LPMACK Interrupt
1	Enable the LPMACK Interrupt

Bit 1 – LPMNYEN LPM NYET Interrupt Enable bit

Value	Description
0	Disable the LPMNYET Interrupt
1	Enable the LPMNYET Interrupt

Bit 0 – LPMSTEN LPM Stall Interrupt Enable bit

Value	Description
0	Disable the LPMST Interrupt
1	Enable the LPMST Interrupt

37.7.48. USB LPM Interrupts Status Register

Name: LPMINTR
Offset: 0x1364
Reset: 0x0000
Property: PAC Write-Protection

Table 37-48. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
			LPMERR	LPMRES	LPMNC	LPMACK	LPMNY	LPMST
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bit 5 – LPMERR LPM Error Interrupt Flag bit
Device Mode only bit. Unimplemented in Host mode.

Value	Description
0	No error condition
1	An LPM transaction was received that had a LINKSTATE field that is not supported. The response will be a STALL.

Bit 4 – LPMRES LPM Resume Interrupt Flag bit
This bit is mutually exclusive from the RESUME bit in the Power register.

Value	Description
0	No Resume condition
1	The USB module has resumed (for any reason)

Bit 3 – LPMNC LPM Not Complete Interrupt Flag bit
When in Device mode:
1= The USB module received an LPM transaction and responded with a NYET due to data pending in the RX FIFOs. This can only occur under following condition: The LPMRESP field in the LPMCNTRL register is set to 0x3, the LPMXMT field is set to 0x1 and there was data pending in the USB module's TX FIFOs.
0 = No NC interrupt condition
When in Host mode:
1= An LPM transaction is transmitted and has failed to complete. The transaction will have failed because a timeout occurred or there were bit errors in the response for three attempts.
0 = No NC interrupt condition

Bit 2 – LPMACK LPM ACK Interrupt Flag bit
When in Device mode:
1= An LPM transaction was received and the USB Module responded with an ACK. This can only occur under following condition: The LPMRESP field in the LPMCNTRL register is set to 0x3, the LPMXMT field is set to 0x1 and there was data pending in the USB module's TX FIFOs.
0 = No ACK interrupt condition
When in Host mode:
1= The LPM transaction is transmitted and the device responds with an ACK
0 = No ACK interrupt condition

Bit 1 – LPMNY LPM NYET Interrupt Flag bit

When in Device mode:

`1`= An LPM transaction is received and the USB Module responded with a NYET. This can only occur under following condition: The LPMRESP field in the LPMCNTRL register is set to 0x3 and the LPMXMT field is set to 0x0.

`0` = No NYET interrupt flag

When in Host mode:

`1`= An LPM transaction is transmitted and the device responded with an NYET

`0`= No NYET interrupt flag

Bit 0 – LPMST LPM Stall Interrupt Flag bit

When in Device mode:

`1`= An LPM transaction was received and the USB Module responded with a STALL. This can only occur under the following condition: The LPMRESP field in the LPMCNTRL register is set to 0x1.

`0` = No Stall condition

When in Host mode:

`1`= An LPM transaction was transmitted and the device responded with a STALL

`0` = No Stall condition

37.7.49. PHY Control Register 00

Name: PHY00
Offset: 0x1500
Reset: 0x00000019
Property: PAC Write-Protection

Note: The USB PHY values must be loaded from the CAL OTP area into the USB PHY registers by software, before enabling the USB, to achieve the specified accuracy.

Table 37-49. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

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

Bits 7:5 – RXPHSSEL[2:0] RX Clock Phase Select
The delay associated with each step is 256ps.

Value	Description
111	Represents the latest phase (7 * 256ps)
110	-
100	-
011	-
010	-
001	-
000	Represents the earliest phase (0 * 256ps)

Bits 4:3 – SLEWRATE[1:0] Adjustment for FS/LS Slew Rate

Value	Description
11	Slowest Slew Rate
10	-
01	-
00	Fastest Slew Rate

Bits 2:0 – PREEMP[2:0] Pre-Emphasis Setting

Value	Description
111	Enable pre-emphasis always
110	Enable pre-emphasis during chirp and non-chirp
100	Enable pre-emphasis during non-chirp
011	Enable pre-emphasis during SOF and EOP and chirp
010	Enable pre-emphasis during chirp
001	Enable pre-emphasis during SOF and EOP
000	Disable pre-emphasis

37.7.50. PHY Control Register 04

Name: PHY04
Offset: 0x1504
Reset: 0x0000008F
Property: PAC Write-Protection

Note: The USB PHY values must be loaded from the CAL OTP area into the USB PHY registers by software, before enabling the USB, to achieve the specified accuracy.

Table 37-50. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
		SQUELCH[2:0]		HIZ	Reserved		TXPHSSEL[2:0]	
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	0	0	0	1	1	1	1

Bits 7:5 – SQUELCH[2:0] Squelch Trigger Point Configuration
Sets the lower 3 bits of the RX squelch trigger point configuration.
Settings include lower bits (PHY04.5:7) and upper bit the upper bit (PHY08.0).

Value	Description
`1111`	200 mV
`1110`	125 mV
`1101`	187.5 mV
`1100`	150 mV (default)
`1011`	175 mV
`1010`	100 mV
`1001`	162.5 mV
`1000`	Reserved
`0111`	Reserved
`0110`	75 mV
`0101`	137 mV

Bit 4 – HIZ

Sets D+/D- to a high impedance state.

Value	Description
1	Enabled
0	Disabled

Bit 3 – Reserved

Bits 2:0 – TXPHSSEL[2:0] TX Clock Phase Select

Value	Description
111	Represents the latest phase (7 * 256ps)
110	-
100	-
011	-
010	-
001	-
000	Represents the earliest phase (0 * 256ps)

37.7.51. PHY Control Register 08

Name: PHY08
Offset: 0x1508
Reset: 0x00000007
Property: PAC Write-Protection

Note: The USB PHY values must be loaded from the CAL OTP area into the USB PHY registers by software, before enabling the USB, to achieve the specified accuracy.

Table 37-51. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

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

Bit 0 – SQUELCH RX Squelch Trigger Point Set
Sets the upper bit of the RX squelch trigger point configuration.
Settings include lower bits (PHY04.5:7) and upper bit the upper bit (PHY08.0).

Value	Description
`1111`	200 mV
`1110`	125 mV
`1101`	187.5 mV
`1100`	150 mV (default)
`1011`	175 mV
`1010`	100 mV
`1001`	162.5 mV
`1000`	Reserved
`0111`	Reserved
`0110`	75 mV
`0101`	137 mV
`0100`	100 mV
`0011`	162.5 mV

Value	Description
0010	87.5 mV
0001	Reserved
0000	112.5 mV

37.7.52. PHY Control Register 0C

Name: PHY0C
Offset: 0x150C
Reset: 0x000000E0
Property: PAC Write-Protection

Note: The USB PHY values must be loaded from the CAL OTP area into the USB PHY registers by software, before enabling the USB, to achieve the specified accuracy.

Table 37-52. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

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

Bits 7:5 – TUNE[2:0] Amplitude Tuning
Sets the lower 3 bits of the HS amplitude tuning.
Settings include the lower bits (PHY0C.5:7) and upper bit the upper bit (PHY10.0:4) – setting of each bit location lowers the amplitude by the same amount regardless of location.

Value	Description
`11111111`	Setting with the smallest amplitude
`10101100`	4-‘0’ and 4-‘1’ is the middle amplitude
`00000000`	Setting with the largest amplitude

37.7.53. PHY Control Register 10

Name: PHY10
Offset: 0x1510
Reset: 0x000000AA
Property: PAC Write-Protection

Note: The USB PHY values must be loaded from the CAL OTP area into the USB PHY registers by software, before enabling the USB, to achieve the specified accuracy.

Table 37-53. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

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

Bits 7:5 – DRVTUNE[2:0] Driver Strength Tuning
Sets the lower 3 bits for the HS/FS/LS driver strength tuning.
Settings include the lower bits (PHY10.5:7) and the upper bit (PHY14.0:1).

Value	Description
`11111`	Fastest rise fall time
`00000`	Slowest rise fall time

Bits 4:0 – TUNE[4:0] Amplitude Tuning
Sets the upper 5 bits of the HS amplitude tuning.
Settings include the lower bits (PHY0C.5:7) and the upper bit the upper bit (PHY10.0:4) – setting of each bit location lowers the amplitude by the same amount regardless of location.

Value	Description
`11111111`	Setting with the smallest amplitude
`10101100`	4-‘0’ and 4-‘1’ is the middle amplitude
`00000000`	Setting with the largest amplitude

37.7.54. PHY Control Register 14

Name: PHY14
Offset: 0x1514
Reset: 0x00000012
Property: PAC Write-Protection

Note: The USB PHY values must be loaded from the CAL OTP area into the USB PHY registers by software, before enabling the USB, to achieve the specified accuracy.

Table 37-54. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

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

Bit 7 – ODT On Die Termination
Sets the lowest bit of the on die termination compensation voltage reference.
Settings include the lower bits (PHY14.8) and the upper bit the upper bits (PHY18.0:1).

Value	Description
111	362.5 mV
110	375 mV
101	387.5 mV
100	450 mV
011	437.5 mV
010	425 mV
001	412.5 mV
000	400 mV

Bit 4 – BYPSSSQUELCH Bypass Squelch Trigger Point
Sets the bypass squelch trigger point configure in chirp mode.

Value	Description
1	Bypass
0	Do not bypass

Bits 2:0 – COMPBYPSS[2:0] Auto-Compression Bypass

Sets the auto-compression bypass.

Settings include the lower bits (PHY0C.5:7) and the upper bit the upper bit (PHY10.0:4) – setting of each bit location lowers the amplitude by the same amount regardless of location.

Value	Description
11	Disable current and disable ODT auto-calibration
10	Disable current and enable ODT auto-calibration
01	Enable current and disable ODT auto-calibration
00	Enable current and ODT auto-calibration

Bits 1:0 – DRVTUNE[1:0] HS/FS/LS Driver Strength Tuning

Sets the upper 2 bits for HS/FS/LS driver strength tuning.

Settings include the lower bits (PHY10.5:7) and the upper bit (PHY14.0:1).

Value	Description
11111	Fastest rise fall time
00000	Slowest rise fall time

37.7.55. PHY Control Register 18

Name: PHY18
Offset: 0x1518
Reset: 0x00000008
Property: PAC Write-Protection

Note: The USB PHY values must be loaded from the CAL OTP area into the USB PHY registers by software, before enabling the USB, to achieve the specified accuracy.

Table 37-55. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

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

Bits 1:0 – ODT[1:0] On Die Termination
Sets the lowest bit of the on die termination compensation voltage reference.
Settings include the lower bits (PHY14.7) and the upper bits (PHY18.0:1).

Value	Description
`111`	362.5 mV
`110`	375 mV
`101`	387.5 mV
`100`	450 mV
`011`	437.5 mV
`010`	425 mV
`001`	412.5 mV
`000`	400 mV

37.7.56. PHY Control Register 1C

Name: PHY1C
Offset: 0x151C
Reset: 0x00000082
Property: PAC Write-Protection

Note: The USB PHY values must be loaded from the CAL OTP area into the USB PHY registers by software, before enabling the USB, to achieve the specified accuracy.

Table 37-56. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

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

Bit 7 – FSLSDIFF FS/LS Differential Receiver
Turns off FS/LS differential receiver in suspend mode.

Value	Description
1	On
0	Off

Bit 1 – ODTBYPASS ODT Auto-Refresh Bypass
Sets the ODT auto-refresh bypass.

Value	Description
1	Bypass
0	Do not bypass

37.7.57. PHY Control Register 20

Name: PHY20
Offset: 0x1520
Reset: 0x00000080
Property: PAC Write-Protection

Note: The USB PHY values must be loaded from the CAL OTP area into the USB PHY registers by software, before enabling the USB, to achieve the specified accuracy.

Table 37-57. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

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

Bits 7:6 – HSSLEW[1:0] HS Slew Rate
Sets the HS slew rate.
Settings include the lower bits (PHY20.6:7) and the upper bits (PHY24.0).

Value	Description
111	Fastest rise/fall time
010	Middle slew rate
001	Slowest rise/fall time
000	Reserved

37.7.58. PHY Control Register 24

Name: PHY24
Offset: 0x1524
Reset: 0x0000000C
Property: PAC Write-Protection

Note: The USB PHY values must be loaded from the CAL OTP area into the USB PHY registers by software, before enabling the USB, to achieve the specified accuracy.

Table 37-58. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	HSDRIVST[1:0]		HSPREEMPST[2:0]			PREEMPHEN	OTGPDN	HSSLEW
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	1	1	0	0

Bits 7:6 – HSDRIVST[1:0] HS Transmit Driver Strength
Sets the HS transmit driver strength.
Settings include the lower bits (PHY24.6:7) and the upper bit (PHY28.0).

Value	Description
111	Strongest drive strength
000	Weakest drive strength

Bits 5:3 – HSPREEMPST[2:0] HS Transmit Pre-Emphasis Strength
Sets the HS transmit pre-emphasis strength.

Value	Description
11	Slowest Slew Rate
10	-
01	-
00	Fastest Slew Rate

Bit 2 – PREEMPHEN HS Transmit Pre-Emphasis Enable
Enable half-bit pre-emphasis for HS transmit.

Value	Description
1	Enable
0	Disable

Bit 1 – OTGPDN ODT Power Down

Sets the ODT power down.

Value	Description
1	On
0	Off

Bit 0 – HSSLEW HS Slew Rate

Sets the HS slew rate.

Settings include the lower bits (PHY20.6:7) and the upper bit (PHY24.0).

Value	Description
111	Fastest rise/fall time
010	Middle slew rate
001	Slowest rise/fall time
000	Reserved

37.7.59. PHY Control Register 28

Name: PHY28
Offset: 0x1528
Reset: 0x0000001B
Property: PAC Write-Protection

Note: The USB PHY values must be loaded from the CAL OTP area into the USB PHY registers by software, before enabling the USB, to achieve the specified accuracy.

Table 37-59. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

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

Bits 7:5 – HSDRVCOMP[2:0] HS Driver Current Compensation
Sets the HS driver current compensation voltage reference.

Value	Description
`111`	362.5 mV
`110`	375 mV
`101`	387.5 mV
`100`	450 mV
`011`	437.5 mV
`010`	425 mV
`001`	412.5 mV
`000`	400 mV

Bits 4:1 – DISCONDET[3:0] HOST Disconnect Detection
Sets the HOST disconnect detection trigger point.

Value	Description
`1111`	Reserved
`1110`	612.5 mV

Value	Description
1101	650 mV
1100	Reserved
1011	Reserved
1010	58705 mV
1001	Reserved
1000	600 mV
0111	Reserved
0110	537.5 mV
0101	Reserved
0100	550 mV
0011	625 mV
0010	562.5 mV
0001	600 mV
0000	575 mV

Bit 0 – HSDRIVST HS Transmit Driver Strength

Sets the HS transmit driver strength.

Settings include the lower bits (PHY24.6:7) and the upper bit (PHY28.0).

Value	Description
111	Strongest drive strength
000	Weakest drive strength

37.7.60. PHY Control Register 44

Name: PHY44
Offset: 0x1544
Reset: 0x00000040
Property: PAC Write-Protection

Note: The USB PHY values must be loaded from the CAL OTP area into the USB PHY registers by software, before enabling the USB, to achieve the specified accuracy.

Table 37-60. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	FRCSESSEND				FRCVBUSVAL	DIGDBG	PLLDAMP	
Access	R/W				R/W	R/W	R/W	
Reset	0				0	0	0	

Bit 7 – FRCSESSEND Force Session End

Value	Description
1	-
0	Default

Bit 3 – FRCVBUSVAL Force Output VBUS_VALID

Value	Description
1	-
0	Default

Bit 2 – DIGDBG Digital Debug Interface (Reserved)

Value	Description
1	-
0	Default

Bit 1 – PLLDAMP Digital Debug Interface (Reserved)

Value	Description
1	Decreased PLL damping factor
0	Increased PLL damping factor (default)

37.7.61. PHY Control Register 48

Name: PHY48
Offset: 0x1548
Reset: 0x00000004
Property: PAC Write-Protection

Note: The USB PHY values must be loaded from the CAL OTP area into the USB PHY registers by software, before enabling the USB, to achieve the specified accuracy.

Table 37-61. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	SESSENDTUNE[2:0]					VBUSCHRGE	FRCBSESSVAL	FRCASESSVAL
Access	R/W	R/W	R/W			R/W	R/W	R/W
Reset	0	0	0			1	0	0

Bits 7:5 – SESSENDTUNE[2:0] Session End Reference Tuning

Value	Description
111	300 mV
110	650 mV
101	600 mV
100	550 mV
011	350 mV
010	400 mV
001	450 mV
000	500 mV

Bit 2 – VBUSCHRGE VBUS Charging/Discharging Bypass

Value	Description
1	Default
0	-

Bit 1 – FRCBSESSVAL Force B Session Valid

Value	Description
1	-
0	Default

Bit 0 – FRCASESSVAL Force A Session Valid

Value	Description
1	-
0	Default

37.7.62. PHY Control Register 4C

Name: PHY4C
Offset: 0x154C
Reset: 0x00000000
Property: PAC Write-Protection

Note: The USB PHY values must be loaded from the CAL OTP area into the USB PHY registers by software, before enabling the USB, to achieve the specified accuracy.

Table 37-62. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

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

Bits 7:6 – BSESSVALIDTUNE[1:0] B Session Valid Reference Tune
Settings include the lower bits (PHY4C.6:7) and the upper bit (PHY50.0).

Value	Description
111	2.16 V
110	2.58 V
101	2.52 V
100	2.46 V
011	2.22 V
010	2.28 V
001	2.34 V
000	2.4 V (Default)

Bits 2:0 – VBUSVALTUNE[2:0] VBUS Valid Reference Tune

Value	Description
111	4.3 V
110	4.65 V
101	4.6 V

Value	Description
100	4.55 V
011	4.3 V
010	4.4 V
001	4.5 V
000	4.45 V

37.7.63. PHY Control Register 50

Name: PHY50
Offset: 0x1550
Reset: 0x00000000
Property: PAC Write-Protection

Note: The USB PHY values must be loaded from the CAL OTP area into the USB PHY registers by software, before enabling the USB, to achieve the specified accuracy.

Table 37-63. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

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

Bits 7:5 – COMPCURREF[2:0] Compensation Current Tuning Reference

Value	Description
111	162.5 mV
110	175 mV
101	212.5 mV
100	250 mV
011	237.5 mV
010	225 mV
001	187.5 mV
000	200 mV

Bits 3:1 – ASESSVALIDTUNE[2:0] A Session Valid Reference Tune

Value	Description
111	1.2 V
110	1.55 V
101	1.5 V

Value	Description
100	1.45 V
011	1.25 V
010	1.3 V
001	1.35 V
000	1.4 V (Default)

Bit 0 – BSESSVALIDTUNE B Session Valid Reference Tune
Settings include the lower bits (PHY4C.6:7) and the upper bit (PHY50.0).

Value	Description
111	2.16 V
110	2.58 V
101	2.52 V
100	2.46 V
011	2.22 V
010	2.28 V
001	2.34 V
000	2.4 V (Default)

37.8. Register Summary: USB Endpoint0 Common Registers

For descriptions and definitions of both Register and bitfield properties, refer to Register Properties.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00 ... 0x1017	Reserved									
0x1018	COUNT0	7:0		ENDPOINT0RXCOUNT[6:0]						
0x1018	RXCOUNT	15:8			ENDPOINTRXCOUNT[13:8]					
		7:0	ENDPOINTRXCOUNT[7:0]							
0x101A ... 0x101E	Reserved									
0x101F	CONFIGDATA	7:0	MPRXE	MPTXE	BIGENDIAN	HBRXE	HBTXE	DYNOFIFOSIZING	SOFTCONE	UTMIDATAWIDTH

37.8.1. Endpoint0 RX Count Register

Name: COUNT0
Offset: 0x1018
Reset: 0x0000
Property: Read Only

Count0 is a read-only register that indicates the number of received data bytes in the Endpoint 0 FIFO. The value returned changes as the contents of the FIFO change and is only valid while RxPktRdy (CSR0.D0) is set.

Table 37-64. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
		ENDPOINT0RXCOUNT[6:0]						
Access		R	R	R	R	R	R	R
Reset		0	0	0	0	0	0	0

Bits 6:0 – ENDPOINT0RXCOUNT[6:0]
The number of received data bytes in the Endpoint 0 FIFO. The value returned changes as the contents of the FIFO change and is only valid while the RXPKTRDY bit is set.

37.8.2. USB Configuration Data Register

Name: CONFIGDATA
Offset: 0x101F
Reset: 0x0000
Property: Read-Only

Table 37-65. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
	MPRXE	MPTXE	BIGENDIAN	HBRXE	HBTXE	DYNOFIFOSIZING	SOFTCONE	UTMIDATAWIDTH
Access	R	R	R	R	R	R	R	R
Reset	x	x	0	x	x	x	1	0

Bit 7 – MPRXE Automatic Amalgamation Option bit

Value	Description
0	No automatic amalgamation
1	Automatic amalgamation of bulk packets is done

Bit 6 – MPTXE Automatic Splitting Option bit

Value	Description
0	No automatic splitting
1	Automatic splitting of bulk packets is done

Bit 5 – BIGENDIAN Byte Ordering Option bit

Value	Description
0	Little Endian ordering
1	Big Endian ordering

Bit 4 – HBRXE High-Bandwidth RX ISO Option bit

Value	Description
0	No High-bandwidth RX ISO support
1	High-bandwidth RX ISO endpoint support is selected

Bit 3 – HBTXE High-Bandwidth TX ISO Option bit

Value	Description
0	No High-bandwidth TX ISO support
1	High-bandwidth TX ISO endpoint support is selected

Bit 2 – DYNOFIFOSIZING Dynamic FIFO Sizing Option bit

Value	Description
0	No Dynamic FIFO sizing
1	Dynamic FIFO sizing is supported

Bit 1 – SOFTCONE Soft Connect/Disconnect Option bit

Value	Description
0	Soft Connect/Disconnect is not supported
1	Soft Connect/Disconnect is supported

Bit 0 – UTMIDATAWIDTH UTMI+ Data Width Option bit
Always '0', indicating 8-bit UTMI+ data width.

37.8.3. Endpoint 1-7 RX Count Registers

Name: RXCOUNT
Offset: 0x1018
Reset: 0x0000
Property: Read Only

RXCOUNT is a read-only register that holds the number of data bytes in the packet currently in line to be read from the RXFIFO. If the packet was transmitted as multiple bulk packets, the number given will be for the combined packet.

Note: The value returned changes as the FIFO is unloaded and is only valid while RXPKTRDY (RXCSR.D0) is set.

Table 37-66. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	15	14	13	12	11	10	9	8
			ENDPOINTRXCOUNT[13:8]					
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

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

Bits 13:0 – ENDPOINTRXCOUNT[13:0]

The number of received data bytes in the Endpoint 0 FIFO. The value returned changes as the contents of the FIFO change and is only valid while the RXPKTRDY bit is set.

37.9. Register Summary: USB Host Mode Only Registers

For descriptions and definitions of both Register and bitfield properties, refer to Register Properties.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00 ... 0x107F	Reserved									
0x1080	TXFUNCADDREP0	7:0		TXFUNCADDR[6:0]						
0x1081	Reserved									
0x1082	TXHUBADDREP0	7:0	MULLTRANS	TXHUBADDR[6:0]						
0x1083	TXHUBPORT0	7:0		HUBPORT[6:0]						
0x1084	RXFUNCADDREP0	7:0		RXFUNCADDR[6:0]						
0x1085	Reserved									
0x1086	RXHUBADDREP0	7:0	MULLTRANS	RXHUBADDR[6:0]						
0x1087	RXHUBPORT0	7:0		HUBPORT[6:0]						
0x1088	TXFUNCADDREP1	7:0		TXFUNCADDR[6:0]						
0x1089	Reserved									
0x108A	TXHUBADDREP1	7:0	MULLTRANS	TXHUBADDR[6:0]						
0x108B	TXHUBPORT1	7:0		HUBPORT[6:0]						
0x108C	RXFUNCADDREP1	7:0		RXFUNCADDR[6:0]						
0x108D	Reserved									
0x108E	RXHUBADDREP1	7:0	MULLTRANS	RXHUBADDR[6:0]						
0x108F	RXHUBPORT1	7:0		HUBPORT[6:0]						
0x1090	TXFUNCADDREP2	7:0		TXFUNCADDR[6:0]						
0x1091	Reserved									
0x1092	TXHUBADDREP2	7:0	MULLTRANS	TXHUBADDR[6:0]						
0x1093	TXHUBPORT2	7:0		HUBPORT[6:0]						
0x1094	RXFUNCADDREP2	7:0		RXFUNCADDR[6:0]						
0x1095	Reserved									
0x1096	RXHUBADDREP2	7:0	MULLTRANS	RXHUBADDR[6:0]						
0x1097	RXHUBPORT2	7:0		HUBPORT[6:0]						
0x1098	TXFUNCADDREP3	7:0		TXFUNCADDR[6:0]						
0x1099	Reserved									
0x109A	TXHUBADDREP3	7:0	MULLTRANS	TXHUBADDR[6:0]						
0x109B	TXHUBPORT3	7:0		HUBPORT[6:0]						
0x109C	RXFUNCADDREP3	7:0		RXFUNCADDR[6:0]						
0x109D	Reserved									
0x109E	RXHUBADDREP3	7:0	MULLTRANS	RXHUBADDR[6:0]						
0x109F	RXHUBPORT3	7:0		HUBPORT[6:0]						
0x10A0	TXFUNCADDREP4	7:0		TXFUNCADDR[6:0]						
0x10A1	Reserved									
0x10A2	TXHUBADDREP4	7:0	MULLTRANS	TXHUBADDR[6:0]						
0x10A3	TXHUBPORT4	7:0		HUBPORT[6:0]						
0x10A4	RXFUNCADDREP4	7:0		RXFUNCADDR[6:0]						
0x10A5	Reserved									
0x10A6	RXHUBADDREP4	7:0	MULLTRANS	RXHUBADDR[6:0]						
0x10A7	RXHUBPORT4	7:0		HUBPORT[6:0]						
0x10A8	TXFUNCADDREP5	7:0		TXFUNCADDR[6:0]						
0x10A9	Reserved									
0x10AA	TXHUBADDREP5	7:0	MULLTRANS	TXHUBADDR[6:0]						
0x10AB	TXHUBPORT5	7:0		HUBPORT[6:0]						
0x10AC	RXFUNCADDREP5	7:0		RXFUNCADDR[6:0]						
0x10AD	Reserved									
0x10AE	RXHUBADDREP5	7:0	MULLTRANS	RXHUBADDR[6:0]						
0x10AF	RXHUBPORT5	7:0		HUBPORT[6:0]						
0x10B0	TXFUNCADDREP6	7:0		TXFUNCADDR[6:0]						
0x10B1	Reserved									
0x10B2	TXHUBADDREP6	7:0	MULLTRANS	TXHUBADDR[6:0]						
0x10B3	TXHUBPORT6	7:0		HUBPORT[6:0]						
0x10B4	RXFUNCADDREP6	7:0		RXFUNCADDR[6:0]						
0x10B5	Reserved									
0x10B6	RXHUBADDREP6	7:0	MULLTRANS	RXHUBADDR[6:0]						

Register Summary: USB Host Mode Only Registers (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x10B7	RXHUBPORT6	7:0		HUBPORT[6:0]						
0x10B8	TXFUNCADDREP7	7:0		TXFUNCADDR[6:0]						
0x10B9	Reserved									
0x10BA	TXHUBADDREP7	7:0	MULTTRANS	TXHUBADDR[6:0]						
0x10BB	TXHUBPORT7	7:0		HUBPORT[6:0]						
0x10BC	RXFUNCADDREP7	7:0		RXFUNCADDR[6:0]						
0x10BD	Reserved									
0x10BE	RXHUBADDREP7	7:0	MULTTRANS	RXHUBADDR[6:0]						
0x10BF	RXHUBPORT7	7:0		HUBPORT[6:0]						
0x10C0 ... 0x1364	Reserved									
0x1365	LPMFADDR	7:0		FUNCADDR[6:0]						

37.9.1. USB Transmit Endpoint n Function Address Register

Name: TXFUNCADDREP
Offset: 0x1080 + n*0x08 [n=0..7]
Reset: 0x0000
Property: Read/Write

Table 37-67. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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

Bits 6:0 – TXFUNCADDR[6:0] TX Functional Address Bits
Specifies the address for the target function that is be accessed through the associated endpoint. It needs to be defined for each TX endpoint that is used.

37.9.2. USB Transmit Endpoint n Hub Address Register

Name: TXHUBADDREP
Offset: 0x1082 + n*0x08 [n=0..7]
Reset: 0x0000
Property: Read/Write

Table 37-68. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
	MULTTRANS	TXHUBADDR[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 – MULTTRANS TX Hub Multiple Translators Bit

Value	Description
0	The USB 2.0 hub has a single transaction translator
1	The USB 2.0 hub has multiple transaction translators

Bits 6:0 – TXHUBADDR[6:0] TX Hub Address Bits
When a Low-Speed or Full-Speed device is connected to this endpoint via a Hi-Speed USB 2.0 hub, these bits record the address of the USB 2.0 hub.

37.9.3. USB Transmit Endpoint n Hub Port Register

Name: TXHUBPORT
Offset: 0x1083 + n*0x08 [n=0..7]
Reset: 0x0000
Property: Read/Write

Table 37-69. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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

Bits 6:0 – HUBPORT[6:0] TX Hub Port Bits
When a Low-Speed or Full-Speed device is connected to this endpoint via a Hi-Speed USB 2.0 hub, this field records the port number of that USB 2.0 hub.

37.9.4. USB Receive Endpoint n Function Address Register

Name: RXFUNCADDREP
Offset: 0x1084 + n*0x08 [n=0..7]
Reset: 0x0000
Property: Read/Write

Table 37-70. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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

Bits 6:0 – RXFUNCADDR[6:0] RX Functional Address Bits
Specifies the address for the target function that is be accessed through the associated endpoint. It needs to be defined for each RX endpoint that is used.

37.9.5. USB Recieve Endpoint n Hub Address Register

Name: RXHUBADDREP
Offset: 0x1086 + n*0x08 [n=0..7]
Reset: 0x0000
Property: Read/Write

Table 37-71. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
	MULTTRANS	RXHUBADDR[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 – MULTTRANS RX Hub Multiple Translators Bit

Value	Description
0	The USB 2.0 hub has a single transaction translator
1	The USB 2.0 hub has multiple transaction translators

Bits 6:0 – RXHUBADDR[6:0] RX Hub Address Bits
When a Low-Speed or Full-Speed device is connected to this endpoint via a Hi-Speed USB 2.0 hub, these bits record the address of the USB 2.0 hub.

37.9.6. USB Receive Endpoint n Hub Port Register

Name: RXHUBPORT
Offset: 0x1087 + n*0x08 [n=0..7]
Reset: 0x0000
Property: Read/Write

Table 37-72. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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

Bits 6:0 – HUBPORT[6:0] RX Hub Port Bits
When a Low-Speed or Full-Speed device is connected to this endpoint via a Hi-Speed USB 2.0 hub, this field records the port number of that USB 2.0 hub.

37.9.7. USB Function Address in LPM Payload Register

Name: LPMFADDR
Offset: 0x1365
Reset: 0x0000
Property: PAC Write-Protection

Table 37-73. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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

Bits 6:0 – FUNCADDR[6:0] Function Address bits.
Function address that will be placed in the LPM payload.

37.10. Register Summary: Home Mode Endpoint0 Registers

For descriptions and definitions of both Register and bitfield properties, refer to Register Properties.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00 ... 0x1011	Reserved									
0x1012	CSR0L	7:0	NAKTMOUT	STATUSPKT	REQPKT	ERROR	SETUPPKT	RXSTALL	TXPKTRDY	RXPKTRDY
0x1013	CSR0H	7:0					DISPING	DTWREN	DATATGGL	FLSHFIFO
0x1014 ... 0x1019	Reserved									
0x101A	TYPE0	7:0	SPEED[1:0]							
0x101B	NAKLIMIT0	7:0				EP0NAKLIMIT[4:0]				

37.10.1. USB Control Status Register Low for Endpoint0

Name: CSR0L
Offset: 0x1012
Reset: 0x0000
Property: PAC Write-Protection

Table 37-74. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
	NAKTMOUT	STATUSPKT	REQPKT	ERROR	SETUPPKT	RXSTALL	TXPKTRDY	RXPKTRDY
Access	R/W/HC	R/W/HC	R/W/HC	R/W/HS	R/W/HS	R/W/HS	R	R
Reset	0	0	0	0	0	0	0	0

Bit 7 – NAKTMOUT NAK Time-out Control bit

Value	Description
0	Allow the endpoint to continue
1	Endpoint 0 is halted following the receipt of NAK responses for longer than the time set by the NAKLIM<4:0> bits (USBICSR<28:24>)

Bit 6 – STATUSPKT Status Stage Transaction Control Bit

Value	Description
0	Do not perform a status stage transaction
1	When set at the same time as the TXPKTRDY or REQPKT bit is set, performs a status stage transaction

Bit 5 – REQPKT IN Transaction Request Control Bit

Value	Description
0	Do not request an IN transaction
1	Request an IN transaction. This bit is cleared when the RXPKTRDY bit is set.

Bit 4 – ERROR No Response Error Status bit

Value	Description
0	Clear this flag. Software must write a '0' to this bit to clear it.
1	Three attempts have been made to perform a transaction with no response from the peripheral. An interrupt is generated.

Bit 3 – SETUPPKT Definition bit

Value	Description
0	Normal OUT token for the transaction bit 19
1	When set at the same time as the TXPKTRDY bit is set, send a SETUP token instead of an OUT token for the transaction. This also clears the Data Toggle.

Bit 2 – RXSTALL STALL handshake received Status bit

Value	Description
0	Software clear of bit
1	STALL handshake was received

Bit 1 – TXPKTRDY TX Packet Ready Control bit

Value	Description
0	No data packet is ready for transmit
1	Data packet has been loaded into the FIFO. It is cleared automatically.

Bit 0 – RXPKTRDY RX Packet Ready Status bit

This bit is cleared by setting the SVCRPR bit.

Value	Description
0	No data packet has been received
1	Data packet has been received. Interrupt is generated (when enabled) when this bit is set.

37.10.2. USB Control Status Register High for Endpoint0

Name: CSR0H
Offset: 0x1013
Reset: 0x0000
Property: PAC Write-Protection

Table 37-75. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
					DISPING	DTWREN	DATATGGL	FLSHFIFO
Access					R/W	R/W/HC	R/W	R/W/HC
Reset					0	0	0	0

Bit 3 – DISPING Disable Ping tokens control bit

Value	Description
0	Ping tokens are issued
1	USB Module will not issue PING tokens in data and status phases of a Hi-Speed Control transfer

Bit 2 – DTWREN Data Toggle Write Enable bit

Value	Description
0	Disable data toggle write
1	Enable the current state of the Endpoint 0 data toggle to be written. Automatically cleared.

Bit 1 – DATATGGL Data Toggle bit
When read, this bit indicates the current state of the Endpoint 0 data toggle.
If DTWREN = 1, this bit is writable with the desired setting.
If DTWREN = 0, this bit is read-only.

Bit 0 – FLSHFIFO Flush FIFO Control bit

Value	Description
0	No Flush operation
1	Flush the next packet to be transmitted/read from the Endpoint 0 FIFO. The FIFO pointer is reset and theTXPKTRDY/RXPKTRDY bit is cleared. Automatically cleared when the operation completes. Should only be used when TXPKTRDY/RXPKTRDY = 1.

37.10.3. Endpoint0 Operating Speed Registers

Name: TYPE0
Offset: 0x101A
Reset: 0x0000
Property: PAC Write-Protection

This register defines the speed of the Endpoint 0.

Table 37-76. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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

Bits 7:6 – SPEED[1:0] Operating Speed Control bits.

Value	Description
11	Low-Speed
10	Full-Speed
01	Hi-Speed
00	Reserved

37.10.4. Endpoint0 NAK Response Limit Registers

Name: NAKLIMIT0
Offset: 0x101B
Reset: 0x0000
Property: PAC Write-Protection

Table 37-77. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
				EP0NAKLIMIT[4:0]				
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bits 4:0 – EP0NAKLIMIT[4:0] Endpoint0 NAK Limit bits.
The number of frames/microframes (Hi-Speed transfers) after which Endpoint 0 should time-out on receiving a stream of NAK responses.

37.11. Register Summary: USB Host Mode Endpoint1-7 Registers

For descriptions and definitions of both Register and bitfield properties, refer to Register Properties.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00 ... 0x1011	Reserved									
0x1012	TXCSRL	7:0	NAKTMOUT	CLRDATATOG	RXSTALL	SETUPPKT	FLUSHFIFO	ERROR	FIFONOTEMPTY	TXPKTRDY
0x1013	TXCSRH	7:0	AUTOSET		MODE	DMAREQENAB	FRCDATATOG	DMAREQMODE	DATATOGGLEWRENABLE	DATATOGGLE
0x1014 ... 0x1015	Reserved									
0x1016	RXCSRL	7:0	CLRDATATOG	RXSTALL	REQPKT	FLUSHFIFO	NAKTIMEOUT	ERROR	FIFOFULL	RXPKTRDY
0x1017	RXCSRH	7:0	AUTOCLEAR	AUTOREQ	DMAREQENAB	PIDERROR	DMAREQMODE	DATATOGGLEWRENABLE	DATATOGGLE	INCOMPRX
0x1018 ... 0x1019	Reserved									
0x101A	TXTYPE	7:0	SPEED[1:0]		PROTOCOL[1:0]		ENDPOINTNUMBER[3:0]			
0x101B	TXINTERVAL	7:0	TXPOLLINGINTERVAL[7:0]							
0x101C	RXTYPE	7:0	SPEED[1:0]		PROTOCOL[1:0]		ENDPOINTNUMBER[3:0]			
0x101D	RXINTERVAL	7:0	RXPOLLINGINTERVAL[7:0]							
0x101E ... 0x12FF	Reserved									
0x1300	USBE1RPC	31:24								
		23:16								
		15:8	RQPKTCNT[15:8]							
		7:0	RQPKTCNT[7:0]							
0x1304	USBE2RPC	31:24								
		23:16								
		15:8	RQPKTCNT[15:8]							
		7:0	RQPKTCNT[7:0]							
0x1308	USBE3RPC	31:24								
		23:16								
		15:8	RQPKTCNT[15:8]							
		7:0	RQPKTCNT[7:0]							
0x130C	USBE4RPC	31:24								
		23:16								
		15:8	RQPKTCNT[15:8]							
		7:0	RQPKTCNT[7:0]							
0x1310	USBE5RPC	31:24								
		23:16								
		15:8	RQPKTCNT[15:8]							
		7:0	RQPKTCNT[7:0]							
0x1314	USBE6RPC	31:24								
		23:16								
		15:8	RQPKTCNT[15:8]							
		7:0	RQPKTCNT[7:0]							
0x1318	USBE7RPC	31:24								
		23:16								
		15:8	RQPKTCNT[15:8]							
		7:0	RQPKTCNT[7:0]							

37.11.1. TX Control Status Register Low for Endpoint 1-7

Name: TXCSRL
Offset: 0x1012
Reset: 0x0000
Property: PAC Write-Protection

Table 37-78. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
	NAKTMOUT	CLRDATATOG	RXSTALL	SETUPPKT	FLUSHFIFO	ERROR	FIFONOTEMPTY	TXPKTRDY
Access	R/W/HS	R/W/HC	R/W	R/W	R/W	R/W/HC	R/W	R/W/HC
Reset	0	0	0	0	0	0	0	0

Bit 7 – NAKTMOUT NAK Time-out Status bit

Value	Description
0	Written by software to clear this bit
1	TX endpoint is halted following the receipt of NAK responses for longer than the NAKLIM setting

Bit 6 – CLRDATATOG Clear Data Toggle Control Bit

Value	Description
0	Do not clear the data toggle
1	Resets the endpoint data toggle to 0

Bit 5 – RXSTALL Stall Receipt Bit

Value	Description
0	Written by software to clear this bit
1	STALL handshake is received. Any DMA request in progress is stopped, the FIFO is completely flushed and the TXPKTRDY bit is cleared.

Bit 4 – SETUPPKT Definition bit

Value	Description
0	Normal OUT token for the transaction
1	When set at the same time as the TXPKTRDY bit is set, send a SETUP token instead of an OUT token for the transaction. This also clears the Data Toggle.

Bit 3 – FLUSHFIFO FIFO Flush Control bit

Value	Description
0	Do not flush the FIFO
1	Flush the latest packet from the endpoint TX FIFO. The FIFO pointer is reset, the TXPKTRDY bit is cleared and an interrupt is generated.

Bit 2 – ERROR Handshake Failure Status bit

Value	Description
0	Written by software to clear this bit.
1	Three attempts have been made to send a packet and no handshake packet has been received

Bit 1 – FIFONOTEMPTY FIFO Not Empty Status bit

Value	Description
0	TX FIFO is empty
1	There is at least 1 packet in the TX FIFO

Bit 0 – TXPKTRDY TX Packet Ready Control bit

The software sets this bit after loading a data packet into the FIFO. It is cleared automatically when a data packet has been transmitted. This bit is also automatically cleared prior to loading a second packet into a double-buffered FIFO.

37.11.2. TX Control Status Register High for Endpoint 1-7

Name: TXCSRH
Offset: 0x1013
Reset: 0x0000
Property: PAC Write-Protection

Table 37-79. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
	AUTOSET		MODE	DMAREQENAB	FRCDATATOG	DMAREQMODE	DATATOGGLEWRENABLE	DATATOGGLE
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 – AUTOSET Auto Set Control bit

Value	Description
0	TXPKTRDY must be set manually for all packet sizes
1	TXPKTRDY will be automatically set when data of the maximum packet size (value in TXMAXP) is loaded into the TX FIFO. If a packet of less than the maximum packet size is loaded, then TXPKTRDY will have to be set manually.

Bit 5 – MODE Endpoint Direction Control Bit

Value	Description
0	Endpoint is RX
1	Endpoint is TX

Bit 4 – DMAREQENAB Endpoint DMA Request Enable bit

Value	Description
0	DMA requests are disabled for this endpoint
1	DMA requests are enabled for this endpoint

Bit 3 – FRCDATATOG Force Endpoint Data Toggle Control bit

Value	Description
0	No forced behavior
1	Forces the endpoint data toggle to switch and the data packet to be cleared from the FIFO, regardless of whether an ACK was received.

Bit 2 – DMAREQMODE Endpoint DMA Request Mode Control bit

Value	Description
0	DMA Request Mode0
1	DMA Request Mode1

Bit 1 – DATATOGGLEWRENABLE Data Toggle Write Enable bit

Value	Description
0	Disables writing the DATATOGGLE bit
1	Enable the current state of the TX Endpoint data toggle (DATATOGGLE) to be written

Bit 0 – DATATOGGLE Data Toggle Control bit

When read, this bit indicates the current state of the TX Endpoint data toggle.
If DATATOGGLEWRENABLE = 1, this bit may be written with the required setting of the data toggle.
If DATATOGGLEWRENABLE = 0, any value written to this bit is ignored.

37.11.3. RX Control Status Register Low for Endpoint 1-7

Name: RXCSRL
Offset: 0x1016
Reset: 0x0000
Property: PAC Write-Protection

Table 37-80. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
	CLRDATATOG	RXSTALL	REQPKT	FLUSHFIFO	NAKTIMEOUT	ERROR	FIFOFULL	RXPKTRDY
Access	R/W/HC	R/W/HS	R/W	R/W/HC	R/W/HS	R/W/HS	R/W/HC	R/W/HS
Reset	0	0	0	0	0	0	0	0

Bit 7 – CLRDATATOG Clear Data Toggle Control bit

Value	Description
0	Leave endpoint data toggle alone
1	Reset the endpoint data toggle to 0

Bit 6 – RXSTALL Stall Handshake Receive Status Bit

Value	Description
0	Written by software to clear this bit
1	STALL handshake is received. An interrupt is generated.

Bit 5 – REQPKT IN Transaction Request Control bit
This bit is cleared when RXPKTRDY is set.

Value	Description
0	No request
1	Request an IN transaction.

Bit 4 – FLUSHFIFO FIFO Flush Control bit
This bit is automatically cleared.

Value	Description
0	Normal FIFO operation
1	Flush the next packet to be read from the endpoint RX FIFO. The FIFO pointer is reset and the RXPKTRDY bit is cleared. This should only be used when RXPKTRDY is set. If the FIFO is double- buffered, FLUSH may need to be set twice to completely clear the FIFO.

Bit 3 – NAKTIMEOUT Data Error/NAK Time-out Status bit (Host mode)

Value	Description
0	No data or NAK time-out error
1	The data packet has a CRC or bit-stuff error. In Bulk mode, the RX endpoint is halted following the receipt of NAK responses for longer than the time set as the NAK limit.

Bit 2 – ERROR No Data Packet Received Status bit
This bit is only valid when the RX endpoint is operating in Bulk or Interrupt mode. In ISO mode, it always returns zero.

Value	Description
0	Written by software to clear this bit.
1	Three attempts have been made to receive a packet and no data packet has been received. An interrupt is generated.

Bit 1 – FIFOFULL FIFO Full Status bit

Value	Description
0	The RX FIFO has at least one free space
1	No more packets can be loaded into the RX FIFO

Bit 0 – RXPKTRDY Data Packet Reception Status bit

Value	Description
0	Written by software to clear this bit when the packet has been unloaded from the RX FIFO.
1	A data packet has been received. An interrupt is generated.

37.11.4. RX Control Status Register High for Endpoint 1-7

Name: RXCSRH
Offset: 0x1017
Reset: 0x0000
Property: PAC Write-Protection

Table 37-81. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
	AUTOCLEAR	AUTOREQ	DMAREQENAB	PIDERROR	DMAREQMODE	DATATOGGLEWRENABLE	DATATOGGLE	INCOMPRX
Access	R/W	R/W	R/W	R/W	R/W	R/W/HC	R	R/W
Reset	0	0	0	0	0	0	0	0

Bit 7 – AUTOCLEAR RXPKTRDY Automatic Clear Control bit
This bit should not be set for high-bandwidth Isochronous endpoints.

Value	Description
0	No automatic clearing of RXPKTRDY
1	RXPKTRDY will be automatically cleared when a packet of RXMAXP bytes has been unloaded from the RXFIFO. When packets of less than the maximum packet size are unloaded, RXPKTRDY will have to be cleared manually. When using a DMA to unload the RX FIFO, data is read from the RX FIFO in 4-byte chunks regardless of the RXMAXP.

Bit 6 – AUTOREQ Automatic Packet Request Control bit
This bit is automatically cleared when a short packet is received.

Value	Description
0	No automatic packet request
1	REQPKT will be automatically set when RXPKTRDY bit is cleared.

Bit 5 – DMAREQENAB DMA Request Enable Control bit

Value	Description
0	Disable DMA requests for the RX endpoint.
1	Enable DMA requests for the RX endpoint.

Bit 4 – PIDERROR PID Error Status bit

Value	Description
0	No error
1	In ISO transactions, this indicates a PID error in the received packet.

Bit 3 – DMAREQMODE DMA Request Mode Selection bit

Value	Description
0	DMA Request Mode 0
1	DMA Request Mode 1

Bit 2 – DATATOGGLEWRENABLE Data Toggle Write Enable Control bit

Value	Description
0	DATATGGL is not writable
1	DATATGGL can be written

Bit 1 – DATATOGGLE Data Toggle bit

When read, this bit indicates the current state of the endpoint data toggle.
If DATATWEN = 1, this bit may be written with the required setting of the data toggle.
If DATATWEN = 0, any value written to this bit is ignored.

Bit 0 – INCOMPRX Incomplete Packet Status bit

Value	Description
0	Written by then software to clear this bit
1	The packet in the RX FIFO during a high-bandwidth Isochronous/Interrupt transfer is incomplete because parts of the data were not received

37.11.5. Endpoint1-7 TX Type Registers

Name: TXTYPE
Offset: 0x101A
Reset: 0x0000
Property: PAC Write-Protection

Table 37-82. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
	SPEED[1:0]		PROTOCOL[1:0]		ENDPOINTNUMBER[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

Bits 7:6 – SPEED[1:0] Operating Speed Control bits.

Value	Description
11	Low-Speed
10	Full-Speed
01	Hi-Speed
00	Reserved

Bits 5:4 – PROTOCOL[1:0] TX Endpoint Protocol Control bits.

Value	Description
11	Interrupt
10	Bulk
01	Isochronous
00	Control

Bits 3:0 – ENDPOINTNUMBER[3:0] TX Target Endpoint Number bits
This value is the endpoint number contained in the TX endpoint descriptor returned to the USB module during device enumeration.

37.11.6. Host Endpoint 1-7 Polling Interval Register

Name: TXINTERVAL
Offset: 0x101B
Reset: 0x0000
Property: PAC Write-Protection

Table 37-83. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
	TXPOLLINGINTERVAL[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 – TXPOLLINGINTERVAL[7:0] Endpoint TX Polling Interval/NAK Limit bits (Host mode)
For Interrupt and Isochronous transfers, this field defines the polling interval for the endpoint. For Bulk end- points,this field sets the number of frames/microframes after which the endpoint should time out on receiving a stream of NAK responses.
The following table describes the valid values and interpretation for these bits:

Transfer Type	Speed	Valid Values (m)	Interpretation
Interrupt	Low/Full	0x01 to 0xFF	Polling interval is 'm' frames.
	High	0x01 to 0x10	Polling interval is $2^{(m-1)}$ frames.
Isochronous	Full or High	0x01 to 0x10	Polling interval is $2^{(m-1)}$ frames/microframes.
Bulk	Full or High	0x02 to 0x10	NAK limit is $2^{(m-1)}$ frames/microframes. A value of '0' or '1' disables the NAK time-out function.

37.11.7. Endpoint1-7 RX Type Registers

Name: RXTYPE
Offset: 0x101C
Reset: 0x0000
Property: PAC Write-Protection

Table 37-84. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
	SPEED[1:0]		PROTOCOL[1:0]		ENDPOINTNUMBER[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

Bits 7:6 – SPEED[1:0] Operating Speed Control bits.

Value	Description
11	Low-Speed
10	Full-Speed
01	Hi-Speed
00	Reserved

Bits 5:4 – PROTOCOL[1:0] RX Endpoint Protocol Control bits.

Value	Description
11	Interrupt
10	Bulk
01	Isochronous
00	Control

Bits 3:0 – ENDPOINTNUMBER[3:0] RX Target Endpoint Number bits

This value is the endpoint number contained in the RX endpoint descriptor returned to the USB module during device enumeration.

37.11.8. Host Endpoint 1-7 Polling Interval Register

Name: RXINTERVAL
Offset: 0x101D
Reset: 0x0000
Property: PAC Write-Protection

Table 37-85. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
	RXPOLLINGINTERVAL[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 – RXPOLLINGINTERVAL[7:0] Endpoint RX Polling Interval/NAK Limit bits
For Interrupt and Isochronous transfers, this field defines the polling interval for the endpoint. For Bulk end- points,this field sets the number of frames/microframes after which the endpoint should time out on receiving a stream of NAK responses.
The following table describes the valid values and interpretation for these bits:

Transfer Type	Speed	Valid Values (m)	Interpretation
Interrupt	Low/Full	0x01 to 0xFF	Polling interval is 'm' frames.
	High	0x01 to 0x10	Polling interval is $2^{(m-1)}$ frames.
Isochronous	Full or High	0x01 to 0x10	Polling interval is $2^{(m-1)}$ frames/microframes.
Bulk	Full or High	0x02 to 0x10	NAK limit is $2^{(m-1)}$ frames/microframes. A value of '0' or '1' disables the NAK time-out function.

37.11.9. USB Endpoint ‘x’ Request Packet Count Register (Host Mode Only)

Name: USBExRPC
Offset: 0x1300 + (x-1)*0x04 [x=1..7]
Reset: 0x0000
Property: Read/Write

Table 37-86. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
	RQPKTCNT[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
	RQPKTCNT[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 – RQPKTCNT[15:0] Request Packet Count Bits
Sets the number of packets of size MAXP that are to be transferred in a block transfer. This register is only available in *Host mode* when AUTOREQ is set.

37.12. Register Summary: Device Mode Only Common Registers

For descriptions and definitions of both Register and bitfield properties, refer to Register Properties.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00 ... 0x0FFF	Reserved									
0x1000	FUNCADDR	7:0		FUNCADDR[6:0]						
0x1001 ... 0x14FF	Reserved									
0x1500	PHY00	31:24								
		23:16								
		15:8								
		7:0	RXPHSSEL[2:0]			SLEWRATE[1:0]		PREEMP[2:0]		

37.12.1. Device Function Address Register

Name: FUNCADDR
Offset: 0x1000
Reset: 0x0000
Property: Read/Write

Table 37-87. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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

Bits 6:0 – FUNCADDR[6:0] Device Functional Address Bits
These bits are only available in *Device mode*. This field is written with the address received through a SET_ADDRESS command, which will then be used for decoding the function address in subsequent token packets.

37.12.2. USB PHY00 Control Register

Name: PHY00
Offset: 0x1500
Reset: 0x0000000000
Property: PAC Write-Protection

Table 37-88. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	RXPHSSEL[2:0]			SLEWRATE[1:0]		PREEMP[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 7:5 – RXPHSSEL[2:0] RX Clock Phase Select
Manually set the Rx Clock phase select. these bits will tune the HS RX path sample timing between digital and analog inside PHY. The delay associated with each step is 256ps.
0x0 = represents the earliest phase
0x07 = represents the latest phase

Bits 4:3 – SLEWRATE[1:0] Adjust FS/LS Slew Rate
These bits will increase/decrease the FS/LS rising/falling time. This tuning can be done when long cable or large capacitance is introduced on DP/DM. It is not applicable for HS signal quality.
0x0: highest slew rate
0x1: middle slew rate
0x2: middle slew rate
0x3 : smallest slew rate

Value	Description
0	IDDIG value from PHY is the source of ID
1	IDVAL is the source of ID

Bits 2:0 – PREEMP[2:0] Enables Pre-emphasis
Enables pre-emphasis under certain circumstances in order to compensate for excessive capacitive loading on D+ / D-. This increases the slew rate.

0x1= enable pre-emphasis during SOF and EOP
0x2=enable pre-emphasis during chirp
0x0 =enable pre-emphasis in non-chirp state
0x3 = always enable pre-emphasis
0x4-0x7 = reserved

37.13. Register Summary: Device Mode Endpoint0 Registers

For descriptions and definitions of both Register and bitfield properties, refer to Register Properties.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00 ... 0x1011	Reserved									
0x1012	CSR0L	7:0	SERVICEDSETUPEND	SERVICEDRXPKTRDY	SENDSTALL	SETUPEND	DATAEND	SENTSTALL	TXPKTRDY	RXPKTRDY
0x1013	CSR0H	7:0								FLSHFIFO

37.13.1. USB Control Status Register Low for Endpoint0

Name: CSR0L
Offset: 0x1012
Reset: 0x0000
Property: PAC Write-Protection

Table 37-89. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
	SERVICEDSETUPEND	SERVICEDRXPKTRDY	SENDSTALL	SETUPEND	DATAEND	SENTSTALL	TXPKTRDY	RXPKTRDY
Access	R/W/HC	R/W/HC	R/W/HC	R/W/HS	R/W/HS	R/HS	R	R
Reset	0	0	0	0	0	0	0	0

Bit 7 – SERVICEDSETUPEND Clear SETUPEND Control bit

Value	Description
0	Do not clear
1	Clear the SETUPEND bit in this register. This bit is automatically cleared.

Bit 6 – SERVICEDRXPKTRDY Clear Control Bit

Value	Description
0	Do not clear
1	Clear the RXPKTRDY bit in this register. This bit is automatically cleared.

Bit 5 – SENDSTALL Send Stall Control Bit

Value	Description
0	Do not send STALL handshake.
1	Terminate the current transaction and transmit a STALL handshake. This bit is automatically cleared.

Bit 4 – SETUPEND Early Control Transaction End Status bit
This bit is cleared by writing a '1' to the SERVICEDSETUPEND bit in this register.

Value	Description
0	Normal operation
1	A control transaction ended before the DATAEND bit has been set. An interrupt will be generated and the FIFO flushed at this time.

Bit 3 – DATAEND End of Data Control bit
The software sets this bit when:

- Setting TXPKTRDY for the last data packet
- Clearing RXPKTRDY after unloading the last data packet
- Setting TXPKTRDY for a zero length data packet

Hardware clears this bit.

Bit 2 – SENTSTALL STALL Sent Status bit

Value	Description
0	Software clear of bit
1	STALL handshake has been transmitted

Bit 1 – TXPKTRDY TX Packet Ready Control bit

Value	Description
0	No data packet is ready for transmit
1	Data packet has been loaded into the FIFO. It is cleared automatically.

Bit 0 – RXPKTRDY RX Packet Ready Status bit
This bit is cleared by setting the SERVICEDRXPKTRDY bit.

Value	Description
0	No data packet has been received
1	Data packet has been received. Interrupt is generated (when enabled) when this bit is set.

37.13.2. USB Control Status Register High for Endpoint0

Name: CSR0H
Offset: 0x1013
Reset: 0x0000
Property: PAC Write-Protection

Table 37-90. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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

Bit 0 – FLSHFIFO Flush FIFO Control bit

Value	Description
0	No Flush operation
1	Flush the next packet to be transmitted/read from the Endpoint 0 FIFO. The FIFO pointer is reset and the TXPKTRDY/RXPKTRDY bit is cleared. Automatically cleared when the operation completes. Should only be used when TXPKTRDY/RXPKTRDY = 1.

37.14. Register Summary: Device Mode Endpoint1-7 Registers

For descriptions and definitions of both Register and bitfield properties, refer to Register Properties.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00 ... 0x1011	Reserved									
0x1012	TXCSRL	7:0	INCOMPRX	CLRDT	SENTSTALL	SENDSTALL	FLUSHFIFO	UNDERRUN	FIFONE	TXPKTRDY
0x1013	TXCSRH	7:0	AUTOSET	ISO	MODE	DMAREQEN	FRCDATATOG	DMAREQMODE		
0x1014 ... 0x1015	Reserved									
0x1016	RXCSRL	7:0	CLRDATATOG	SENTSTALL	SENDSTALL	FLUSHFIFO	DATAERROR	OVERRUN	FIFOFULL	RXPKTRDY
0x1017	RXCSRH	7:0	AUTOCLEAR	ISO	DMAREQENAB	DISNYET	DMAREQMODE			INCOMPRX
0x1018 ... 0x14FF	Reserved									
0x1500	PHY00	31:24								
		23:16								
		15:8								
		7:0	RXPHSSEL[2:0]			SLEWRATE[1:0]		PREEMP[2:0]		
0x1504	PHY04	31:24								
		23:16								
		15:8								
		7:0	SQUELCH[2:0]			HIZ	Reserved	TXPHSSEL[2:0]		
0x1508	PHY08	31:24								
		23:16								
		15:8								
		7:0								SQUELCH
0x150C	PHY0C	31:24								
		23:16								
		15:8								
		7:0	TUNE[2:0]							
0x1510	PHY10	31:24								
		23:16								
		15:8								
		7:0	DRVTUNE[2:0]			TUNE[4:0]				
0x1514	PHY14	31:24								
		23:16								
		15:8								
		7:0	ODT			BYPSSSQUELCH		COMPBYPSS[2:0]		
0x1518	PHY18	31:24								
		23:16								
		15:8								
		7:0							ODT[1:0]	
0x151C	PHY1C	31:24								
		23:16								
		15:8								
		7:0	FSLSDIFF						ODTBYPASS	
0x1520	PHY20	31:24								
		23:16								
		15:8								
		7:0	HSSLEW[1:0]							
0x1524	PHY24	31:24								
		23:16								
		15:8								
		7:0	HSDRIVST[1:0]		HSPREEMPST[2:0]			PREEMPHEN	OTGPDN	HSSLEW

Register Summary: Device Mode Endpoint1-7 Registers (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x1528	PHY28	31:24								
		23:16								
		15:8								
		7:0	HSDRVCOMP[2:0]			DISCONDET[3:0]				HSDRIVST
0x152C ... 0x1543	Reserved									
0x1544	PHY44	31:24								
		23:16								
		15:8								
		7:0	FRCSESSEND				FRCVBUSVAL	DIGDBG	PLLDAMP	
0x1548	PHY48	31:24								
		23:16								
		15:8								
		7:0	SESSENDTUNE[2:0]					VBUSCHRGE	FRCBSESSVAL	FRCASESSVAL
0x154C	PHY4C	31:24								
		23:16								
		15:8								
		7:0	BSESSVALIDTUNE[1:0]					VBUSVALTUNE[2:0]		
0x1550	PHY50	31:24								
		23:16								
		15:8								
		7:0	COMPCURREF[2:0]				ASESSVALIDTUNE[2:0]			BSESSVALIDTUNE

37.14.1. TX Control Status Register Low for Endpoint 1-7

Name: TXCSRL
Offset: 0x1012
Reset: 0x0000
Property: PAC Write-Protection

Table 37-91. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
	INCOMPRX	CLRDT	SENTSTALL	SENDSTALL	FLUSHFIFO	UNDERRUN	FIFONE	TXPKTRDY
Access	R/W/HS	R/W	R/W	R/W	R/W/HC	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 7 – INCOMPRX Incomplete Packet Status bit
In anything other than Isochronous transfer, this bit will always return 0.

Value	Description
0	Written by the software to clear this bit
1	The packet in the RX FIFO during a high-bandwidth Isochronous/Interrupt transfer is incomplete because parts of the data were not received

Bit 6 – CLRDT Clear Data Toggle Control Bit

Value	Description
0	Do not clear the data toggle
1	Resets the endpoint data toggle to 0

Bit 5 – SENTSTALL Stall Handshake Status Bit

Value	Description
0	Written by the software to clear this bit
1	STALL handshake is transmitted

Bit 4 – SENDSTALL STALL Handshake Control bit

Value	Description
0	Terminate stall condition
1	Issue a STALL handshake

Bit 3 – FLUSHFIFO FIFO Flush Control bit

Value	Description
0	Do not flush the FIFO
1	Flush the latest packet from the endpoint TX FIFO. The FIFO pointer is reset, the TXPKTRDY bit is cleared and an interrupt is generated.

Bit 2 – UNDERRUN Underrun Status bit

Value	Description
0	Written by software to clear this bit.
1	An IN token has been received when the TXPKTRDY bit is not set.

Bit 1 – FIFONE FIFO Not Empty Status bit

Value	Description
0	TX FIFO is empty
1	There is at least 1 packet in the TX FIFO

Bit 0 – TXPKTRDY TX Packet Ready Control bit

The software sets this bit after loading a data packet into the FIFO. It is cleared automatically when a data packet has been transmitted. This bit is also automatically cleared prior to loading a second packet into a double-buffered FIFO.

37.14.2. TX Control Status Register High for Endpoint 1-7

Name: TXCSRH
Offset: 0x1013
Reset: 0x0000
Property: PAC Write-Protection

Table 37-92. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
	AUTOSET	ISO	MODE	DMAREQEN	FRCDATATOG	DMAREQMODE		
Access	R/W	R/W	R/W	R/W	R/W	R/W		
Reset	0	0	0	0	0	0		

Bit 7 – AUTOSET Auto Set Control bit

Value	Description
0	TXPKTRDY must be set manually for all packet sizes
1	TXPKTRDY will be automatically set when data of the maximum packet size (value in TXMAXP) is loaded into the TX FIFO. If a packet of less than the maximum packet size is loaded, then TXPKTRDY will have to be set manually.

Bit 6 – ISO Isochronous TX Endpoint Enable bit

This bit only has an effect in Device mode. In Host mode, it always returns zero.

Value	Description
0	Disables the endpoint for Isochronous transfers and enables it for Bulk or Interrupt transfers.
1	Enables the endpoint for Isochronous transfers

Bit 5 – MODE Endpoint Direction Control bit

This bit only has any effect where the same endpoint FIFO is used for both TX and RX transactions.

Value	Description
0	Endpoint is RX
1	Endpoint is TX

Bit 4 – DMAREQEN Endpoint DMA Request Enable bit

Value	Description
0	DMA requests are disabled for this endpoint
1	DMA requests are enabled for this endpoint

Bit 3 – FRCDATATOG Force Endpoint Data Toggle Control bit

Value	Description
0	No forced behavior
1	Forces the endpoint data toggle to switch and the data packet to be cleared from the FIFO, regardless of whether an ACK was received.

Bit 2 – DMAREQMODE Endpoint DMA Request Mode Control bit

This bit must not be cleared either before or in the same cycle as the DMAREQEN bit is cleared.

Value	Description
0	DMA Request Mode0
1	DMA Request Mode1

37.14.3. RX Control Status Register Low for Endpoint 1-7

Name: RXCSRL
Offset: 0x1016
Reset: 0x0000
Property: PAC Write-Protection

Table 37-93. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
	CLRDATATOG	SENTSTALL	SENDSTALL	FLUSHFIFO	DATAERROR	OVERRUN	FIFOFULL	RXPKTRDY
Access	R/W/HC	R/W/HS	R/W	R/W/HC	R/W/HC	R/W/HS	R/W/HC	R/W/HS
Reset	0	0	0	0	0	0	0	0

Bit 7 – CLRDATATOG Clear Data Toggle Control Bit

Value	Description
0	Do not clear the data toggle
1	Resets the endpoint data toggle to 0

Bit 6 – SENTSTALL Stall Handshake Status Bit

Value	Description
0	Written by the software to clear this bit
1	STALL handshake is transmitted

Bit 5 – SENDSTALL STALL Handshake Control bit (Device Mode)

Value	Description
0	Terminate stall condition
1	Issue a STALL handshake

Bit 4 – FLUSHFIFO FIFO Flush Control bit
This bit is automatically cleared.

Value	Description
0	Normal FIFO operation
1	Flush the next packet to be read from the endpoint RX FIFO. The FIFO pointer is reset and the RXPKTRDY bit is cleared. This should only be used when RXPKTRDY is set. If the FIFO is double- buffered, FLUSH may need to be set twice to completely clear the FIFO.

Bit 3 – DATAERROR Data Packet Error Status bit
This bit is cleared when RXPKTRDY is cleared. This bit is only valid when the endpoint is operating in ISO mode. In Bulk mode, it always returns zero.

Value	Description
0	No data error
1	The data packet has a CRC or bit-stuff error.

Bit 2 – OVERRUN Data Overrun Status bit
This bit is only valid when the endpoint is operating in ISO mode. In Bulk mode, it always returns zero.

Value	Description
0	Written by software to clear this bit
1	An OUT packet cannot be loaded into the RX FIFO.

Bit 1 – FIFOFULL FIFO Full Status bit

Value	Description
0	The RX FIFO has at least one free space
1	No more packets can be loaded into the RX FIFO

Bit 0 – RXPKTRDY Data Packet Reception Status bit

Value	Description
0	Written by software to clear this bit when the packet has been unloaded from the RX FIFO.
1	A data packet has been received. An interrupt is generated.

37.14.4. RX Control Status Register High for Endpoint 1-7

Name: RXCSRH
Offset: 0x1017
Reset: 0x0000
Property: PAC Write-Protection

Table 37-94. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
	AUTOCLEAR	ISO	DMAREQENAB	DISNYET	DMAREQMODE			INCOMPRX
Access	R/W	R/W	R/W	R/W	R/W			R/W
Reset	0	0	0	0	0			0

Bit 7 – AUTOCLEAR Automatic Clear Control bit
This bit should not be set for high-bandwidth Isochronous endpoints.

Value	Description
0	No automatic clearing of RXPKTRDY
1	RXPKTRDY will be automatically cleared when a packet of RXMAXP bytes has been unloaded from the RXFIFO. When packets of less than the maximum packet size are unloaded, RXPKTRDY will have to be cleared manually. When using a DMA to unload the RX FIFO, data is read from the RX FIFO in 4-byte chunks regardless of the RXMAXP.

Bit 6 – ISO Isochronous Endpoint Control bit (Device Mode)
This bit only has an effect in Device mode. In Host mode, it always returns zero.

Value	Description
0	Enable the RX endpoint for Bulk/Interrupt transfers
1	Enable the RX endpoint for Isochronous transfers

Bit 5 – DMAREQENAB DMA Request Enable Control bit

Value	Description
0	Disable DMA requests for the RX endpoint.
1	Enable DMA requests for the RX endpoint.

Bit 4 – DISNYET Disable NYET Handshakes Control/PID Error Status bit (Device mode)
In Bulk/Interrupt transactions, this bit only has any effect in Hi-Speed mode, in which mode it should be set for all Interrupt endpoints.

Value	Description
0	Normal operation.
1	In Bulk/Interrupt transactions, disables the sending of NYET handshakes. All successfully received RX packets are ACKed including at the point at which the FIFO becomes full.

Bit 3 – DMAREQMODE DMA Request Mode Selection bit

Value	Description
0	DMA Request Mode 0
1	DMA Request Mode 1

Bit 0 – INCOMPRX Incomplete Packet Status bit

Value	Description
0	Written by the software to clear this bit
1	The packet in the RX FIFO during a high-bandwidth Isochronous/Interrupt transfer is incomplete because parts of the data were not received

37.14.5. PHY Control Register 00

Name: PHY00
Offset: 0x1500
Reset: 0x00000019
Property: PAC Write-Protection

Note: The USB PHY values must be loaded from the CAL OTP area into the USB PHY registers by software, before enabling the USB, to achieve the specified accuracy.

Table 37-95. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

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

Bits 7:5 – RXPHSSEL[2:0] RX Clock Phase Select
The delay associated with each step is 256ps.

Value	Description
111	Represents the latest phase (7 * 256ps)
110	-
100	-
011	-
010	-
001	-
000	Represents the earliest phase (0 * 256ps)

Bits 4:3 – SLEWRATE[1:0] Adjustment for FS/LS Slew Rate

Value	Description
11	Slowest Slew Rate
10	-
01	-
00	Fastest Slew Rate

Bits 2:0 – PREEMP[2:0] Pre-Emphasis Setting

Value	Description
111	Enable pre-emphasis always
110	Enable pre-emphasis during chirp and non-chirp
100	Enable pre-emphasis during non-chirp
011	Enable pre-emphasis during SOF and EOP and chirp
010	Enable pre-emphasis during chirp
001	Enable pre-emphasis during SOF and EOP
000	Disable pre-emphasis

37.14.6. PHY Control Register 04

Name: PHY04
Offset: 0x1504
Reset: 0x0000008F
Property: PAC Write-Protection

Note: The USB PHY values must be loaded from the CAL OTP area into the USB PHY registers by software, before enabling the USB, to achieve the specified accuracy.

Table 37-96. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	SQUELCH[2:0]			HIZ	Reserved	TXPHSSEL[2:0]		
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	0	0	0	1	1	1	1

Bits 7:5 – SQUELCH[2:0] Squelch Trigger Point Configuration
Sets the lower 3 bits of the RX squelch trigger point configuration.
Settings include lower bits (PHY04.5:7) and upper bit the upper bit (PHY08.0).

Value	Description
1111	200 mV
1110	125 mV
1101	187.5 mV
1100	150 mV (default)
1011	175 mV
1010	100 mV
1001	162.5 mV
1000	Reserved
0111	Reserved
0110	75 mV
0101	137 mV

Bit 4 – HIZ

Sets D+/D- to a high impedance state.

Value	Description
1	Enabled
0	Disabled

Bit 3 – Reserved

Bits 2:0 – TXPHSSEL[2:0] TX Clock Phase Select

Value	Description
111	Represents the latest phase (7 * 256ps)
110	-
100	-
011	-
010	-
001	-
000	Represents the earliest phase (0 * 256ps)

37.14.7. PHY Control Register 08

Name: PHY08
Offset: 0x1508
Reset: 0x00000007
Property: PAC Write-Protection

Note: The USB PHY values must be loaded from the CAL OTP area into the USB PHY registers by software, before enabling the USB, to achieve the specified accuracy.

Table 37-97. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

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

Bit 0 – SQUELCH RX Squelch Trigger Point Set
Sets the upper bit of the RX squelch trigger point configuration.
Settings include lower bits (PHY04.5:7) and upper bit the upper bit (PHY08.0).

Value	Description
1111	200 mV
1110	125 mV
1101	187.5 mV
1100	150 mV (default)
1011	175 mV
1010	100 mV
1001	162.5 mV
1000	Reserved
0111	Reserved
0110	75 mV
0101	137 mV
0100	100 mV
0011	162.5 mV

Value	Description
0010	87.5 mV
0001	Reserved
0000	112.5 mV

37.14.8. PHY Control Register 0C

Name: PHY0C
Offset: 0x150C
Reset: 0x000000E0
Property: PAC Write-Protection

Note: The USB PHY values must be loaded from the CAL OTP area into the USB PHY registers by software, before enabling the USB, to achieve the specified accuracy.

Table 37-98. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

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

Bits 7:5 – TUNE[2:0] Amplitude Tuning

Sets the lower 3 bits of the HS amplitude tuning.
Settings include the lower bits (PHY0C.5:7) and upper bit the upper bit (PHY10.0:4) – setting of each bit location lowers the amplitude by the same amount regardless of location.

Value	Description
`11111111`	Setting with the smallest amplitude
`10101100`	4-'0' and 4-'1' is the middle amplitude
`00000000`	Setting with the largest amplitude

37.14.9. PHY Control Register 10

Name: PHY10
Offset: 0x1510
Reset: 0x000000AA
Property: PAC Write-Protection

Note: The USB PHY values must be loaded from the CAL OTP area into the USB PHY registers by software, before enabling the USB, to achieve the specified accuracy.

Table 37-99. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

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

Bits 7:5 – DRVTUNE[2:0] Driver Strength Tuning
Sets the lower 3 bits for the HS/FS/LS driver strength tuning.
Settings include the lower bits (PHY10.5:7) and the upper bit (PHY14.0:1).

Value	Description
11111	Fastest rise fall time
00000	Slowest rise fall time

Bits 4:0 – TUNE[4:0] Amplitude Tuning
Sets the upper 5 bits of the HS amplitude tuning.
Settings include the lower bits (PHY0C.5:7) and the upper bit the upper bit (PHY10.0:4) – setting of each bit location lowers the amplitude by the same amount regardless of location.

Value	Description
11111111	Setting with the smallest amplitude
10101100	4-‘0’ and 4-‘1’ is the middle amplitude
00000000	Setting with the largest amplitude

37.14.10. PHY Control Register 14

Name: PHY14
Offset: 0x1514
Reset: 0x00000012
Property: PAC Write-Protection

Note: The USB PHY values must be loaded from the CAL OTP area into the USB PHY registers by software, before enabling the USB, to achieve the specified accuracy.

Table 37-100. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

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

Bit 7 – ODT On Die Termination
Sets the lowest bit of the on die termination compensation voltage reference.
Settings include the lower bits (PHY14.8) and the upper bit the upper bits (PHY18.0:1).

Value	Description
111	362.5 mV
110	375 mV
101	387.5 mV
100	450 mV
011	437.5 mV
010	425 mV
001	412.5 mV
000	400 mV

Bit 4 – BYPSSSQUELCH Bypass Squelch Trigger Point
Sets the bypass squelch trigger point configure in chirp mode.

Value	Description
1	Bypass
0	Do not bypass

Bits 2:0 – COMPBYPSS[2:0] Auto-Compression Bypass
Sets the auto-compression bypass.
Settings include the lower bits (PHY0C.5:7) and the upper bit the upper bit (PHY10.0:4) – setting of each bit location lowers the amplitude by the same amount regardless of location.

Value	Description
11	Disable current and disable ODT auto-calibration
10	Disable current and enable ODT auto-calibration
01	Enable current and disable ODT auto-calibration
00	Enable current and ODT auto-calibration

Bits 1:0 – DRVTUNE[1:0] HS/FS/LS Driver Strength Tuning
Sets the upper 2 bits for HS/FS/LS driver strength tuning.
Settings include the lower bits (PHY10.5:7) and the upper bit (PHY14.0:1).

Value	Description
11111	Fastest rise fall time
00000	Slowest rise fall time

37.14.11. PHY Control Register 18

Name: PHY18
Offset: 0x1518
Reset: 0x00000008
Property: PAC Write-Protection

Note: The USB PHY values must be loaded from the CAL OTP area into the USB PHY registers by software, before enabling the USB, to achieve the specified accuracy.

Table 37-101. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

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

Bits 1:0 – ODT[1:0] On Die Termination
Sets the lowest bit of the on die termination compensation voltage reference.
Settings include the lower bits (PHY14.7) and the upper bits (PHY18.0:1).

Value	Description
`111`	362.5 mV
`110`	375 mV
`101`	387.5 mV
`100`	450 mV
`011`	437.5 mV
`010`	425 mV
`001`	412.5 mV
`000`	400 mV

37.14.12. PHY Control Register 1C

Name: PHY1C
Offset: 0x151C
Reset: 0x00000082
Property: PAC Write-Protection

Note: The USB PHY values must be loaded from the CAL OTP area into the USB PHY registers by software, before enabling the USB, to achieve the specified accuracy.

Table 37-102. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

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

Bit 7 – FSLSDIFF FS/LS Differential Receiver
Turns off FS/LS differential receiver in suspend mode.

Value	Description
1	On
0	Off

Bit 1 – ODTBYPASS ODT Auto-Refresh Bypass
Sets the ODT auto-refresh bypass.

Value	Description
1	Bypass
0	Do not bypass

37.14.13. PHY Control Register 20

Name: PHY20
Offset: 0x1520
Reset: 0x00000080
Property: PAC Write-Protection

Note: The USB PHY values must be loaded from the CAL OTP area into the USB PHY registers by software, before enabling the USB, to achieve the specified accuracy.

Table 37-103. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

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

Bits 7:6 – HSSLEW[1:0] HS Slew Rate
Sets the HS slew rate.
Settings include the lower bits (PHY20.6:7) and the upper bits (PHY24.0).

Value	Description
111	Fastest rise/fall time
010	Middle slew rate
001	Slowest rise/fall time
000	Reserved

37.14.14. PHY Control Register 24

Name: PHY24
Offset: 0x1524
Reset: 0x0000000C
Property: PAC Write-Protection

Note: The USB PHY values must be loaded from the CAL OTP area into the USB PHY registers by software, before enabling the USB, to achieve the specified accuracy.

Table 37-104. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	HSDRIVST[1:0]		HSPREEMPST[2:0]			PREEMPHEN	OTGPDN	HSSLEW
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	1	1	0	0

Bits 7:6 – HSDRIVST[1:0] HS Transmit Driver Strength
Sets the HS transmit driver strength.
Settings include the lower bits (PHY24.6:7) and the upper bit (PHY28.0).

Value	Description
`111`	Strongest drive strength
`000`	Weakest drive strength

Bits 5:3 – HSPREEMPST[2:0] HS Transmit Pre-Emphasis Strength
Sets the HS transmit pre-emphasis strength.

Value	Description
`11`	Slowest Slew Rate
`10`	-
`01`	-
`00`	Fastest Slew Rate

Bit 2 – PREEMPHEN HS Transmit Pre-Emphasis Enable
Enable half-bit pre-emphasis for HS transmit.

Value	Description
1	Enable
0	Disable

Bit 1 – OTGPDN ODT Power Down

Sets the ODT power down.

Value	Description
1	On
0	Off

Bit 0 – HSSLEW HS Slew Rate

Sets the HS slew rate.

Settings include the lower bits (PHY20.6:7) and the upper bit (PHY24.0).

Value	Description
111	Fastest rise/fall time
010	Middle slew rate
001	Slowest rise/fall time
000	Reserved

37.14.15. PHY Control Register 28

Name: PHY28
Offset: 0x1528
Reset: 0x0000001B
Property: PAC Write-Protection

Note: The USB PHY values must be loaded from the CAL OTP area into the USB PHY registers by software, before enabling the USB, to achieve the specified accuracy.

Table 37-105. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

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

Bits 7:5 – HSDRVCOMP[2:0] HS Driver Current Compensation
Sets the HS driver current compensation voltage reference.

Value	Description
111	362.5 mV
110	375 mV
101	387.5 mV
100	450 mV
011	437.5 mV
010	425 mV
001	412.5 mV
000	400 mV

Bits 4:1 – DISCONDET[3:0] HOST Disconnect Detection
Sets the HOST disconnect detection trigger point.

Value	Description
1111	Reserved
1110	612.5 mV

Value	Description
1101	650 mV
1100	Reserved
1011	Reserved
1010	58705 mV
1001	Reserved
1000	600 mV
0111	Reserved
0110	537.5 mV
0101	Reserved
0100	550 mV
0011	625 mV
0010	562.5 mV
0001	600 mV
0000	575 mV

Bit 0 – HSDRIVST HS Transmit Driver Strength
Sets the HS transmit driver strength.
Settings include the lower bits (PHY24.6:7) and the upper bit (PHY28.0).

Value	Description
111	Strongest drive strength
000	Weakest drive strength

37.14.16. PHY Control Register 44

Name: PHY44
Offset: 0x1544
Reset: 0x00000040
Property: PAC Write-Protection

Note: The USB PHY values must be loaded from the CAL OTP area into the USB PHY registers by software, before enabling the USB, to achieve the specified accuracy.

Table 37-106. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	FRCSESSEND				FRCVBUSVAL	DIGDBG	PLLDAMP	
Access	R/W				R/W	R/W	R/W	
Reset	0				0	0	0	

Bit 7 – FRCSESSEND Force Session End

Value	Description
1	-
0	Default

Bit 3 – FRCVBUSVAL Force Output VBUS_VALID

Value	Description
1	-
0	Default

Bit 2 – DIGDBG Digital Debug Interface (Reserved)

Value	Description
1	-
0	Default

Bit 1 – PLLDAMP Digital Debug Interface (Reserved)

Value	Description
1	Decreased PLL damping factor
0	Increased PLL damping factor (default)

37.14.17. PHY Control Register 48

Name: PHY48
Offset: 0x1548
Reset: 0x00000004
Property: PAC Write-Protection

Note: The USB PHY values must be loaded from the CAL OTP area into the USB PHY registers by software, before enabling the USB, to achieve the specified accuracy.

Table 37-107. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	SESSENDTUNE[2:0]					VBUSCHRGE	FRCBSESSVAL	FRCASESSVAL
Access	R/W	R/W	R/W			R/W	R/W	R/W
Reset	0	0	0			1	0	0

Bits 7:5 – SESSENDTUNE[2:0] Session End Reference Tuning

Value	Description
111	300 mV
110	650 mV
101	600 mV
100	550 mV
011	350 mV
010	400 mV
001	450 mV
000	500 mV

Bit 2 – VBUSCHRGE VBUS Charging/Discharging Bypass

Value	Description
1	Default
0	-

Bit 1 – FRCBSESSVAL Force B Session Valid

Value	Description
1	-
0	Default

Bit 0 – FRCASESSVAL Force A Session Valid

Value	Description
1	-
0	Default

37.14.18. PHY Control Register 4C

Name: PHY4C
Offset: 0x154C
Reset: 0x00000000
Property: PAC Write-Protection

Note: The USB PHY values must be loaded from the CAL OTP area into the USB PHY registers by software, before enabling the USB, to achieve the specified accuracy.

Table 37-108. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

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

Bits 7:6 – BSESSVALIDTUNE[1:0] B Session Valid Reference Tune
Settings include the lower bits (PHY4C.6:7) and the upper bit (PHY50.0).

Value	Description
111	2.16 V
110	2.58 V
101	2.52 V
100	2.46 V
011	2.22 V
010	2.28 V
001	2.34 V
000	2.4 V (Default)

Bits 2:0 – VBUSVALTUNE[2:0] VBUS Valid Reference Tune

Value	Description
111	4.3 V
110	4.65 V
101	4.6 V

Value	Description
100	4.55 V
011	4.3 V
010	4.4 V
001	4.5 V
000	4.45 V

37.14.19. PHY Control Register 50

Name: PHY50
Offset: 0x1550
Reset: 0x00000000
Property: PAC Write-Protection

Note: The USB PHY values must be loaded from the CAL OTP area into the USB PHY registers by software, before enabling the USB, to achieve the specified accuracy.

Table 37-109. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

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

Bits 7:5 – COMPCURREF[2:0] Compensation Current Tuning Reference

Value	Description
`111`	162.5 mV
`110`	175 mV
`101`	212.5 mV
`100`	250 mV
`011`	237.5 mV
`010`	225 mV
`001`	187.5 mV
`000`	200 mV

Bits 3:1 – ASESSVALIDTUNE[2:0] A Session Valid Reference Tune

Value	Description
`111`	1.2 V
`110`	1.55 V
`101`	1.5 V

Value	Description
100	1.45 V
011	1.25 V
010	1.3 V
001	1.35 V
000	1.4 V (Default)

Bit 0 – BSESSVALIDTUNE B Session Valid Reference Tune
Settings include the lower bits (PHY4C.6:7) and the upper bit (PHY50.0).

Value	Description
111	2.16 V
110	2.58 V
101	2.52 V
100	2.46 V
011	2.22 V
010	2.28 V
001	2.34 V
000	2.4 V (Default)

38. Full-Speed Universal Serial Bus (USBFS)

38.1. Overview

The Full-Speed Universal Serial Bus interface (USBFS) module complies with the Universal Serial Bus (USB) 2.0 specification supporting both device and embedded host 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 USB host mode supports up to 8 pipes. 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

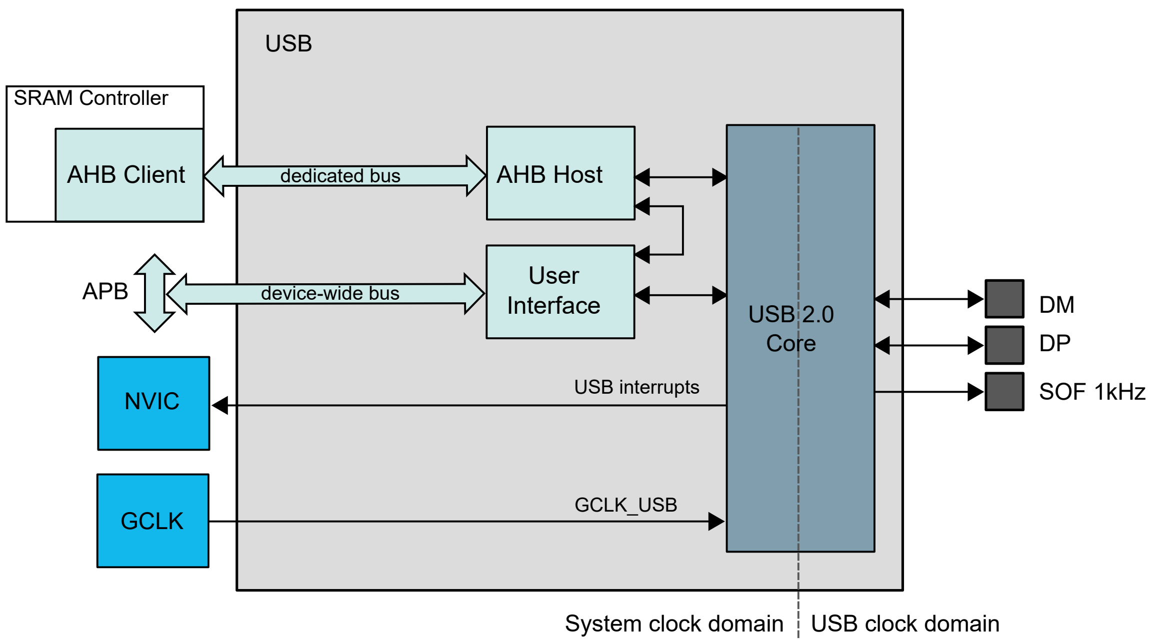
The following are key features of the USB:

- Compatible with the USB 2.0 specification
- USB Embedded Host and Device mode
- Supports full (12 Mbit/s) and low (1.5 Mbit/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
- 1 kHz 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

Refer to the Pinout for additional information on the pin mapping for this peripheral. One signal can be mapped on several pins.

38.5. Peripheral Dependencies

Peripheral Name	Base Address	NVIC IRQ Index: Source	MCLK AHBx/APBx Clock Enable Mask Bit	GCLK Peripheral Channel Clock Name:Register	PAC Peripheral Identifier (PAC.WRCTRL.PERIDx)	Power Domain
USBFS	0x4502_A000	150 : EORSM_DNRSM, EORST_RST, LPM_DCONN, LPMSUSP_DDISC, MSOF, RAMACER, RXSTP_TXSTP[0:7], STALL0_STALL[0:7], STALL1[0:7], SUSPEND, TRFAIL0_TRFAIL[0:7], TRFAIL1_PERR[0:7], UPRSM, WAKEUP 151: SOF_HSOF 1 52 : TRCPT0[0:7] 153 : TRCPT1[0:7]	AHB : MCLK.CLKMSK0[19] APB : MCLK.CLKMSK3[17]	GCLK_USB : GCLK.PCHCTRL[46]	57	VDDREG / VUSB3V3

38.5.1. I/O Lines

The USB pins may be multiplexed with the I/O lines Controller. The user must first configure the I/O Controller to assign the USB pins to their peripheral functions.

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.

38.5.2. Power Management

This peripheral can continue to operate in any Sleep mode where its source clock is running. The interrupts can wake up the device from Sleep modes. Events connected to the event system can trigger other operations in the system without exiting Sleep modes.

38.5.3. Clocks

The USB bus clock (CLK_USB_AHB) can be enabled and disabled in the Main Clock module, MCLK, and the default state of CLK_USB_AHB can be found in the Peripheral Clock Masking.

A generic clock (GCLK_USB) is required to clock the USB. This clock must be configured and enabled in the GCLK before using the USB.

This generic clock is asynchronous to the bus clock (CLK_USB_AHB). Due to this asynchronicity, writes to certain registers will require synchronization between the clock domains.

The USB module requires a GCLK_USB of 48 MHz ± 0.25% clock for low speed and full speed operation. To follow the USB data rate at 12 Mbit/s in full-speed mode, the CLK_USB_AHB clock should be at minimum 8 MHz. The GCLK_USB clock is generated by the DFLL48 using a reference clock. When the USB is disabled, the GCLK used as DFLL reference should be disabled.

38.5.4. DMA

The USB has a built-in Direct Memory Access (DMA) and will read/write data to/from the system RAM when a USB transaction takes place. No CPU or DMA Controller resources are required.

38.5.5. 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.

38.5.6. Register Access Protection

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

- The Device Interrupt Flag (INTFLAG) register
- The Endpoint Interrupt Flag (EPINTFLAG) register
- The Host Interrupt Flag (INTFLAG) register
- The Pipe Interrupt Flag (PINTFLAG) register

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

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

38.5.7. Calibration

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 CAL OTP memory region into the USB Pad Calibration register (PADCAL) by software, after enabling the USB, to achieve the specified accuracy. Refer to the *CAL OTP Configuration* section for additional information.

For additional information on Pad Calibration, refer to the Pad Calibration (PADCAL) register.

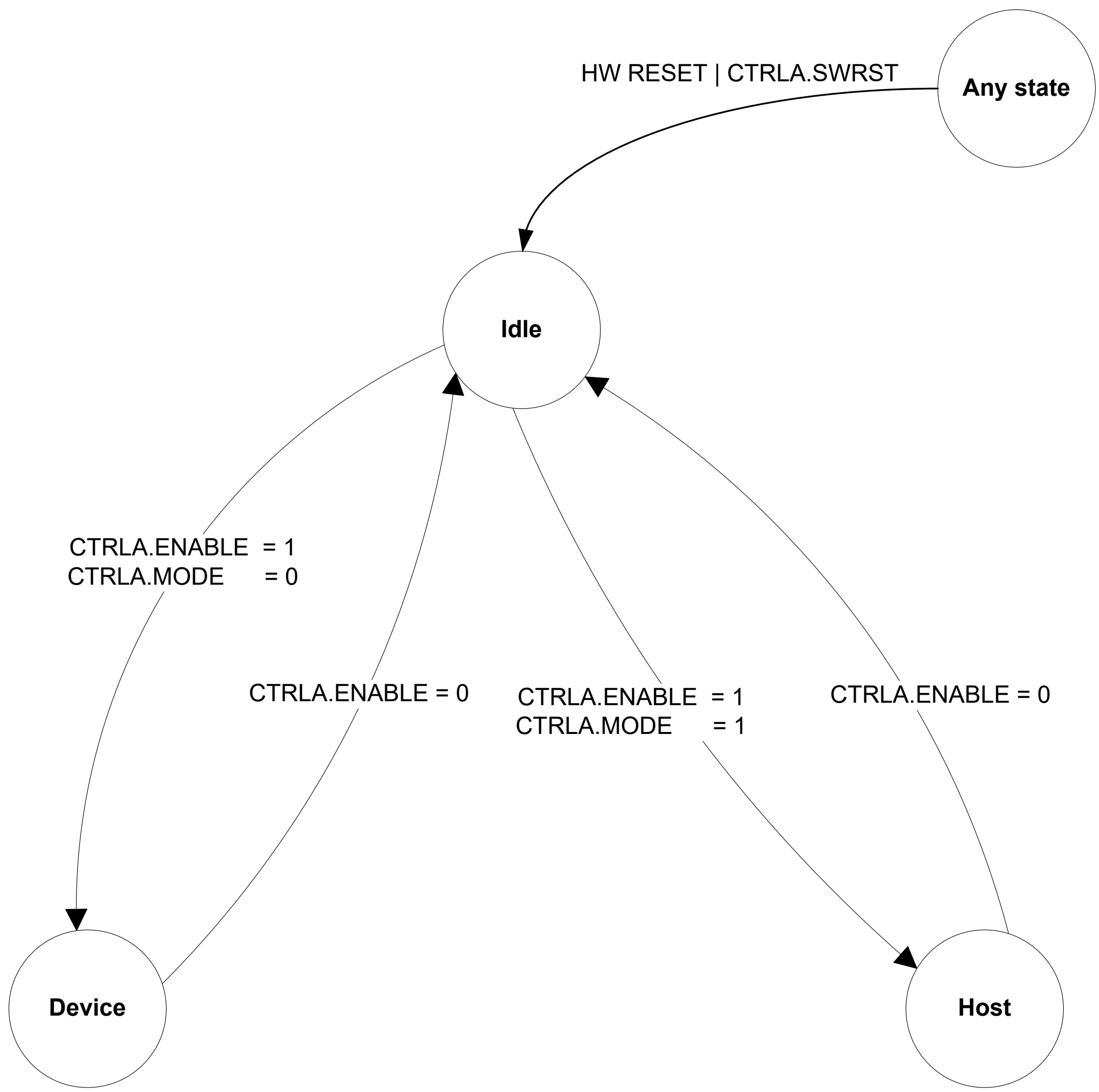
38.6. Functional Description

38.6.1. USB General Operation

38.6.1.1. Initialization

After a hardware reset, the USB is disabled. The user must first enable the USB (CTRLA.ENABLE) in either device mode or host mode (CTRLA.MODE).

Figure 38-2. General States



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 and host are reset.

After enabling the USB, the Pad Calibration register (PADCAL) must be loaded with production calibration values from the CAL OTP memory region.

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 the USB Device Operations for the basic operation of the device mode.

Refer to the Host Operations for the basic operation of the host mode.

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. For additional information, refer to Endpoint Description 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:

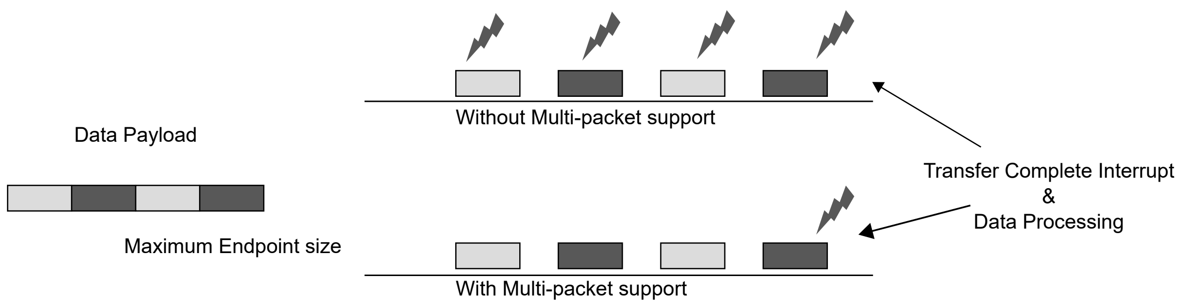
- Device Endpoint Interrupt Enable Clear/Set (EPINTENCLR/SET) register
- Device Endpoint Interrupt Flag (EPINTFLAG) register
- Transmit Stall 0 bit in the Endpoint Status register (EPSTATUS.STALLRQ0)
- Transmit Stall 1 bit in the Endpoint Status register (EPSTATUS.STALLRQ1)

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.

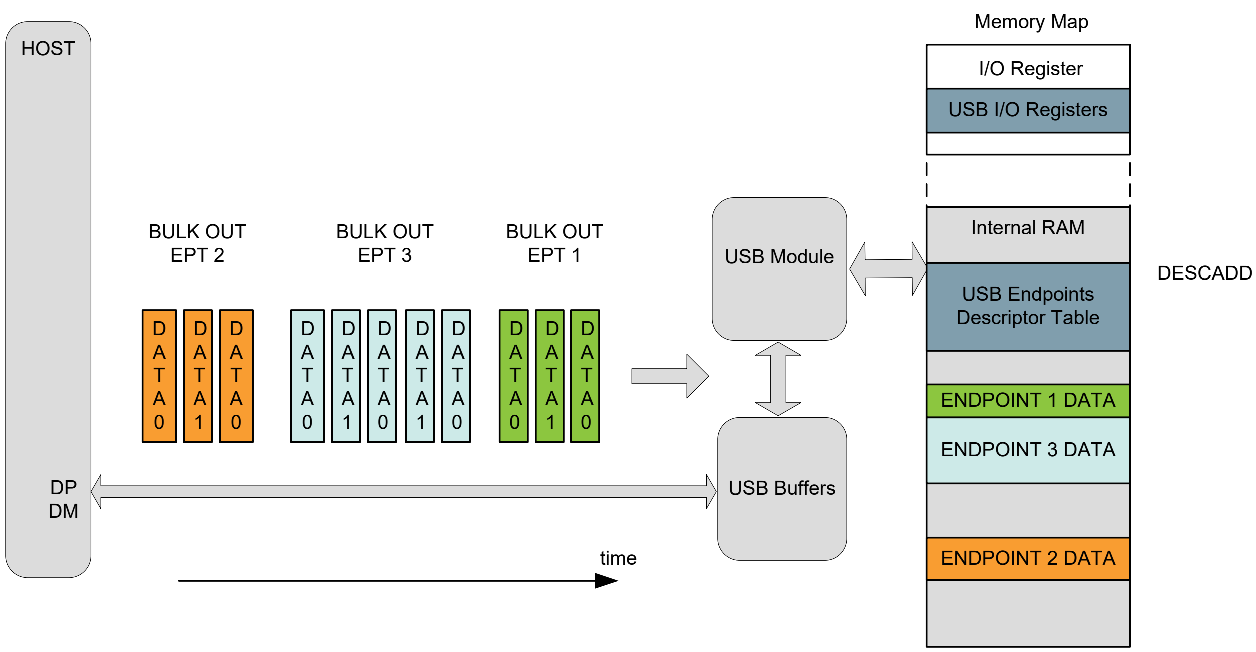
When the data PID matches and if the Received Setup Complete interrupt bit in the Device Endpoint Interrupt Flag register (EPINTFLAG.RXSTP) is equal to zero, ignoring the Bank 0 Ready bit in the Device Endpoint Status register (EPSTATUS.BK0RDY), the incoming data is written to the data buffer pointed 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

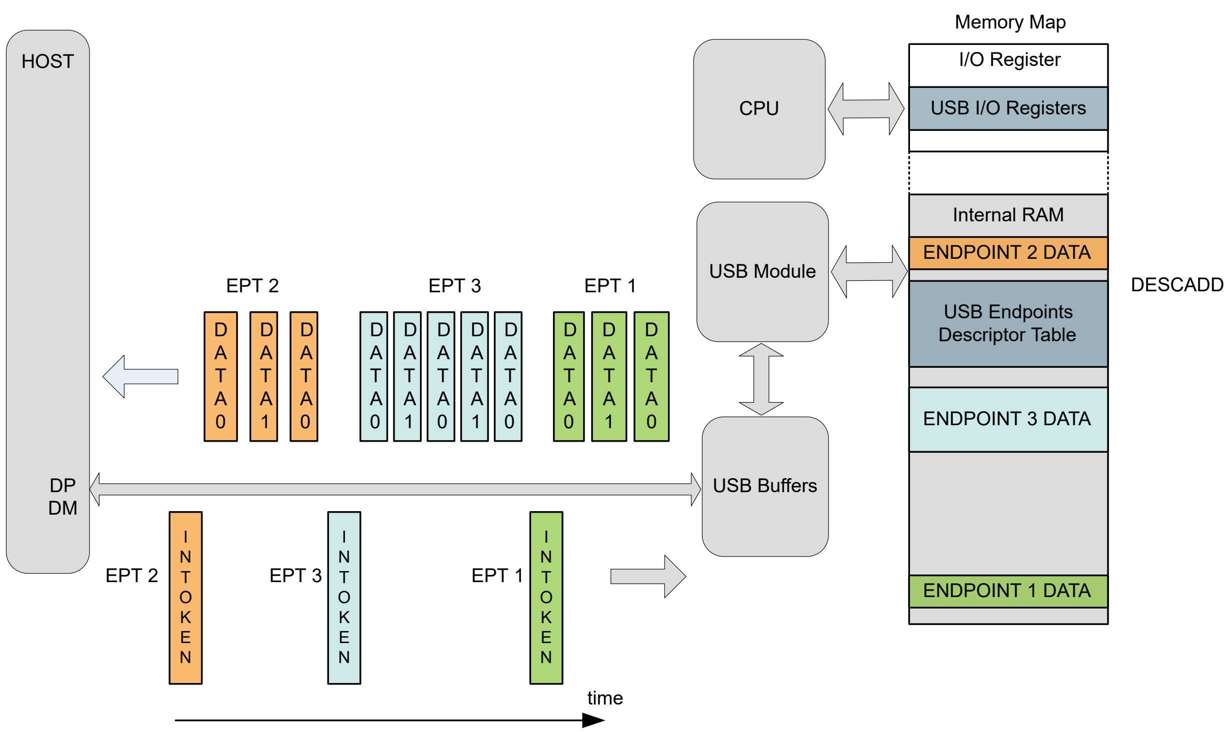
The number of data bytes received is stored in endpoint PCKSIZE.BYTE_COUNT as for normal operation. Since PCKSIZE.BYTE_COUNT is updated after each transaction, it must be set to zero when setting up a new transfer. The total number of bytes to be received must be written to PCKSIZE.MULTI_PACKET_SIZE. 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.

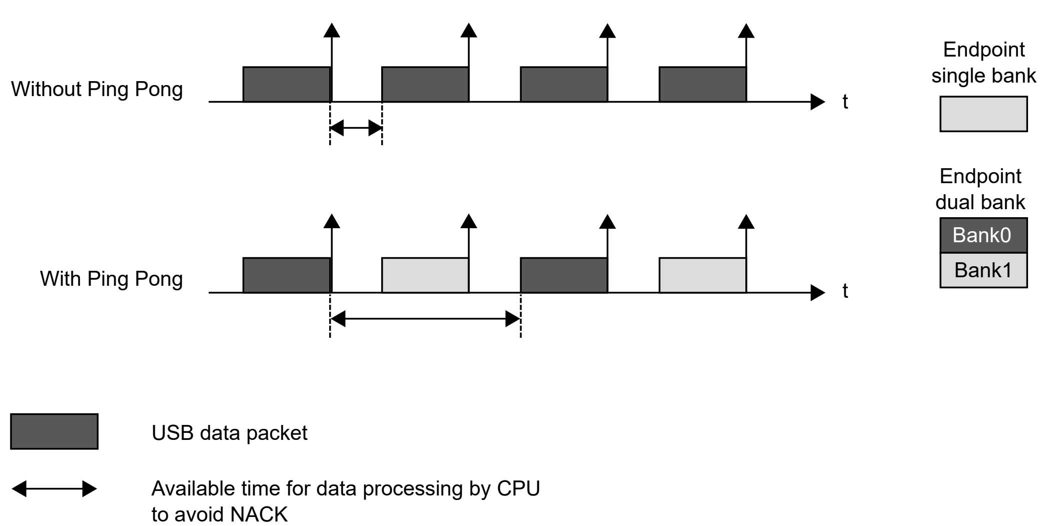
If a maximum payload size packet was sent (i.e. not the last transaction), MULTI_PACKET_SIZE will be incremented by the PCKSIZE.SIZE. If the endpoint is not isochronous the EPSTATUS.DTLGIN bit will be toggled when the transaction has completed. If a short packet was sent (i.e. the last transaction), 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 same the 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.

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.

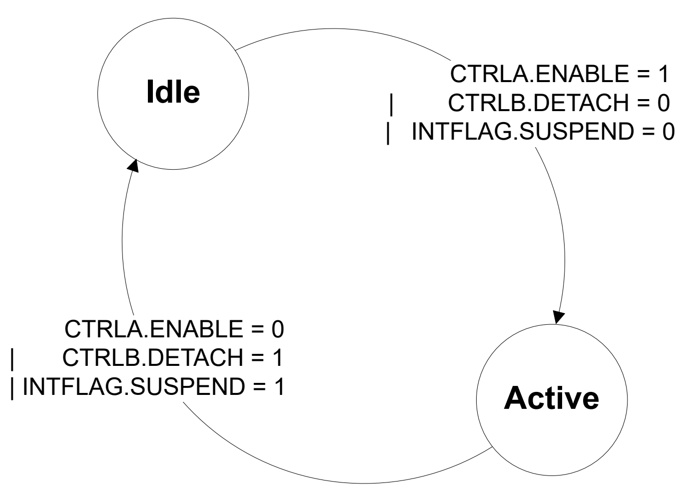
Example Configuration for Feedback Operation:

- 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

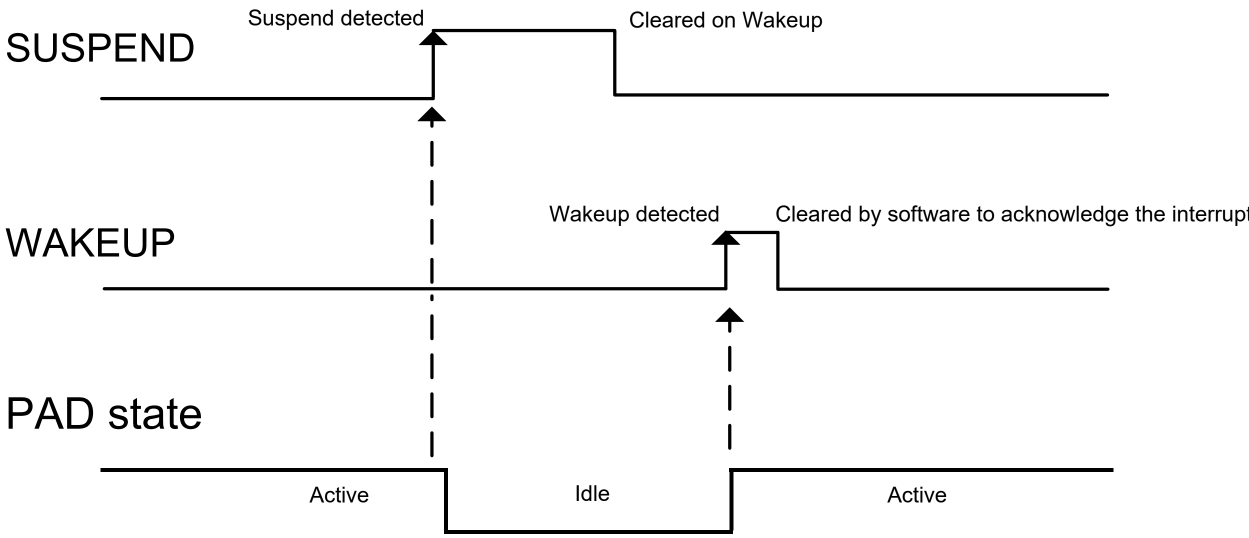


In Idle state, the pad is in Low Power Consumption mode.

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, refer to the BESL parameter in EXTREG register.

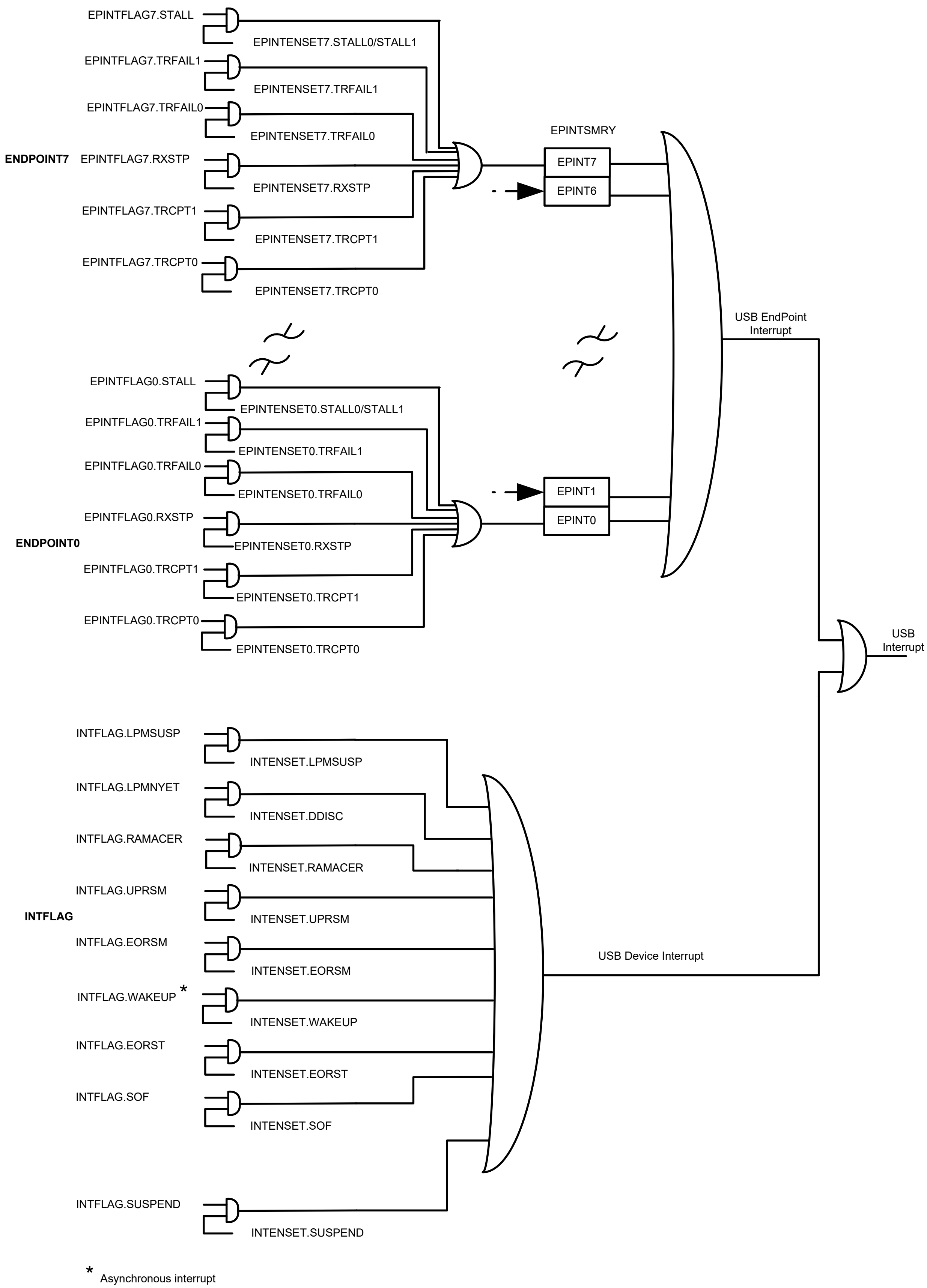
Recovering from this LPM-L1 suspend state is exactly the same as the Suspend state (see Section Suspend State and Pad Behavior) except that the remote wakeup duration initiated by USB is shorter to comply with the Link Power Management specification.

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 a STALL or no handshake, no flag is set, and the transaction is ignored.

38.6.2.16. USB Device Interrupt

Figure 38-9. Device Interrupt



The WAKEUP is an asynchronous interrupt and can be used to wake-up the device from any sleep mode.

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 Pipe Description 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 Host Control Pipe and Host Status Pipe registers.

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 the 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.

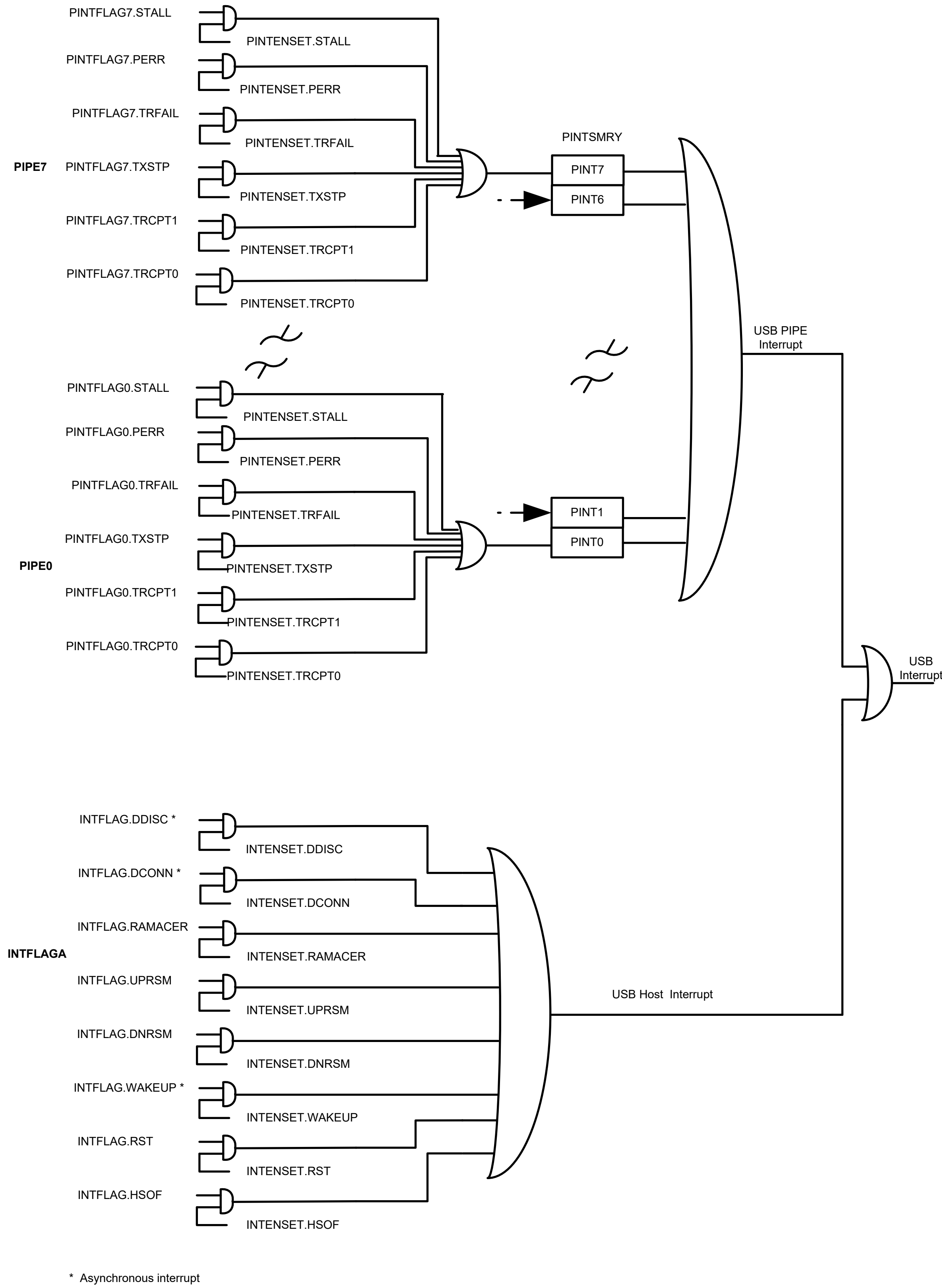
To exit the L1 STATE, the user initiate a DOWNSTREAM RESUME by setting the bit CTRLB.RESUME or a L1 RESUME by setting the Send L1 Resume bit in CTRLB (CTRLB.L1RESUME). In the case of a L1 RESUME, the K STATE duration is given by the BESL bit field in the EXTREG.VARIABLE field. Refer to the EXTREG Register.

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.

38.6.3.17. Host Interrupt

Figure 38-10. Host Interrupt



The WAKEUP is an asynchronous interrupt and can be used to wake-up the device from any sleep mode.

38.7. Communication Device Host Register Summary

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 ... 0x0C	Reserved									
0x0D	FSMSTATUS	7:0		FSMSTATE[6:0]						
0x0E ... 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]				

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
Due to synchronization there is delay from writing CTRLA.ENABLE until the peripheral is enabled/disabled. The value written to CTRLA.ENABLE will read back immediately and the Synchronization status enable bit in the Synchronization Busy register (SYNCBUSY.ENABLE) will be set. SYNCBUSY.ENABLE will be cleared when the operation is complete.
This bit is Write-Synchronized.

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.
Due to synchronization there is a delay from writing CTRLA.SWRST until the reset is complete. CTRLA.SWRST and SYNCBUSY.SWRST will both be cleared when the reset is complete.
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 the SWRST are disallowed until the 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: 0x000x0F
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					0	0	0	0

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. Finite State Machine Status

Name: FSMSTATUS
Offset: 0x0D
Reset: 0xXXXX
Property: Read only

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 (L3)	Corresponds to the powered-off, disconnected, and disabled state.
0x02	ON (L0)	Corresponds to the Idle and Active states.
0x04	SUSPEND (L2)	
0x08	SLEEP (L1)	
0x10	DNRESUME	Down Stream Resume.
0x20	UPRESUME	Up Stream Resume.
0x40	RESET	USB lines Reset.
Others		Reserved

38.7.5. 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.

38.7.6. Pad Calibration

Name: PADCAL
Offset: 0x28
Reset: 0x0000
Property: PAC Write-Protection

The Pad Calibration values must be loaded from the CAL OTP into the USB Pad Calibration register by software, after 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**Bit 14:** RISE**Bit 13:** FALL**Bit 12:** VCSH
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.

38.8. Device Registers - Common -Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00 ... 0x07	Reserved									
0x08	CTRLB	15:8					LPMHDSK[1:0]		GNAK	OPMODE2
		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 ... 0x0F	Reserved									
0x10	FNUM	15:8	FNCERR		FNUM[10:5]					
		7:0	FNUM[4:0]					MFNUM[2:0]		
0x12 ... 0x13	Reserved									
0x14	INTENCLR	15:8							LPMSUSP	LPMNYET
		7:0	RAMACER	UPRSM	EORSM	WAKEUP	EORST	SOF	MSOF	SUSPEND
0x16 ... 0x17	Reserved									
0x18	INTENSET	15:8							LPMSUSP	LPMNYET
		7:0	RAMACER	UPRSM	EORSM	WAKEUP	EORST	SOF	MSOF	SUSPEND
0x1A ... 0x1B	Reserved									
0x1C	INTFLAG	15:8							LPMSUSP	LPMNYET
		7:0	RAMACER	UPRSM	EORSM	WAKEUP	EORST	SOF	MSOF	SUSPEND
0x1E ... 0x1F	Reserved									
0x20	EPINTSMRY	15:8								
		7:0	EPINT7	EPINT6	EPINT5	EPINT4	EPINT3	EPINT2	EPINT1	EPINT0

38.8.1. 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	OPMODE2
Access					R/W	R/W	R/W	R/W
Reset					0	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	Description
0x0	No handshake. LPM is not supported.
0x1	ACK
0x2	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 8 – OPMODE2 Specific Operational Mode

Value	Description
0	The UTMI transceiver is in normal operation Mode.
1	The UTMI transceiver is in the “disabled bit stuffing and NRZI encoding” operational mode for test purpose.

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

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.

38.8.2. 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.

38.8.3. 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/W	R/W		
Reset	0	1			0	1		

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 of the device.

SPEED[1:0]	SPEED STATUS
0x0	Full-speed mode
0x1	Low-speed mode
0x2	Reserved
0x3	Reserved

38.8.4. Device Frame Number

Name: FNUM
Offset: 0x10
Reset: 0x0000
Property: Read only

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]					MFNUM[2:0]		
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 15 – 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 (or MSOF) 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.

Bits 2:0 – MFNUM[2:0] Micro Frame Number
These bits are cleared upon receiving a USB reset or at the beginning of each Start-of-Frame (SOF interrupt).
These bits are updated with the micro-frame number information as provided from the last MSOF packet even if a corrupted MSOF is received.

38.8.5. 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	MSOF	SUSPEND
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	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.

Value	Description
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 1 – MSOF Micro Start-of-Frame Interrupt Enable in High Speed Mode

Writing a zero to this bit has no effect.
Writing a one to this bit will clear the Micro Start-of-Frame interrupt Enable bit and disable the corresponding interrupt request.

Value	Description
0	The Micro Start-of-Frame interrupt is disabled.
1	The Micro Start-of-Frame interrupt is enabled and an interrupt request will be generated when the Micro Start-of-Frame Access 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.

38.8.6. 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	MSOF	SUSPEND
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	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 1 – MSOF Micro Start-of-Frame Interrupt Enable in High Speed Mode

Writing a zero to this bit has no effect.
Writing a one to this bit will set the Micro Start-of-Frame interrupt Enable bit and enable the corresponding interrupt request.

Value	Description
0	The Micro Start-of-Frame interrupt is disabled.
1	The Micro 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.

38.8.7. 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	MSOF	SUSPEND
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	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 INTENCLR/SET.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 INTENCLR/SET.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 INTENCLR/SET.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 INTENCLR/SET.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 INTENCLR/SET.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 INTENCLR/SET.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 INTENCLR/SET.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 INTENCLR/SET.SOF is one.
The FNUM is updated.
Writing a zero to this bit has no effect.

Bit 1 – MSOF Micro Start-of-Frame Interrupt Flag in High Speed Mode
This flag is cleared by writing a one to the flag.
This flag is set when a USB “Micro Start-of-Frame” has been detected (every 125 us) and will generate an interrupt if INTENCLR/SET.MSOF is one.
The MFNUM register is updated.The FNUM register is unchanged.
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 INTENCLR/SET.SUSPEND is one.
Writing a zero to this bit has no effect.

38.8.8. 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. Refer to the EPINTFLAGn register in the Device EndPoint section.

This bit will be cleared when no interrupts are pending for EndPoint n.

38.9. Device Endpoint Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00 ... 0xFF	Reserved									
0x0100	EPCFGn	7:0	NYETDIS	EPTYPE1[2:0]				EPTYPE0[2:0]		
0x0101 ... 0x0103	Reserved									
0x0104	EPSTATUSCLRn	7:0	BK1RDY	BK0RDY	STALLRQ1	STALLRQ0		CURBK	DTGLIN	DTGLOUT
0x0105	EPSTATUSSETn	7:0	BK1RDY	BK0RDY	STALLRQ1	STALLRQ0		CURBK	DTGLIN	DTGLOUT
0x0106	EPSTATUSn	7:0	BK1RDY	BK0RDY	STALLRQ1	STALLRQ0		CURBK	DTGLIN	DTGLOUT
0x0107	EPINTFLAGn	7:0		STALL1	STALL0	RXSTP	TRFAIL1	TRFAIL0	TRCPT1	TRCPT0
0x0108	EPINTENCLRn	7:0		STALL1	STALL0	RXSTP	TRFAIL1	TRFAIL0	TRCPT1	TRCPT0
0x0109	EPINTENSETn	7:0		STALL1	STALL0	RXSTP	TRFAIL1	TRFAIL0	TRCPT1	TRCPT0

38.9.1. Device Endpoint Configuration register n

Name: EPCFGn
Offset: 0x100
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
	NYETDIS	EPTYPE1[2:0]				EPTYPE0[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 – NYETDIS NYET token disable

Value	Description
0	Disable the “NYETDIS” feature: In high-speed, handshake will be handled according to the USB2.0 standard.
1	Enable the “NYETDIS” feature: An ack handshake will be sent instead of an NYET handshake in high-speed mode.

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

38.9.2. EndPoint Status Clear n

Name: EPSTATUSCLRn
Offset: 0x104
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.

38.9.3. EndPoint Status Set n

Name: EPSTATUSSETn
Offset: 0x105
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.

38.9.4. EndPoint Status n

Name: EPSTATUSn
Offset: 0x106
Reset: 0x00
Property: PAC Write-Protection

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	2		0	0	0

Bit 7 – BK1RDY Bank 1 is ready
For Control/OUT direction Endpoints, the bank is empty.
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.
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.

38.9.5. Device EndPoint Interrupt Flag n

Name: EPINTFLAGn
Offset: 0x107
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	2	0	0	2	0	2

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 TRCPT0 Interrupt Flag.

38.9.6. Device EndPoint Interrupt Enable n

Name: EPINTENCLRn
Offset: 0x108
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	2	0	0	2	0	2

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.

38.9.7. Device Interrupt EndPoint Set n

Name: EPINTENSETn
Offset: 0x109
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	2	0	0	2	0	2

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.

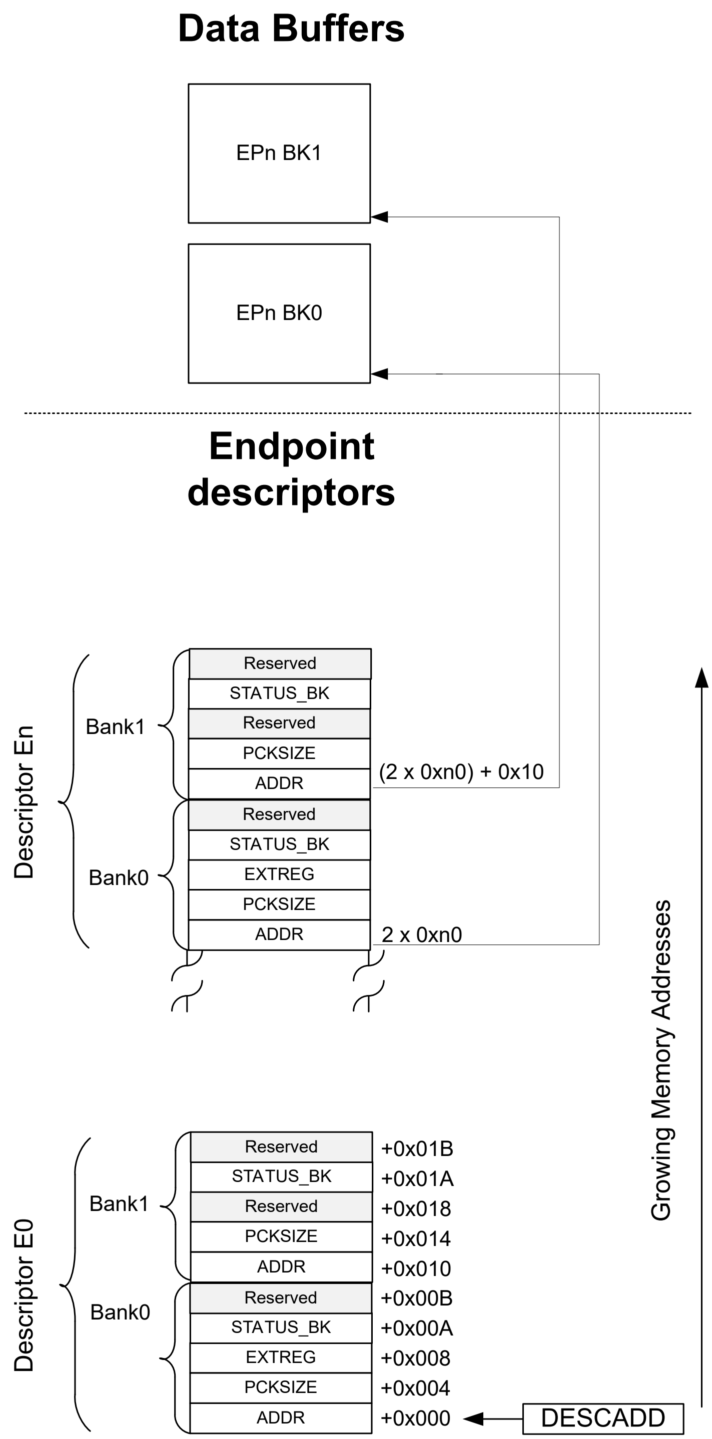
Bits 2, 3 – TRFAIL Transfer Fail bank x Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will enable the Transfer Fail interrupt.

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.

38.10. Endpoint Descriptor Structure



38.11. Device Endpoint RAM Register Summary

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

38.11.1. Address of Data Buffer

Name: ADDR
Offset: 0x00
Reset: 0xXXXXXXXX
Property: N/A

Old address offset 0x00 and 0x10

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.

38.11.2. Packet Size

Name: PCKSIZE
Offset: 0x04
Reset: 0xXXXXXXXX
Property: N/A

Original offset 0x04 & 0x14

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	x	x	x	x	x	x	x

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	0	x	x	x	x	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	0

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)
Note: 1. For isochronous endpoint 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.

38.11.3. Extended Register

Name: EXTREG
Offset: 0x08
Reset: 0xXXXXXXX
Property: N/A

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		x	x	x	x	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	0	x	x	x	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”.

38.11.4. Device Status Bank

Name: STATUS_BK
Offset: 0x0A
Reset: 0xXXXXXXX
Property: N/A

Original offset 0x0A & 0x1A

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.

38.12. Host Registers - Common - Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00 ... 0x07	Reserved									
0x08	CTRLB	15:8					L1RESUME	VBUSOK	BUSRESET	SOFE
		7:0		TSTK	TSTJ	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 ... 0x0F	Reserved									
0x10	FNUM	15:8			FNUM[10:5]					
		7:0	FNUM[4:0]					MFNUM[2:0]		
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	EPINT7	EPINT6	EPINT5	EPINT4	EPINT3	EPINT2	EPINT1	EPINT0

38.12.1. 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
		TSTK	TSTJ	AUTORESUME	SPDCONF[1:0]		RESUME	
Access		R/W	R/W	R/W	R/W	R/W	R/W	
Reset		0	0	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 the 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 6 – TSTK Test mode K

Value	Description
0	The UTMI transceiver is in normal operation Mode
1	The UTMI transceiver generates high speed K state for test purposes.

Bit 5 – TSTJ Test mode J

Value	Description
0	The UTMI transceiver is in normal operation Mode
1	The UTMI transceiver generates high speed J state for test purposes.

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, Full and High Speed capable
0x1	Reserved
0x2	Reserved
0x3	Low and Full Speed capable

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.

38.12.2. 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. Therefore, 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'.

FLENCE	Frame Timing	Internal Frame Length Down-Counter Load Value
0	Internal Frame Length (Full Speed)	11999 (1ms frame rate at 12MHz)
0	Internal Frame Length in Low and Full speed	59999 (1ms frame rate at 60MHz)
	Internal Frame Length in High speed	7499 (0.125ms micro-frame rate at 60MHz)
1	Beginning of Frame	FLENC[3:0]
	Internal Frame Length with Frame correction	11999 + FLENC[3:0] at all speeds.

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.

38.12.3. Status

Name: STATUS
Offset: 0x0C
Reset: 0x00
Property: Read only

Bit	7	6	5	4	3	2	1	0
	LINESTATE[1:0]				SPEED[1:0]			
Access	R	R			R/W	R/W		
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 High-speed mode
0x2	Low-speed mode
0x3	Reserved

38.12.4. 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]					MFNUM[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 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 and the MFNUM bits are cleared.
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).

Bits 2:0 – MFNUM[2:0] Micro Frame Number

These bits are tied to zero when operating in full-speed mode.
These bits contains the current Micro Frame number (can vary from 0 to 7) updated every 125 us.

38.12.5. Host Frame Length

Name: FLENHIGH
Offset: 0x12
Reset: 0x00
Property: Read-Only

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 38-1. 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.
Full Speed	With a USB clock running at 60MHz, counter length is 60000 to ensure a SOF generation every 1 ms.
High Speed	With a USB clock running at 60MHz, counter length is 7500 to ensure a SOF generation every 125 μs.

38.12.6. 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 Disable
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 Disable
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 Disable
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 Disable
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.

Value	Description
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 Disable

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 Disable

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 Disable

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 Disable

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.

38.12.7. 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 bit and enable the DDSIC interrupt.

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 bit and enable the DCONN interrupt.

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 bit and enable the RAMACER interrupt.

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 bit and enable the UPRSM interrupt.

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 DNRSM interrupt.

Value	Description
0	The Down Resume interrupt is disabled.
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 WAKEUP 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 Bus RST interrupt.

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 HSOF interrupt.

Value	Description
0	The Host Start-of-Frame interrupt is disabled.
1	The Host Start-of-Frame interrupt is enabled.

38.12.8. 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:
l The host controller is in suspend mode (SOFE is zero) and an upstream resume from the device is detected.

l The host controller is in suspend mode (SOFE is zero) and an device disconnection is detected.
l 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.

38.12.9. Pipe Interrupt Summary

Name: PINTSMRY
Offset: 0x20
Reset: 0x0000
Property: Read-only

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

The flag EPINTn is set when an interrupt is triggered by the pipe n. See the PINTFLAG 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.

38.13. Host Registers - Pipe - Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00 ... 0xFF	Reserved									
0x0100	PCFGn	7:0			PTYPE[2:0]			BK	PTOKEN[1:0]	
0x0101 ... 0x0102	Reserved									
0x0103	BINTERVAL	7:0	BINTERVAL[7:0]							
0x0104	PSTATUSCLR	7:0	BK1RDY	BK0RDY		PFREEZE		CURBK		DTGL
0x0105	PSTATUSSET	7:0	BK1RDY	BK0RDY		PFREEZE		CURBK		DTGL
0x0106	PSTATUS	7:0	BK1RDY	BK0RDY		PFREEZE		CURBK		DTGL
0x0107	PINTFLAG	7:0			STALL	TXSTP	PERR	TRFAIL	TRCPT1	TRCPT0
0x0108	PINTENCLR	7:0			STALL	TXSTP	PERR	TRFAIL	TRCPT1	TRCPT0
0x0109	PINTENSET	7:0			STALL	TXSTP	PERR	TRFAIL	TRCPT1	TRCPT0

38.13.1. Host Pipe n Configuration

Name: PCFGn
Offset: 0x100
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 (1)	Description
0x0	Single-bank endpoint
0x1	Dual-bank endpoint

1. Bank field is ignored when PTYPE is configured as EXTENDED.

Value	Description
0	A single bank is used for the pipe.
1	A dual bank is used for the pipe.

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. 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.

38.13.2. Interval for the Bulk-Out/Ping Transaction

Name: BINTERVAL
Offset: 0x103
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

PCFGn.PINGEN	BINTERVAL	Description
0	=0	Multiple consecutive OUT token is sent in the same frame until it is acked by the peripheral
0	>0	One OUT token is sent every BINTERVAL micro frame until it is acked by the peripheral
1	=0	Multiple consecutive PING token is sent in the same frame until it is acked by the peripheral
1	>0	One PING 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.

38.13.3. Pipe Status Clear n

Name: PSTATUSCLR
Offset: 0x104
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.

38.13.4. Pipe Status Set Register n

Name: PSTATUSSET
Offset: 0x105
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.

38.13.5. Pipe Status Register n

Name: PSTATUS
Offset: 0x106
Reset: 0x00
Property: PAC Write-Protection

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 the PINTFLAG 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.
1	The PID of the next expected transaction will be one: data 1.

38.13.6. Host Pipe Interrupt Flag Register

Name: PINTFLAG
Offset: 0x107
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	2

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 Transfer 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.

38.13.7. Host Pipe Interrupt Clear Register

Name: PINTENCLR
Offset: 0x108
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	2

Bit 5 – STALL Received Stall Interrupt Disable
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 Disable
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 Disable
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 Disable
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 Disable
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.
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.

38.13.8. Host Interrupt Pipe Set Register

Name: PINTENSET
Offset: 0x109
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	2

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.

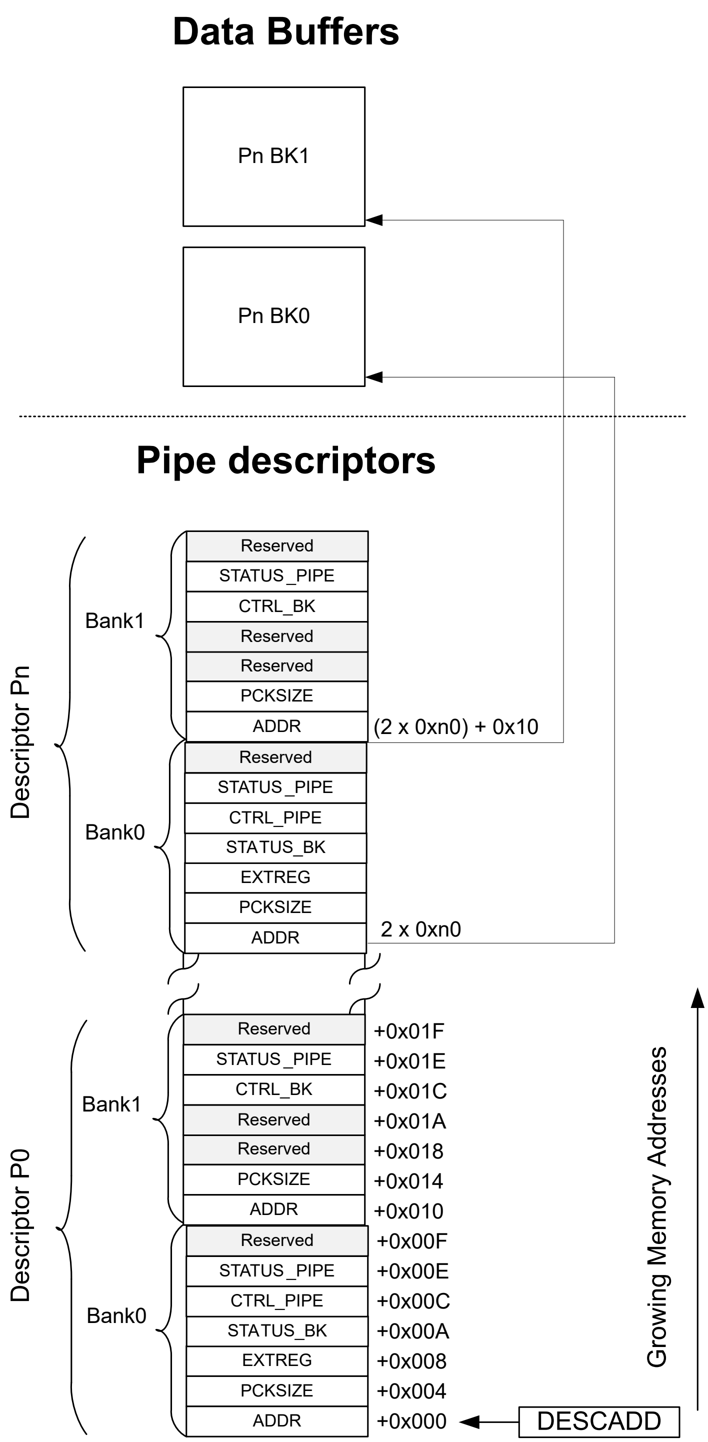
Bit 2 – TRFAIL Transfer Fail Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will enable the Transfer Fail interrupt.

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.

38.14. Pipe Descriptor Structure



38.15. Host Registers - Pipe RAM - Register Summary

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

38.15.1. Address of the Data Buffer

Name: ADDR
Offset: 0x00
Reset: 0xxxxxxxxx
Property: N/A

Original offset 0x00 & 0x10

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	0	0	0	0

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

Bit	15	14	13	12	11	10	9	8
	ADDR[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

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

Bits 31:0 – ADDR[31:0] 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.

38.15.2. Packet Size

Name: PCKSIZE
Offset: 0x04
Reset: 0xXXXXXXX
Property: N/A

Original offset 0x04 & 0x14

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	x	x	x	x	x	x	x

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	0	x	x	x	x	0	0

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)

Note:

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.

38.15.3. Extended Register

Name: EXTREG
Offset: 0x08
Reset: 0xXXXXXXX
Property: N/A

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		x	x	x	x	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	0	x	x	x	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

Notes:
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”.

38.15.4. Host Status Bank

Name: STATUS_BK
Offset: 0x0A
Reset: 0xXXXXXXX
Property: N/A

Original offset 0x0A & 0x1A

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.

38.15.5. Host Control Pipe

Name: CTRL_PIPE
Offset: 0x0C
Reset: 0xXXXX
Property: PAC Write-Protection, Write-Synchronized, Read-Synchronized

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	x	x	x	x	x	x	x	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		x	x	x	x	0	0	0

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.

38.15.6. Host Status Pipe

Name: STATUS_PIPE
Offset: 0x0E
Reset: 0xXXXXXXX
Property: PAC Write-Protection, Write-Synchronized, Read-Synchronized

Original offset 0x0E & 0x1E

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	x	x	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. Controller Area Network (CAN)

39.1. Overview

The CAN communication controller provides hardware support for the CAN protocol version 2.0 part A and B plus ISO 11898-1:2015 which includes the CAN Flexible Data Rate Frame format (CAN-FD). The CAN module uses system RAM, which it accesses through a host AHB bus for FIFO and filter storage.

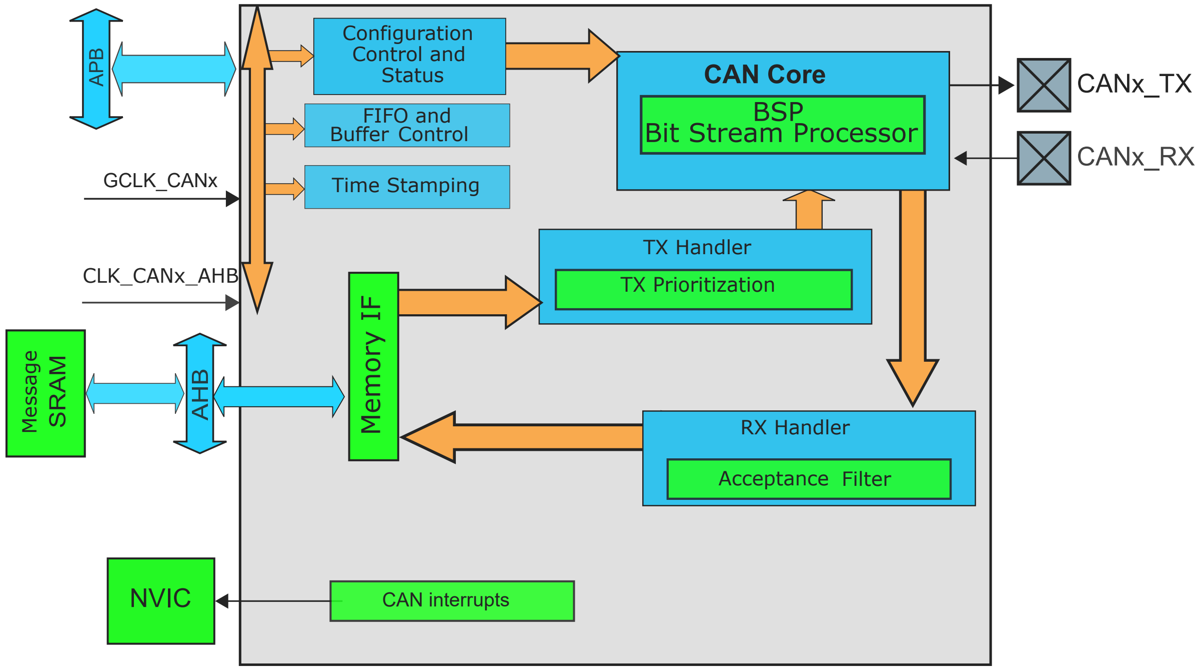
39.2. Features

The following are key features of the CAN module:

- Conforms with CAN protocol version 2.0 part A and B plus ISO 11898-1:2015
- Up to two Controller Area Network
  - Supporting CAN2.0 A/B and CAN-FD (ISO 11898-1:2015)
- CAN FD with up to 64 data bytes supported
- CAN Error Logging
- Acceptance filtering
  - Up to 128 configurable filter elements
  - Special filter for SAE J1939 IDs
  - Separate signaling on reception of High Priority Messages
- AUTOSAR optimized
- SAE J1939 optimized
- Two configurable Receive FIFOs
- Separate signaling on reception of High-Priority Messages
- Up to 64 dedicated Receive Buffers and up to 32 dedicated Transmit Buffers
- Configurable Transmit FIFO, Transmit Queue, Transmit Event FIFO
- Direct Message RAM access for CPU
- Programmable Loop-Back Test mode
- Maskable module interrupts
- Power-down support; Debug on CAN support
- Transfer rates:
  - 1 Mb/s for CAN 2.0 mode
  - 10 Mb/s for CAN-FD mode

39.3. Block Diagram

Figure 39-1. CAN Block Diagram



Note: x = 0, 1. This family of devices contain up to two CAN modules.

39.4. Signal Description

Table 39-1. Signal Description

PIN NAME	DESCRIPTION	TYPE
CAN0_RX	CAN0_FD Receive	Digital Input
CAN0_TX	CAN0_FD Transmit	Digital Output
CAN1_RX	CAN1_FD Receive	Digital Input
CAN1_TX	CAN1_FD Transmit	Digital Output

39.5. Peripheral Dependencies

Peripheral Name	Base Address	NVIC IRQ Index:Source	MCLK AHBx/APBx Clock Enable Mask Bit	GCLK Peripheral Channel Clock Name : Register	PAC Peripheral ID (PAC.WRCTRL.PERIDx)	DMA Trigger Index:Source (DMAC.CHCTRLBk.TRIGx)	Power Domain
CAN0	0x4501_E000	143 : LINE0, LINE1, ERROR	MCLK.CLKMSK0[13]	GCLK_CAN0 : GCLK.PCHCTRL[39]	51	68 : DEBUG	VDDREG
CAN1	0x4502_0000	144: LINE0, LINE1, ERROR	MCLK.CLKMSK0[14]	GCLK_CAN1 : GCLK.PCHCTRL[40]	52	69 : DEBUG	VDDREG

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

39.5.1. I/O Lines

Using the CAN's I/O lines requires the I/O pins to be configured.

References:

PORT - I/O Pin Controller

39.5.2. Clocks

An AHB clock CLK_CANx_AHB (where x = 0, 1) is required to clock the CAN. The CAN AHB BUS interface clocks are enabled by default on reset. This clock can be configured in the Main Clock peripheral (MCLK) before using the CAN, and the default state of CLK_CANx_AHB can be found in the AHBMASK register of the MCLK (Main Clock Controller) module.

A generic clock GCLK_CANx (where x = 0, 1) is required to clock the CAN. This clock must be configured and enabled in the generic clock controller before using the CAN.

This generic clock is asynchronous to the bus clock (CLK_CANx_AHB). Due to this asynchronicity, writes to certain registers will require synchronization between the clock domains.

References:

Peripheral Clock Masking

Generic Clock Controller - GCLK

39.5.3. DMA

The CAN has a built-in Direct Memory Access (DMA) and will read/write data to/from the system RAM when a CAN transaction takes place. No CPU or *DMA Controller (DMAC)* resources are required.

The DMAC can be used for debug messages functionality.

39.6. Functional Description

39.6.1. Principle of Operation

The CAN performs communication according to ISO 11898-1:2015 (identical to Bosch CAN protocol specification 2.0 part A,B, ISO CAN FD).

For FIFO and filter storage, the CAN module uses system RAM, which it accesses through a host AHB bus. Each CAN FD instance may use up to 4864 bytes of system memory.

All functions concerning the handling of messages are implemented by the Rx Handler and the Tx Handler. The Rx Handler manages message acceptance filtering, the transfer of received messages from the CAN Core to the Message RAM as well as providing receive message status information. The Tx Handler is responsible for the transfer of transmit messages from the Message RAM to the CAN Core as well as providing transmit status information.

Acceptance filtering is implemented by a combination of up to 128 filter elements where each one can be configured as a range, as a bit mask, or as a dedicated ID filter.

39.6.2. Operating Modes

39.6.2.1. Software Initialization

Software initialization is started by setting the CCCR.INIT bit (CCCR <0>), either by software, or by going “bus off.” While the CCCR.INIT bit (CCCR <0>) is set, message transfer from and to the CAN bus is stopped, the status of the CAN bus output CANx_TX is ”recessive” (HIGH). The counters of the Error Management Logic EML are unchanged. Setting the CCCR.INIT bit (CCCR <0>) does not change any configuration register. Resetting the CCCR.INIT bit (CCCR <0>) finishes the software initialization.

Afterwords the Bit Stream Processor (BSP) synchronizes itself to the data transfer on the CAN bus by waiting for the occurrence of a sequence of 11 consecutive ”recessive” bits (= Bus_Idle) before it can take part in bus activities and start the message transfer.

Access to the CAN configuration registers is only enabled when the CCCR.CCE bit (CCCR <1>) and the CCCR.INIT bit (CCCR <0>) are set (protected write).

The CCCR.CCE bit (CCCR <1>) can only be set and cleared while the CCCR.INIT bit (CCCR <0>) = ‘1’. The CCCR.CCE bit (CCCR <1>) is automatically cleared when the CCCR.INIT bit (CCCR <0>) is cleared.

The following registers are reset when the CCCR.CCE bit (CCCR <1>) is set

- HPMS - High Priority Message Status
- RXF0S - Rx FIFO 0 Status
- RXF1S - Rx FIFO 1 Status
- TXFQS - Tx FIFO/Queue Status
- TXBRP - Tx Buffer Request Pending
- TXBTO - Tx Buffer Transmission Occurred
- TXBCF - Tx Buffer Cancellation Finished
- TXEFS - Tx Event FIFO Status

The Timeout Counter value (TOCV.TOC) bits (TOCV <15:0>) is preset to the value configured by the TOCC.TOP bits (TOCC <31:16>) when the CCCR.CCE bit (CCCR <1>) is set.

In addition, the state machines of the Tx Handler and Rx Handler are held in idle state while the CCCR.CCE bit (CCCR <1>) = ‘1’.

The following registers are only writable while the CCCR.CCE bit (CCCR <1>) = ‘0’.

- TXBAR - Tx Buffer Add Request
- TXBCR - Tx Buffer Cancellation Request

The CCCR.TEST bit (CCCR <7>) and CCCR.MON bit (CCCR <5>) can only be set by the CPU while the CCCR.INIT bit (CCCR <0>) = ‘1’ and CCR.CCE bit (CCCR <1>) = ‘1’. Both bits may be cleared at any time. The CCCR.DAR bit (CCCR <6>) can only be set/cleared while the CCCR.INIT bit (CCCR <0>) = ‘1’ and the CCCR.CCE bit (CCCR <1>) = ‘1’.

39.6.2.2. Normal Operation

Once the CAN is initialized and CCCR.INIT bit (CCCR <0>) is reset to ‘0’, the CAN synchronizes itself to the CAN bus and is ready for communication.

After passing the acceptance filtering, received messages including Message ID and DLC are stored into a dedicated Rx Buffer or into Rx FIFO0 or Rx FIFO1.

For messages to be transmitted dedicated Tx Buffers and/or a Tx FIFO or a Tx Queue can be initialized or updated. Automated transmission on reception of remote frames is not implemented.

39.6.2.3. CAN FD Operation

There are two variants in the CAN FD frame format, first the CAN FD frame without bit rate switching where the data field of a CAN frame may be longer than 8 bytes. The second variant is the CAN FD frame where control field, data field, and CRC field of a CAN frame are transmitted with a higher bit rate than the beginning and the end of the frame.

The previously reserved bit in standard CAN frames with 11-bit identifiers and the first previously reserved bit in extended CAN frames with 29-bit identifiers will now be decoded as FDF bit. A recessive FDF signifies a CAN FD frame, where as a dominant FDF bit signifies a Classic CAN frame. In a CAN FD frame, the two bits following FDF, (i.e.,res and BRS), decide whether the bit rate inside of this CAN FD frame is switched. A CAN FD bit rate switch is signified by a dominant res bit and a recessive BRS bit. The coding of a recessive res bit is reserved for future expansion of the protocol.

In case the CAN receives a frame with a recessive FDF bit and a recessive res bit, it will signal a Protocol Exception Event by setting bit PSR.PXE bit (PSR <14>). When Protocol Exception Handling is enabled (CCCR.PXHD bit (CCCR<12>) = ‘0’), this causes the operation state to change from Receiver (PSR.ACT = “0x2”) to synchronizing on CAN communication (PSR.ACT bits (PSR <4:3>) = “0x0”) at the next sample point. In case Protocol Exception Handling is disabled (CCCR.PXHD bit (CCCR<12>) = ‘1’), the CAN will treat a recessive res bit as a form error and will respond with an error frame.

CAN FD operation is enabled by programming CCCR.FDOE bit (CCCR<8>). In case CCCR.FDOE bit (CCCR<8>) = ‘1’, transmission and reception of CAN FD frames is enabled. Transmission and reception of Classic CAN frames is always possible. Whether a CAN FD frame or a Classic CAN frame is transmitted can be configured via FDF bit in the respective Tx Buffer element. With CCCR.FDOE bit (CCCR<8>) = ‘0’, received frames are interpreted as Classic CAN frames, witch leads to the transmission of an error frame when receiving a CAN FD frame. When CAN FD operation is disabled, no CAN FD frames are transmitted even if FDF bit of a Tx Buffer element is set. CCCR.FDOE bit (CCCR<8>) and CCCR.BRSE bit (CCCR<9>) can only be changed while CCCR.INIT bit (CCCR <0>) and CCCR.CCE (CCCR <1>) are both set.

With CCCR.FDOE bit (CCCR<8>) = ‘0’, the setting of FDF and BRS bits is ignored and frames are transmitted in Classic CAN format. With CCCR.FDOE bit (CCCR<8>) = ‘1’ and CCCR.BRSE bit (CCCR<9>) = ‘0’, only bit FDF of a Tx Buffer element is evaluated. With CCCR.FDOE bit (CCCR<8>) = ‘1’ and CCCR.BRSE bit (CCCR<9>) = ‘1’, transmission of CAN FD frames with bit rate switching is enabled. All Tx Buffer elements with FDF and BRS bits set are transmitted in CAN FD format with bit rate switching.

A mode change during CAN operation is only recommended under the following conditions:

- The failure rate in the CAN FD data phase is significantly higher than in the CAN FD arbitration phase. In this case disable the CAN FD bit rate switching option for transmissions.
- During system startup all nodes are transmitting Classic CAN messages until it is verified that they are able to communicate in CAN FD format. If this is true, all nodes switch to CAN FD operation.
- Wake-up messages in CAN Partial Networking have to be transmitted in Classic CAN format.
- End-of-line programming in case not all nodes are CAN FD capable. Non CAN FD nodes are held in silent mode until programming has completed. Then all nodes switch back to Classic CAN communication.

In the CAN FD format, the coding of the DLC differs from the standard CAN format. The DLC codes 0 to 8 have the same coding as in standard CAN. However, the codes 9 to 15, (which in standard CAN all code a data field of 8 bytes), are coded according to the following table.

Table 39-2. Coding of DLC in CAN FD

DLC	9	10	11	12	13	14	15
Number of Data Bytes	12	16	20	24	32	48	64

In CAN FD frames, the bit timing will be switched inside the frame, after the BRS (Bit Rate Switch) bit, if this bit is recessive. Before the BRS bit, in the CAN FD arbitration phase, the nominal CAN bit timing is used as defined by the Nominal Bit Timing & Prescaler Register (NBTP). In the following CAN FD data phase, the fast CAN bit timing is used as defined by the Data Bit Timing & Prescaler Register (DBTP). The bit timing is switched back from the fast timing at the CRC delimiter or when an error is detected, whichever occurs first.

The maximum configurable bit rate in the CAN FD data phase depends on the CAN clock frequency (GCLK_CANx). Example: with a CAN clock frequency of 20MHz and the shortest configurable bit time of 4 t_q, the bit rate in the data phase is 5 Mbit/s.

In both data frame formats, CAN FD long and CAN FD fast, the value of the bit ESI (Error Status Indicator) is determined by the transmitter's error state at the start of the transmission. If the transmitter is error passive, ESI is transmitted recessive, else it is transmitted dominant.

39.6.2.4. Transceiver Delay Compensation

During the data phase of a CAN FD transmission only one node is transmitting, all others are receivers. The length of the bus line has no impact. When transmitting via pin CANx_TX the CAN receives the transmitted data from its local CAN transceiver via pin CANx_RX. The received data is delayed by the CAN transceiver's loop delay. In case this delay is greater than TSEG1 (time segment before sample point), a bit error is detected. In order to enable a data phase bit time that is even shorter than the transceiver loop delay, the delay compensation is introduced. Without transceiver delay compensation, the bit rate in the data phase of a CAN FD frame is limited by the transceivers loop delay.

The CAN's protocol unit has implemented a delay compensation mechanism to compensate the transmitter delay, thereby enabling transmission with higher bit rates during the CAN FD data phase independent of the delay of a specific CAN transceiver.

To check for bit errors during the data phase of transmitting nodes, the delayed transmit data is compared against the received data at the Secondary Sample Point SSP. If a bit error is detected, the transmitter will react on this bit error at the next following regular sample point. During arbitration phase the delay compensation is always disabled.

The transmitter delay compensation enables configurations where the data bit time is shorter than the transmitter delay, it is described in detail in the new ISO11898-1. It is enabled by setting the DBTP.TDC bit (DBTP<23>).

The received bit is compared against the transmitted bit at the SSP. The SSP position is defined as the sum of the measured delay from the CAN's transmit output CANx_TX through the transceiver to the receive input CANx_RX plus the transmitter delay compensation offset as configured by TDCR.TDCO bits (TDCR <15:8>). The transmitter delay compensation offset is used to adjust the position of the SSP inside the received bit (e.g. half of the bit time in the data phase). The position of the secondary sample point is rounded down to the next integer number of minimum time quantum (mtq). 1 mtq is equal to time period of GCLK_CANx clock.

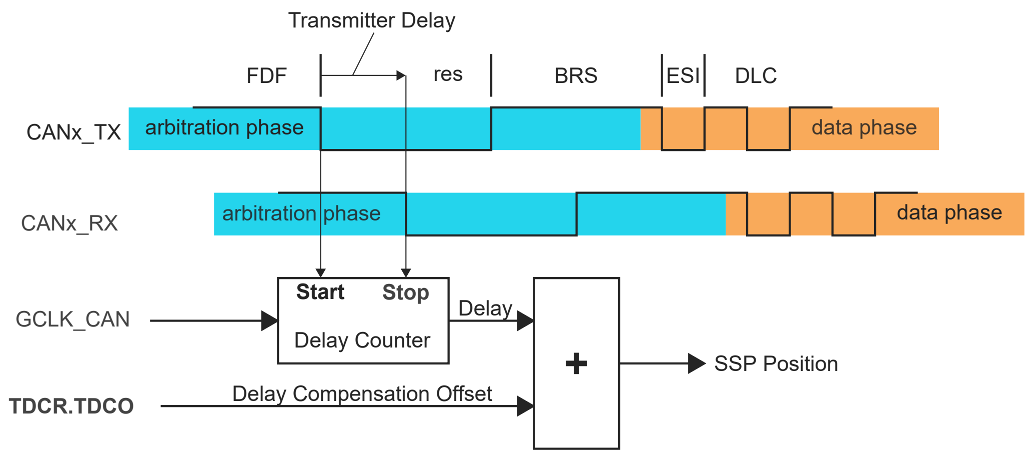
The PSR.TDCV bits (PSR <22:16>) show the actual transmitter delay compensation value. PSR.TDCV bits (PSR <22:16>) are cleared when CCCR.INIT bit (CCCR <0>) is set and is updated at each transmission of an FD frame while DBTP.TDC bit (DBTP <23>) is set.

The following boundary conditions have to be considered for the transmitter delay compensation implemented in the CAN:

- The sum of the measured delay from CANx_TX to CANx_RX and the configured transceiver delay compensation offset FBTP.TDCR bits (TDCR <15:8>) has to be less than 6 bit times in the data phase.
- The sum of the measured delay from CANx_TX to CANx_RX and the configured transceiver delay compensation offset FBTP.TDCR bits (TDCR <15:8>) has to be less or equal to 127 mtq. In case this sum exceeds 127 mtq, the maximum value of 127 mtq is used for transceiver delay compensation.
- The data phase ends at the sample point of the CRC delimiter, that stops checking of receive bits at the SSPs.
  Transmitter Delay Compensation Measurement

If transmitter delay compensation is enabled by programming the DBTP.TDC bit (DBTP <23>) = '1', the measurement is started within each transmitted CAN FD frame at the falling edge of bit FDF to bit res. The measurement is stopped when this edge is seen at the receive input CANx_TX of the transmitter. The resolution of this measurement is one mtq.

Figure 39-2. Transceiver delay measurement



To avoid that a dominant glitch inside the received FDF bit ends the delay compensation measurement before the falling edge of the received res bit, resulting in a too early SSP position, the use of a transmitter delay compensation filter window can be enabled by programming TDCR.TDCF bit (TDCR <6:0>). This defines a minimum value for the SSP position. Dominant edges of CANx_RX, that would result in an earlier SSP position are ignored for transmitter delay measurement. The measurement is stopped when the SSP position is at least TDCR.TDCF bit (TDCR <6:0>) and CANx_RX is low.

39.6.2.5. Restricted Operation Mode

In Restricted Operation Mode the node is able to receive data and remote frames and to give acknowledge to valid frames, but it does not send data frames, remote frames, active error frames, or overload frames. In case of an error condition or overload condition, it does not send dominant bits, instead it waits for the occurrence of bus idle condition to resynchronize itself to the CAN communication. The error counters (ECR.REC bit (ECR <14:8>), ECR.TEC bit (ECR <7:0>)) are frozen while Error Logging (ECR.CEL bit (ECR <23:16>)) is still incremented. The CPU can set the CAN into Restricted Operation mode by setting bit CCCR.ASM bit (CCCR <2>). The bit can only be set by the CPU when both CCCR.CCE bit (CCCR <1>) and CCCR.INIT bit (CCCR <0>) are set to '1'. The bit can be reset by the CPU at any time.

Restricted Operation Mode is automatically entered when the Tx Handler was not able to read data from the Message RAM in time. To leave Restricted Operation Mode, the CPU has to reset CCCR.ASM bit (CCCR <2>).

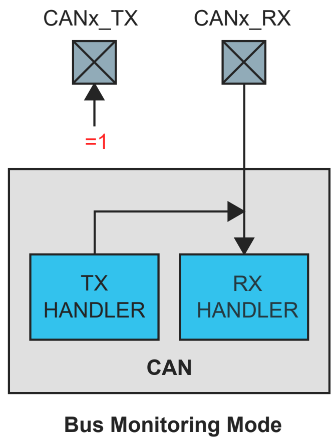
The Restricted Operation Mode can be used in applications that adapt themselves to different CAN bit rates. In this case the application tests different bit rates and leaves the Restricted Operation Mode after it has received a valid frame.

39.6.2.6. Bus Monitoring Mode

The CAN is set in Bus Monitoring Mode by programming CCCR.MON bit (CCCR <5>) to '1'. In Bus Monitoring Mode (see ISO 11898-1, 10.12 Bus monitoring), the CAN is able to receive valid data frames and valid remote frames, but cannot start a transmission. In this mode, it sends only recessive bits on the CAN bus. If the CAN is required to send a dominant bit (ACK bit, overload flag, active error flag), the bit is rerouted internally so that the CAN monitors this dominant bit, although the CAN bus may remain in recessive state. In Bus Monitoring Mode register TXBRP is held in reset state.

The Bus Monitoring Mode can be used to analyze the traffic on a CAN bus without affecting it by the transmission of dominant bits. The figure below shows the connection of signals CANx_TX and CANx_RX to the CAN in Bus Monitoring Mode.

Figure 39-3. Pin Control in Bus Monitoring Mode



39.6.2.7. Disabled Automatic Retransmission

According to the CAN Specification (see ISO 11898-1, 6.3.3 Recovery Management), the CAN provides means for automatic retransmission of frames that have lost arbitration or that have been disturbed by errors during transmission. By default automatic retransmission is enabled. To support time-triggered communication as described in ISO 11898-1, chapter 9.2, the automatic retransmission may be disabled via CCCR.DAR bit (CCCR <6>).

Frame Transmission in DAR Mode

In DAR mode all transmissions are automatically cancelled after they started on the CAN bus. A Tx Buffer's Tx Request Pending bit (TXBRP.TRPx) is reset after successful transmission, when a transmission has not yet been started at the point of cancellation, has been aborted due to lost arbitration, or when an error occurred during frame transmission.

- Successful transmission:
  - Corresponding Tx Buffer Transmission Occurred bit TXBTO.TOx set
  - Corresponding Tx Buffer Cancellation Finished bit TXBCF.CFx not set
- Successful transmission in spite of cancellation:
  - Corresponding Tx Buffer Transmission Occurred bit TXBTO.TOx set
  - Corresponding Tx Buffer Cancellation Finished bit TXBCF.CFx set
- Arbitration lost or frame transmission disturbed:
  - Corresponding Tx Buffer Transmission Occurred bit TXBTO.TOx not set
  - Corresponding Tx Buffer Cancellation Finished bit TXBCF.CFx set

In case of a successful frame transmission, and if storage of Tx events is enabled, a Tx Event FIFO element is written with Event Type ET = "10" (transmission in spite of cancellation).

39.6.2.8. Test Modes

To enable write access to register TEST, bit CCCR.TEST bit (CCCR <7>) has to be set to '1'. This allows the configuration of the test modes and test functions.

Four output functions are available for the CAN transmit pin CANx_TX by programming TEST.TX bits (TEST <6:5>). Additionally to its default function – the serial data output – it can drive the CAN Sample Point signal to monitor the CAN's bit timing and it can drive constant dominant or recessive values. The actual value at pin CANx_RX can be read from TEST.RX bit (TEST <7>). Both functions can be used to check the CAN bus' physical layer.

Due to the synchronization mechanism between GCLK_CANx and CLK_CANx_AHB domains, there may be a delay of several CLK_CANx_AHB periods between writing to TEST.TX bits (TEST <6:5>) until the new configuration is visible at output pin CANx_TX. This applies also when reading input pin CANx_RX via TEST.RX bit (TEST <7>).

Note: Test modes should be used for production tests or self test only. The software control for pin CANx_TX interferes with all CAN protocol functions. It is not recommended to use test modes for application.

External Loop Back Mode

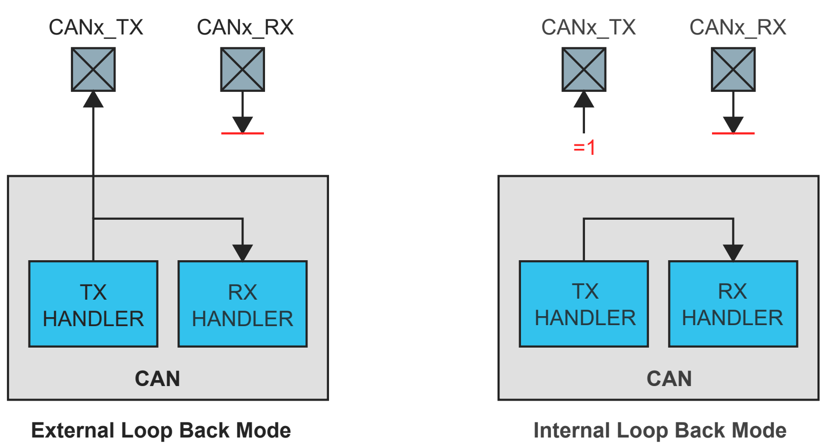
The CAN can be set in External Loop Back Mode by programming TEST.LBCK bit (TEST <4>) to ‘1’. In Loop Back Mode, the CAN treats its own transmitted messages as received messages and stores them (if they pass acceptance filtering) into an Rx Buffer or an Rx FIFO. The figure below shows the connection of signals CANx_TX and CANx_RX to the CAN in External Loop Back Mode.

This mode is provided for hardware self-test. To be independent from external stimulation, the CAN ignores acknowledge errors (recessive bit sampled in the acknowledge slot of a data/remote frame) in Loop Back Mode. In this mode the CAN performs an internal feedback from its Tx output to its Rx input. The actual value of the CANx_RX input pin is disregarded by the CAN. The transmitted messages can be monitored at the CANx_TX pin.

Internal Loop Back Mode

Internal Loop Back Mode is entered by programming bits TEST.LBCK bit (TEST <4>) and CCCR.MON bit (CCCR <5>) to ‘1’. This mode can be used for a “Hot Selftest”, meaning the CAN can be tested without affecting a running CAN system connected to the pins CANx_TX and CANx_RX. In this mode pin CANx_RX is disconnected from the CAN and pin CANx_TX is held recessive. The following figure shows the connection of CANx_TX and CANx_RX to the CAN in case of Internal Loop Back Mode.

Figure 39-4. Pin Control in Loop Back Modes



39.6.3. Timestamp Generation

For timestamp generation the CAN supplies a 16-bit wrap-around counter. A prescaler TSCC.TCP bit (TSCC <19:16>) can be configured to clock the counter in multiples of CAN bit times (1…16). The counter is readable via TSCV.TSC bit (TSCV <15:8>). A write access to register TSCV resets the counter to zero. When the timestamp counter wraps around interrupt flag IR.TSW bit (TSW <16>) is set.

On start of frame reception / transmission the counter value is captured and stored into the timestamp section of an Rx Buffer / Rx FIFO (RXTS[15:0]) or Tx Event FIFO (TXTS[15:0]) element.

39.6.4. Timeout Counter

To signal timeout conditions for Rx FIFO 0, Rx FIFO 1, and the Tx Event FIFO the CAN supplies a 16-bit Timeout Counter. It operates as down-counter and uses the same prescaler controlled by TSCC.TCP bit (TSCC <19:16>) as the Timestamp Counter. The Timeout Counter is configured via register TOCC. The actual counter value can be read from TOCV.TOC bits (TOCV <15:0>). The Timeout Counter can only be started while CCCR.INIT bit (CCCR <0>) = '0'. It is stopped when CCCR.INIT bit (CCCR <0>) = '1', e.g. when the CAN enters "bus off" state.

The operation mode is selected by TOCC.TOS bits (TOCC <2:1>). When operating in Continuous Mode, the counter starts when CCCR.INIT bit (CCCR <0>) is reset. A write to TOCV presets the counter to the value configured by TOCC.TOP bits (TOCC <31:16>) and continues down-counting.

When the Timeout Counter is controlled by one of the FIFOs, an empty FIFO presets the counter to the value configured by TOCC.TOP bits (TOCC <31:16>). Down-counting is started when the first FIFO element is stored. Writing to TOCV has no effect.

When the counter reaches zero, interrupt flag IR.TOO bit (IR<18>) is set. In Continuous Mode, the counter is immediately restarted at TOCC.TOP bits (TOCC <31:16>).

Note: The clock signal for the Timeout Counter is derived from the CAN Core's sample point signal. Therefore the point in time where the Timeout Counter is decremented may vary due to the synchronization / re-synchronization mechanism of the CAN Core. If the baud rate switch feature in CAN FD is used, the timeout counter is clocked differently in arbitration and data field.

39.6.5. Rx Handling

The Rx Handler controls the acceptance filtering, the transfer of received messages to the Rx Buffers or to one of the two Rx FIFOs, as well as the Rx FIFO's Put and Get Indices.

39.6.5.1. Acceptance Filtering

The CAN offers the possibility to configure two sets of acceptance filters, one for standard identifiers and one for extended identifiers. These filters can be assigned to an Rx Buffer or to Rx FIFO 0,1. For acceptance filtering each list of filters is executed from element #0 until the first matching element. Acceptance filtering stops at the first matching element. The following filter elements are not evaluated for this message.

The main features are:

- Each filter element can be configured as
  - range filter (from - to)
  - filter for one or two dedicated IDs
  - classic bit mask filter
- Each filter element is configurable for acceptance or rejection filtering
- Each filter element can be enabled / disabled individually
- Filters are checked sequentially, execution stops with the first matching filter element

Related configuration registers are:

- Global Filter Configuration GFC
- Standard ID Filter Configuration SIDFC
- Extended ID Filter Configuration XIDFC
- Extended ID AND Mask XIDAM

Depending on the configuration of the filter element (SFEC/EFEC) a match triggers one of the following actions:

- Store received frame in FIFO 0 or FIFO 1

- Store received frame in Rx Buffer
- Store received frame in Rx Buffer and generate pulse at filter event pin
- Reject received frame
- Set High Priority Message interrupt flag IR.HPM (IR <8>)
- Set High Priority Message interrupt flag IR.HPM (IR <8>) and store received frame in FIFO 0 or FIFO 1

Acceptance filtering is started after the complete identifier has been received. After acceptance filtering has completed, and if a matching Rx Buffer or Rx FIFO has been found, the Message Handler starts writing the received message data in portions of 32 bit to the matching Rx Buffer or Rx FIFO. If the CAN protocol controller has detected an error condition (e.g. CRC error), this message is discarded with the following impact on the affected Rx Buffer or Rx FIFO:

Rx Buffer

New Data flag of matching Rx Buffer is not set, but Rx Buffer (partly) overwritten with received data. For error type see PSR.LEC bits (PSR <2:0>) respectively PSR.DLEC bit field (PSR<10:8>).

Rx FIFO

Put index of matching Rx FIFO is not updated, but related Rx FIFO element (partly) overwritten with received data. For error type see PSR.LEC bits (PSR <2:0>) respectively PSR.DLEC bit field (PSR<10:8>). In case the matching Rx FIFO is operated in overwrite mode, the boundary conditions described in *Rx FIFO Overwrite Mode* have to be considered.

Note: When an accepted message is written to one of the two Rx FIFOs, or into an Rx Buffer, the unmodified received identifier is stored independent of the filter(s) used. The result of the acceptance filter process is strongly depending on the sequence of configured filter elements.

Range Filter

The filter matches for all received frames with Message IDs in the range defined by SF1ID/SF2ID for standard frames or EF1ID/EF2ID for extended frames.

There are two possibilities when range filtering is used together with extended frames:

EFT = “00”	The Message ID of received frames is AND’ed with the Extended ID AND Mask (XIDAM) before the range filter is applied
EFT = “11”	The Extended ID AND Mask (XIDAM) is not used for range filtering

Filter for specific IDs

A filter element can be configured to filter for one or two specific Message IDs. To filter for one specific Message ID, the filter element has to be configured with SF1ID = SF2ID resp. EF1ID = EF2ID.

Classic Bit Mask Filter

Classic bit mask filtering is intended to filter groups of Message IDs by masking single bits of a received Message ID. With classic bit mask filtering SF1ID/EF1ID is used as Message ID filter, while SF2ID/EF2ID is used as filter mask.

A zero bit at the filter mask will mask out the corresponding bit position of the configured ID filter, e.g. the value of the received Message ID at that bit position is not relevant for acceptance filtering. Only those bits of the received Message ID where the corresponding mask bits are one are relevant for acceptance filtering.

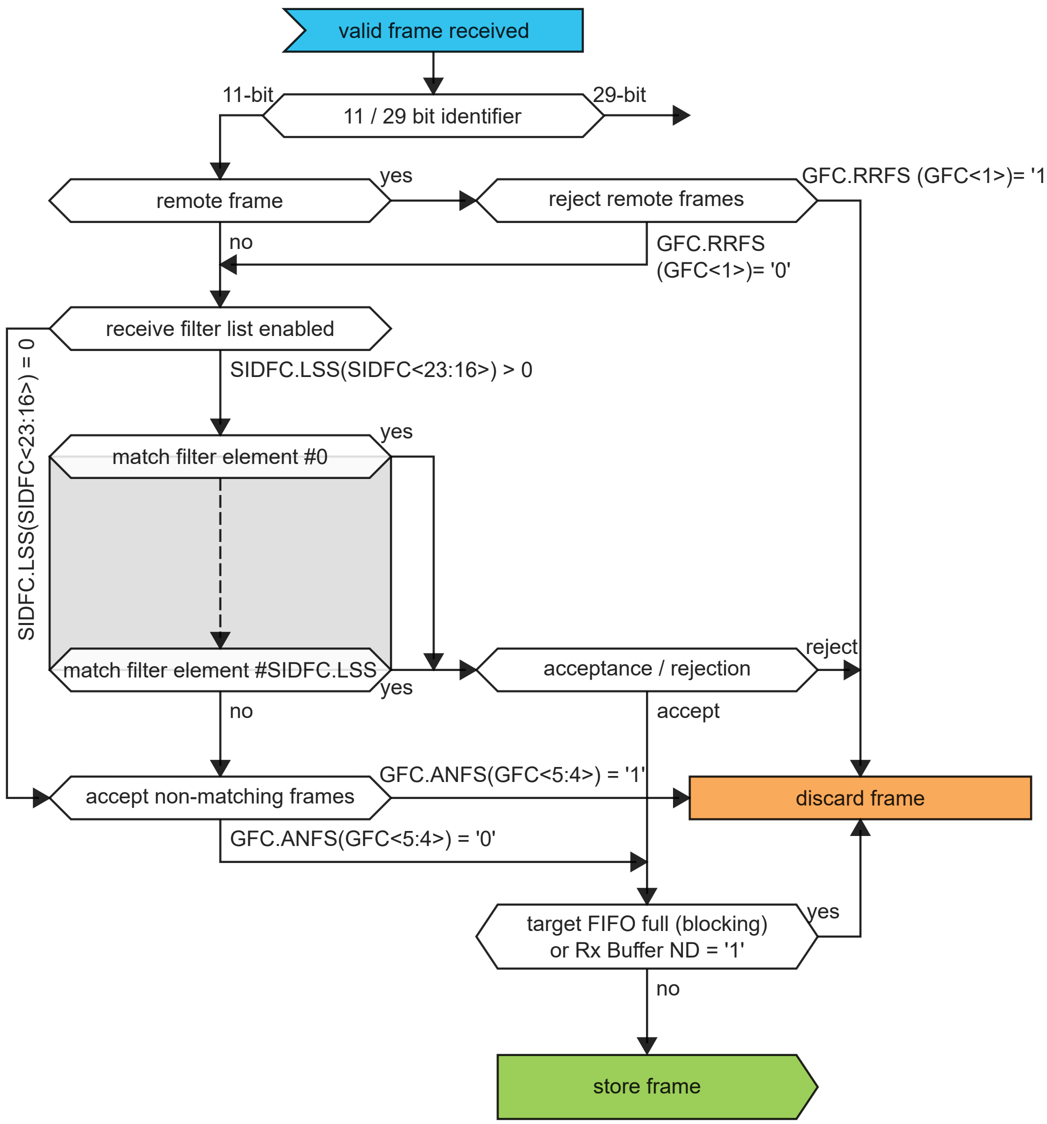
In case all mask bits are one, a match occurs only when the received Message ID and the Message ID filter are identical. If all mask bits are zero, all Message IDs match.

Standard Message ID Filtering

The figure below shows the flow for standard Message ID (11-bit Identifier) filtering. The Standard Message ID Filter element is described in Standard Message ID Filter Element.

Controlled by the Global Filter Configuration GFC and the Standard ID Filter Configuration SIDFC Message ID, Remote Transmission Request bit (RTR), and the Identifier Extension bit (IDE) of received frames are compared against the list of configured filter elements.

Figure 39-5. Standard Message ID Filtering



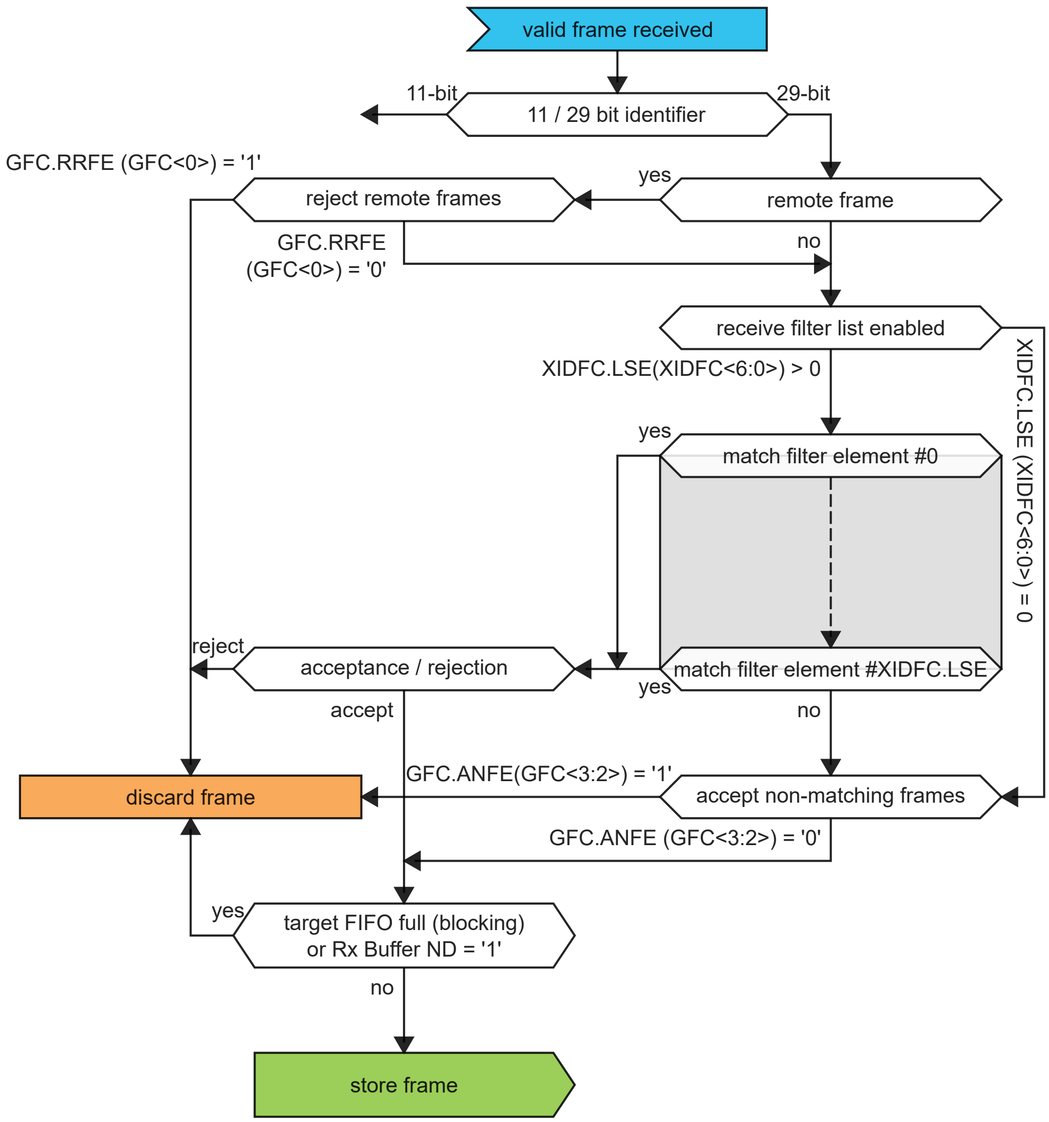
Extended Message ID Filtering

The figure below shows the flow for extended Message ID (29-bit Identifier) filtering. The Extended Message ID Filter element is described in *Extended Message ID Filter Element*.

Controlled by the Global Filter Configuration GFC and the Extended ID Filter Configuration XIDFC Message ID, Remote Transmission Request bit (RTR), and the Identifier Extension bit (IDE) of received frames are compared against the list of configured filter elements.

The Extended ID AND Mask XIDAM is AND'ed with the received identifier before the filter list is executed.

Figure 39-6. Extended Message ID Filtering



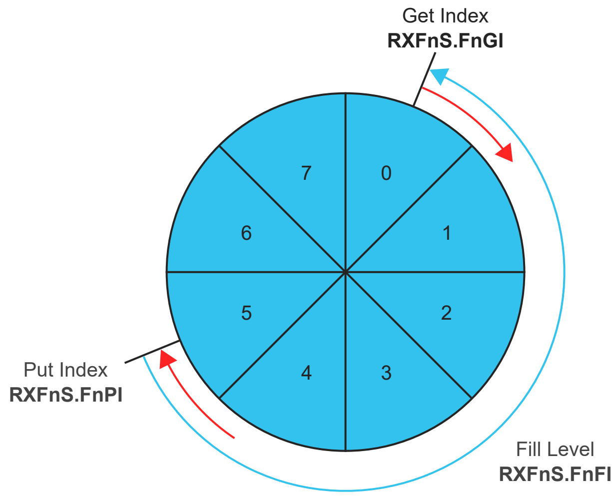
39.6.5.2. Rx FIFOs

Rx FIFO 0 and Rx FIFO 1 can be configured to hold up to 64 elements each. Configuration of the two Rx FIFOs is done via registers RXF0C and RXF1C.

Received messages that passed acceptance filtering are transferred to the Rx FIFO as configured by the matching filter element. For a description of the filter mechanisms available for Rx FIFO 0 and Rx FIFO 1 see *Acceptance Filtering*. The Rx FIFO element is described in *Rx Buffer and FIFO Element*.

To avoid an Rx FIFO overflow, the Rx FIFO watermark can be used. When the Rx FIFO fill level reaches the Rx FIFO watermark configured by RXFnC.FnWM, interrupt flag IR.RFnW is set. When the Rx FIFO Put Index reaches the Rx FIFO Get Index an Rx FIFO Full condition is signalled by RXFnS.FnF. In addition interrupt flag IR.RFnF is set, where n = 0 for Rx FIFO 0 buffer and n = 1 for Rx FIFO 1 buffer.

Figure 39-7. Rx FIFO Status (n =0 for Rx FIFO 0 buffer and n=1 for Rx FIFO 1 buffer)



When reading from an Rx FIFO, Rx FIFO Get Index RXFnS.FnGI • FIFO Element Size has to be added to the corresponding Rx FIFO start address RXFnC.FnSA.

Table 39-3. Rx Buffer / FIFO Element Size

RXESC.RBDS[2:0] RXESC.FnDS[2:0]	Data Field [bytes]	FIFO Element Size [RAM words]
000	8	4
001	12	5
010	16	6
011	20	7
100	24	8
101	32	10
110	48	14
111	64	18

Rx FIFO Blocking Mode

The Rx FIFO blocking mode is configured by RXFnC.FnOM = ‘0’. This is the default operation mode for the Rx FIFOs.

When an Rx FIFO full condition is reached (RXFnS.FnPI = RXFnS.FnGI), no further messages are written to the corresponding Rx FIFO until at least one message has been read out and the Rx FIFO Get Index has been incremented. An Rx FIFO full condition is signaled by RXFnS.FnF = ‘1’. In addition interrupt flag IR.RFnF is set.

In case a message is received while the corresponding Rx FIFO is full, this message is discarded and the message lost condition is signalled by RXFnS.RFnL = ‘1’. In addition interrupt flag IR.RFnL is set.

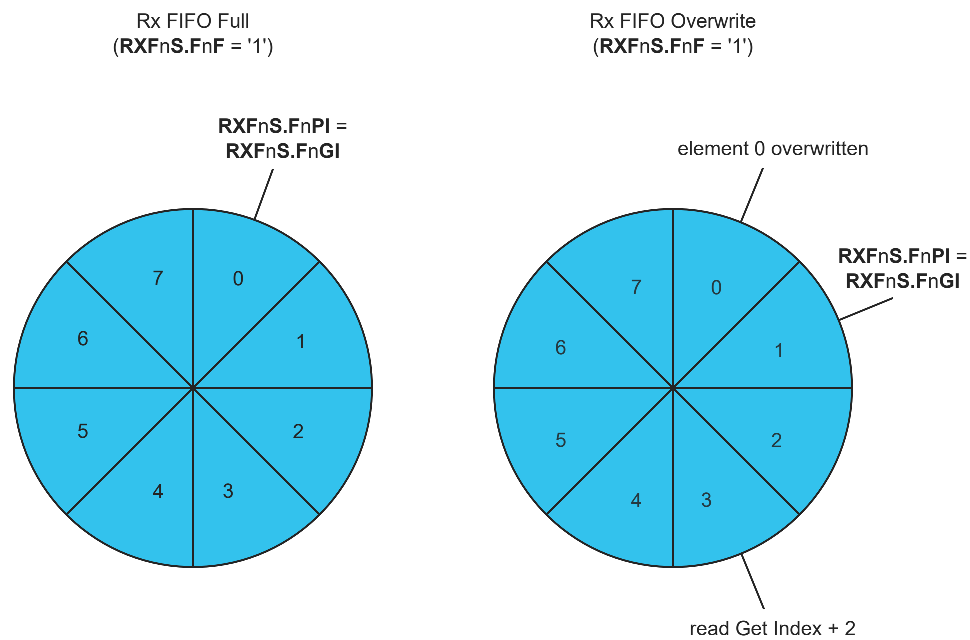
Rx FIFO Overwrite Mode

The Rx FIFO overwrite mode is configured by RXFnC.FnOM = ‘1’.

When an Rx FIFO full condition (RXFnS.FnPI = RXFnS.FnGI) is signaled by RXFnS.FnF = ‘1’, the next message accepted for the FIFO will overwrite the oldest FIFO message. Put and get index are both incremented by one.

When an Rx FIFO is operated in overwrite mode and an Rx FIFO full condition is signaled, reading of the Rx FIFO elements should start at least at get index + 1. The reason for that is, that it might happen, that a received message is written to the Message RAM (put index) while the CPU is reading from the Message RAM (get index). In this case inconsistent data may be read from the respective Rx FIFO element. Adding an offset to the get index when reading from the Rx FIFO avoids this problem. The offset depends on how fast the CPU accesses the Rx FIFO. The figure below shows an offset of two with respect to the get index when reading the Rx FIFO. In this case the two messages stored in element 1 and 2 are lost.

Figure 39-8. Rx FIFO Overflow Handling (n =0 for Rx FIFO 0 buffer and n=1 for Rx FIFO 1 buffer)



After reading from the Rx FIFO, the number of the last element read has to be written to the Rx FIFO Acknowledge Index RXFnA.FnA. This increments the get index to that element number. In case the put index has not been incremented to this Rx FIFO element, the Rx FIFO full condition is reset (RXFnS.FnF = '0').

39.6.5.3. Dedicated Rx Buffers

The CAN supports up to 64 dedicated Rx Buffers. The start address of the dedicated Rx Buffer section is configured via RXBC.RBSA bits (RXBC <15:0>).

For each Rx Buffer a Standard or Extended Message ID Filter Element with SFEC / EFEC = "111" and SFID2 / EFID2[10:9] = "00" has to be configured (see *Standard Message ID Filter Element* and *Extended Message ID Filter Element*).

After a received message has been accepted by a filter element, the message is stored into the Rx Buffer in the Message RAM referenced by the filter element. The format is the same as for an Rx FIFO element. In addition the flag IR.DRX bit (IR <19>) (Message stored in Dedicated Rx Buffer) in the interrupt register is set.

Table 39-4. Example Filter Configuration for Rx Buffers

Filter Element	SFID1[10:0] / EFID1[28:0]	SFID2[10:9] / EFID2[10:9]	SFID2[5:0] / EFID2[5:0]
0	ID message 1	00	00 0000
1	ID message 2	00	00 0001
2	ID message 3	00	00 0010

After the last word of a matching received message has been written to the Message RAM, the respective New Data flag in register NDAT1, NDAT2 is set. As long as the New Data flag is set, the respective Rx Buffer is locked against updates from received matching frames. The New Data flags have to be reset by the CPU by writing a ‘1’ to the respective bit position.

While an Rx Buffer’s New Data flag is set, a Message ID Filter Element referencing this specific Rx Buffer will not match, causing the acceptance filtering to continue. Following Message ID Filter Elements may cause the received message to be stored into another Rx Buffer, or into an Rx FIFO, or the message may be rejected, depending on filter configuration.

Rx Buffer Handling

- Reset interrupt flag IR.DRX bit (IR <19>)
- Read New Data registers
- Read messages from Message RAM
- Reset New Data flags of processed messages

39.6.5.4. Debug on CAN Support

Debug messages are stored into Rx Buffers. For debug handling three consecutive Rx buffers (e.g. #61, #62, #63) have to be used for storage of debug messages A, B, and C. The format is the same as for an Rx Buffer or an Rx FIFO element (see *Rx Buffer and FIFO Element*).

Advantage: Fixed start address for the DMA transfers (relative to RXBC.RBSA bits (RXBC <15:0>)), no additional configuration required.

For filtering of debug messages Standard / Extended Filter Elements with SFEC / EFEC = “111” have to be set up. Messages matching these filter elements are stored into the Rx Buffers addressed by SFID2 / EFID2[5:0].

After message C has been stored, the DMA request output is activated and the three messages can be read from the Message RAM under DMA control. The RAM words holding the debug messages will not be changed by the CAN while DMA request is activated. The behavior is similar to that of an Rx Buffers with its New Data flag set.

After the DMA has completed the DMA unit sets the DMA acknowledge. This resets DMA request. Now the CAN is prepared to receive the next set of debug messages.

Filtering for Debug Messages

Filtering for debug messages is done by configuring one Standard / Extended Message ID Filter Element for each of the three debug messages. To enable a filter element to filter for debug messages SFEC / EFEC has to be programmed to “111”. In this case fields SFID1 / SFID2 and EFID1 / EFID2 have a different meaning (see *Standard Message ID Filter Element* and *Extended Message ID Filter Element*). While SFID2 / EFID2[10:9] controls the debug message handling state machine, SFID2 / EFID2[5:0] controls the location for storage of a received debug message.

When a debug message is stored, neither the respective New Data flag nor IR.DRX bit (IR <19>) are set. The reception of debug messages can be monitored via RXF1S.DMS bits (RXF1S <31:30>).

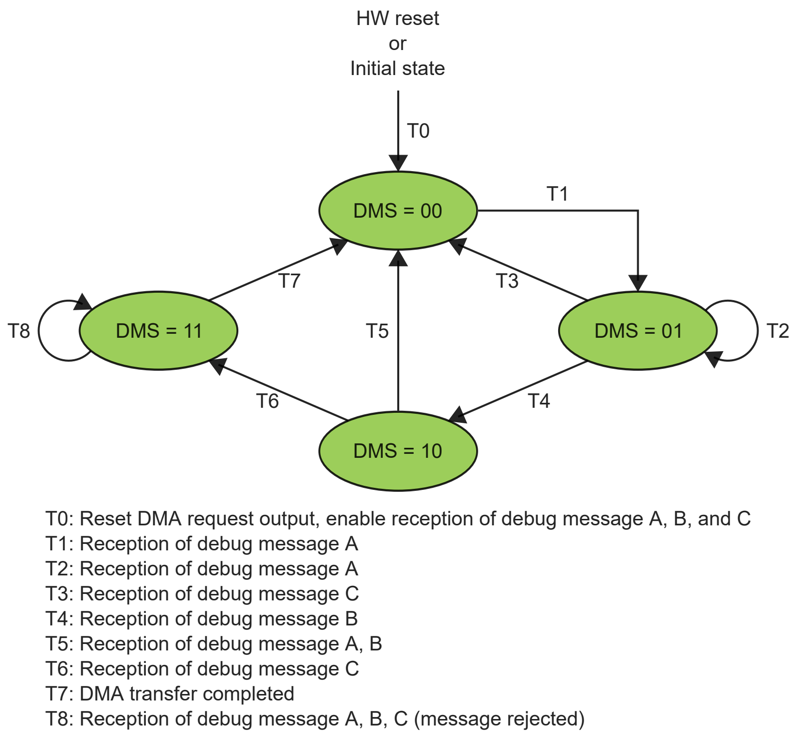
Table 39-5. Example Filter Configuration for Debug Messages

Filter Element	SFID1[10:0] / EFID1[28:0]	SFID2[10:9] / EFID2[10:9]	SFID2[5:0] / EFID2[5:0]
0	ID debug message A	01	11 1101
1	ID debug message B	10	11 1110
2	ID debug message C	11	11 1111

Debug Message Handling

The debug message handling state machine assures that debug messages are stored to three consecutive Rx Buffers in correct order. In case of missing messages the process is restarted. The DMA request is activated only when all three debug messages A, B, C have been received in correct order.

Figure 39-9. Debug Message Handling State Machine



39.6.6. Tx Handling

The Tx Handler handles transmission requests for the dedicated Tx Buffers, the Tx FIFO, and the Tx Queue. It controls the transfer of transmit messages to the CAN Core, the Put and Get Indices, and the Tx Event FIFO. Up to 32 Tx Buffers can be set up for message transmission. The CAN mode for transmission (Classic CAN or CAN FD) can be configured separately for each Tx Buffer element. The Tx Buffer element is described in *Tx Buffer Element*. The table below describes the possible configurations for frame transmission.

Table 39-6. Possible Configurations for Frame Transmission

CCCR		Tx Buffer Element		Frame Transmission
BRSE	FDOE	FDF	BRS	
ignored	0	ignored	ignored	Classic CAN
0	1	0	ignored	Classic CAN
0	1	1	ignored	FD without bit rate switching
1	1	0	ignored	Classic CAN
1	1	1	0	FD without bit rate switching
1	1	1	1	FD with bit rate switching

Note: AUTOSAR requires at least three Tx Queue Buffers and support of transmit cancellation

The Tx Handler starts a Tx scan to check for the highest priority pending Tx request (Tx Buffer with lowest Message ID) when the Tx Buffer Request Pending register TXBRP is updated, or when a transmission has been started.

39.6.6.1. Transmit Pause

The transmit pause feature is intended for use in CAN systems where the CAN message identifiers are (permanently) specified to specific values and cannot easily be changed. These message identifiers may have a higher CAN arbitration priority than other defined messages, while in a specific application their relative arbitration priority should be inverse. This may lead to a case where one ECU sends a burst of CAN messages that cause another ECU's CAN messages to be delayed because that other messages have a lower CAN arbitration priority.

If e.g. CAN ECU-1 has the transmit pause feature enabled and is requested by its application software to transmit four messages, it will, after the first successful message transmission, wait for two CAN bit times of bus idle before it is allowed to start the next requested message. If there are other ECUs with pending messages, those messages are started in the idle time, they would not need to arbitrate with the next message of ECU-1. After having received a message, ECU-1 is allowed to start its next transmission as soon as the received message releases the CAN bus.

The transmit pause feature is controlled by bit CCCR.TXP bit (CCCR<14>). If the bit is set, the CAN will, each time it has successfully transmitted a message, pause for two CAN bit times before starting the next transmission. This enables other CAN nodes in the network to transmit messages even if their messages have lower prior identifiers. Default is transmit pause disabled (CCCR.TXP bit (CCCR<14>) = '0').

This feature looses up burst transmissions coming from a single node and it protects against "babbling idiot" scenarios where the application program erroneously requests too many transmissions.

39.6.6.2. Dedicated Tx Buffers

Dedicated Tx Buffers are intended for message transmission under complete control of the CPU. Each Dedicated Tx Buffer is configured with a specific Message ID. In case that multiple Tx Buffers are configured with the same Message ID, the Tx Buffer with the lowest buffer number is transmitted first.

If the data section has been updated, a transmission is requested by an Add Request via TXBAR.ARn. The requested messages arbitrate internally with messages from an optional Tx FIFO or Tx Queue and externally with messages on the CAN bus, and are sent out according to their Message ID.

A Dedicated Tx Buffer allocates Element Size 32-bit words in the Message RAM (refer to table below). Therefore the start address of a dedicated Tx Buffer in the Message RAM is calculated by adding transmit buffer index (0...31) • Element Size to the Tx Buffer Start Address TXBC.TBSA bit (TXBC <15:0>).

Table 39-7. Tx Buffer / FIFO / Queue Element Size

TXESC.TBDS[2:0]	Data Field [bytes]	Element Size [RAM words]
000	8	4
001	12	5
010	16	6
011	20	7
100	24	8
101	32	10
110	48	14
111	64	18

39.6.6.3. Tx FIFO

Tx FIFO operation is configured by programming TXBC.TFQM bit (TXBC <30>) to '0'. Messages stored in the Tx FIFO are transmitted starting with the message referenced by the Get Index TXFQS.TFGI bit TXFQS (<12:8>). After each transmission the Get Index is incremented cyclically until the Tx FIFO is empty. The Tx FIFO enables transmission of messages with the same Message ID from different Tx Buffers in the order these messages have been written to the Tx FIFO. The CAN calculates the Tx FIFO Free Level TXFQS.TFFL bits (TXFQS <5:0>) as difference between Get and Put Index. It indicates the number of available (free) Tx FIFO elements.

New transmit messages have to be written to the Tx FIFO starting with the Tx Buffer referenced by the Put Index TXFQS.TFQPI bits TXFQS (<20:16>). An Add Request increments the Put Index to the next free Tx FIFO element. When the Put Index reaches the Get Index, Tx FIFO Full (TXFQS.TFQF bit (TXFQS <21>) = '1') is signaled. In this case no further messages should be written to the Tx FIFO until the next message has been transmitted and the Get Index has been incremented.

When a single message is added to the Tx FIFO, the transmission is requested by writing a '1' to the TXBAR bit related to the Tx Buffer referenced by the Tx FIFO's Put Index.

When multiple (n) messages are added to the Tx FIFO, they are written to n consecutive Tx Buffers starting with the Put Index. The transmissions are then requested via TXBAR. The Put Index is then cyclically incremented by n. The number of requested Tx buffers should not exceed the number of free Tx Buffers as indicated by the Tx FIFO Free Level.

When a transmission request for the Tx Buffer referenced by the Get Index is canceled, the Get Index is incremented to the next Tx Buffer with pending transmission request and the Tx FIFO Free Level is recalculated. When transmission cancellation is applied to any other Tx Buffer, the Get Index and the FIFO Free Level remain unchanged.

A Tx FIFO element allocates Element Size 32-bit words in the Message RAM (refer to *Tx Buffer / FIFO / Queue Element Size*). Therefore the start address of the next available (free) Tx FIFO Buffer is calculated by adding Tx FIFO/Queue Put Index TXFQS.TFQPI bits (TXFQS <20:16>) (0...31).

- Element Size to the Tx Buffer Start Address TXBC.TBSA bits (TXBC <15:0>)

39.6.6.4. Tx Queue

Tx Queue operation is configured by programming TXBC.TFQM bit (TXBC <30>) to '1'. Messages stored in the Tx Queue are transmitted starting with the message with the lowest Message ID (highest priority). In case that multiple Queue Buffers are configured with the same Message ID, the Queue Buffer with the lowest buffer number is transmitted first.

New messages have to be written to the Tx Buffer referenced by the Put Index TXFQS.TFQPI bits TXFQS (<20:16>). An Add Request cyclically increments the Put Index to the next free Tx Buffer. In case that the Tx Queue is full (TXFQS.TFQF bit (TXFQS <21>) = '1'), the Put Index is not valid and no further message should be written to the Tx Queue until at least one of the requested messages has been sent out or a pending transmission request has been canceled.

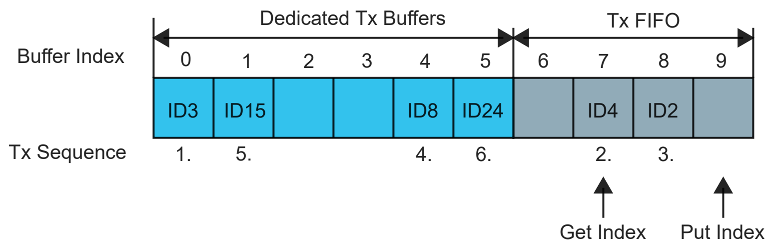
The application may use register TXBRP instead of the Put Index and may place messages to any Tx Buffer without pending transmission request.

A Tx Queue Buffer allocates Element Size 32-bit words in the Message RAM (refer to *Tx Buffer / FIFO / Queue Element Size*). Therefore the start address of the next available (free) Tx Queue Buffer is calculated by adding Tx FIFO/Queue Put Index TXFQS.TFQPI bits TXFQS (<20:16>) (0...31). Element Size to the Tx Buffer Start Address TXBC.TBSA bits (TXBC <15:0>).

39.6.6.5. Mixed Dedicated Tx Buffers / Tx FIFO

In this case the Tx Buffers section in the Message RAM is subdivided into a set of Dedicated Tx Buffers and a Tx FIFO. The number of Dedicated Tx Buffers is configured by TXBC.NDTB bits (TXBC <21:16>). The number of Tx Buffers assigned to the Tx FIFO is configured by TXBC.TFQS bits (TXBC <29:24>). In case TXBC.TFQS bits (TXBC <29:24>) is programmed to zero, only Dedicated Tx Buffers are used.

Figure 39-10. Example of mixed Configuration Dedicated Tx Buffers / Tx FIFO



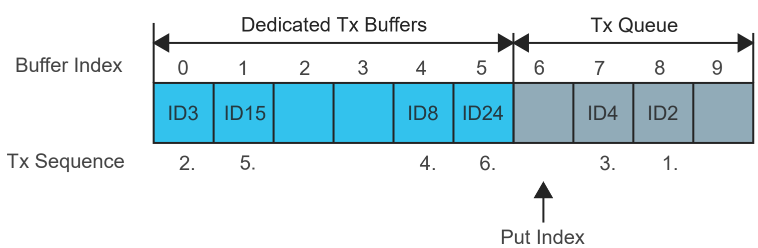
Tx prioritization:

- Scan Dedicated Tx Buffers and oldest pending Tx FIFO Buffer (referenced by TXFQS.TFGI bits (TXFQS <12:8>))
- Buffer with lowest Message ID gets highest priority and is transmitted next

39.6.6.6. Mixed Dedicated Tx Buffers / Tx Queue

In this case the Tx Buffers section in the Message RAM is subdivided into a set of Dedicated Tx Buffers and a Tx Queue. The number of Dedicated Tx Buffers is configured by TXBC.NDTB bits (TXBC <21:16>). The number of Tx Queue Buffers is configured by TXBC.TFQS bits (TXBC <29:24>). In case TXBC.TFQS bits (TXBC <29:24>) is programmed to zero, only Dedicated Tx Buffers are used.

Figure 39-11. Example of mixed Configuration Dedicated Tx Buffers / Tx Queue



Tx prioritization:

- Scan all Tx Buffers with activated transmission request
- Tx Buffer with lowest Message ID gets highest priority and is transmitted next

39.6.6.7. Transmit Cancellation

The CAN supports transmit cancellation. This feature is especially intended for gateway applications and AUTOSAR based applications. To cancel a requested transmission from a dedicated Tx Buffer or a Tx Queue Buffer the CPU has to write a ‘1’ to the corresponding bit position (=number of Tx Buffer) of register TXBCR. Transmit cancellation is not intended for Tx FIFO operation.

Successful cancellation is signaled by setting the corresponding bit of register TXBCF to ‘1’.

In case a transmit cancellation is requested while a transmission from a Tx Buffer is already ongoing, the corresponding TXBRP bit remains set as long as the transmission is in progress. If the transmission was successful, the corresponding TXBTO and TXBCF bits are set. If the transmission was not successful, it is not repeated and only the corresponding TXBCF bit is set.

Note: In case a pending transmission is canceled immediately before this transmission could have been started, there follows a short time window where no transmission is started even if another message is also pending in this node. This may enable another node to transmit a message which may have a lower priority than the second message in this node.

39.6.6.8. Tx Event Handling

To support Tx event handling the CAN has implemented a Tx Event FIFO. After the CAN has transmitted a message on the CAN bus, Message ID and timestamp are stored in a Tx Event FIFO element. To link a Tx event to a Tx Event FIFO element, the Message Marker from the transmitted Tx Buffer is copied into the Tx Event FIFO element.

The Tx Event FIFO can be configured to a maximum of 32 elements. The Tx Event FIFO element is described in *Tx Event FIFO Element*.

When a Tx Event FIFO full condition is signaled by IR.TEFF bit (IR <14>), no further elements are written to the Tx Event FIFO until at least one element has been read out and the Tx Event FIFO Get Index has been incremented. In case a Tx event occurs while the Tx Event FIFO is full, this event is discarded and interrupt flag IR.TEFL bit (IR <15>) is set.

To avoid a Tx Event FIFO overflow, the Tx Event FIFO watermark can be used. When the Tx Event FIFO fill level reaches the Tx Event FIFO watermark configured by TXEFC.EFWM bit (TXEFC <29:24>), interrupt flag IR.TEFW bit (IR <13>) is set.

When reading from the Tx Event FIFO, two times the Tx Event FIFO Get Index TXEFS.EFGI bit (TXEFS <12:8>) has to be added to the Tx Event FIFO start address TXEFC.EFSA bit (TXEFC <15:0>).

39.6.7. FIFO Acknowledge Handling

The Get Indexes of Rx FIFO 0, Rx FIFO 1 and the Tx Event FIFO are controlled by writing to the corresponding FIFO Acknowledge Index (refer to *RXF0A*, *RXF1A* and *TXEFA*). Writing to the FIFO Acknowledge Index will set the FIFO Get Index to the FIFO Acknowledge Index plus one and thereby updates the FIFO Fill Level. There are two use cases:

When only a single element has been read from the FIFO (the one being pointed to by the Get Index), this Get Index value is written to the FIFO Acknowledge Index.

When a sequence of elements has been read from the FIFO, it is sufficient to write the FIFO Acknowledge Index only once at the end of that read sequence (value: Index of the last element read), to update the FIFO's Get Index.

Due to the fact that the CPU has free access to the CAN's Message RAM, special care has to be taken when reading FIFO elements in an arbitrary order (Get Index not considered). This might be useful when reading a High Priority Message from one of the two Rx FIFOs. In this case the FIFO's Acknowledge Index should not be written because this would set the Get Index to a wrong position and also alters the FIFO's Fill Level. In this case some of the older FIFO elements would be lost.

Note: The application has to ensure that a valid value is written to the FIFO Acknowledge Index. The CAN does not check for erroneous values.

39.6.8. Message RAM

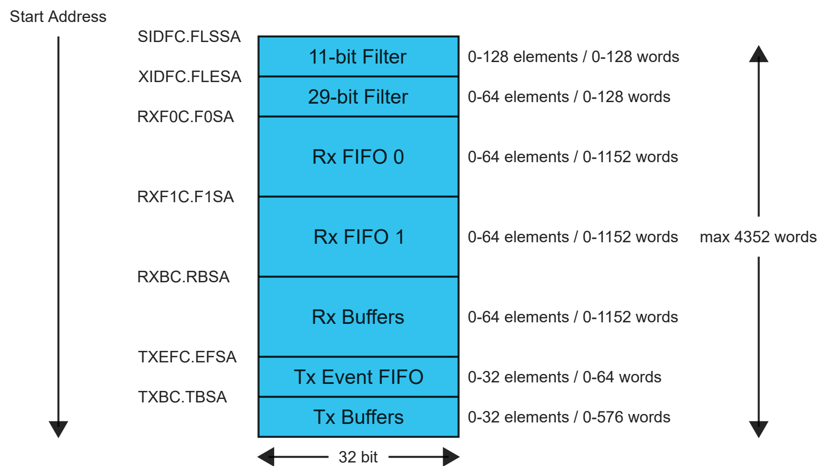
For storage of Rx/Tx messages and for storage of the filter configuration this module uses system RAM and the base address can be set by MRCFG.OFFSET bits (MRCFG <23:16>).

39.6.8.1. Message RAM Configuration

The Message RAM has a width of 32 bits. The CAN module can be configured to allocate up to 4352 words in the Message RAM. It is not necessary to configure each of the sections listed in the figure below, nor is there any restriction with respect to the sequence of the sections.

When operated in CAN FD mode the required Message RAM size strongly depends on the element size configured for Rx FIFO 0, Rx FIFO 1, Rx Buffers, and Tx Buffers via RXESC.F0DS bits (RXESC <2:0>), RXESC.F1DS bits (RXESC <6:4>), RXESC.RBDS bits (RXESC <10:8>), and TXESC.TBDS bits (TXESC <2:0>).

Figure 39-12. Message RAM Configuration



When the CAN addresses the Message RAM it addresses 32-bit words, not single bytes. The configurable start addresses are 32-bit word addresses (i.e. only bits 15 to 2 are evaluated and the two LSBs are ignored).

WARNING The CAN does not check for erroneous configuration of the Message RAM. Especially the configuration of the start addresses of the different sections and the number of elements of each section has to be done carefully to avoid falsification or loss of data.

39.6.8.2. Rx Buffer and FIFO Element

Up to 64 Rx Buffers and two Rx FIFOs can be configured in the Message RAM. Each Rx FIFO section can be configured to store up to 64 received messages. The structure of a Rx Buffer / FIFO element is shown in the table below. The element size can be configured for storage of CAN FD messages with up to 64 bytes data field via register RXESC.

Table 39-8. Rx Buffer and FIFO Element

<table>
<tr><th></th><th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td>R0</td><td>ESI</td><td>XTD</td><td>RTR</td><td colspan="29">ID[28:0]</td></tr>
<tr><td>R1</td><td>ANMF</td><td colspan="7">FIDX[6:0]</td><td></td><td></td><td>FDF</td><td>BRS</td><td colspan="4">DLC[3:0]</td><td colspan="16">RXTS[15:0]</td></tr>
<tr><td>R2</td><td colspan="8">DB3[7:0]</td><td colspan="8">DB2[7:0]</td><td colspan="8">DB1[7:0]</td><td colspan="8">DB0[7:0]</td></tr>
<tr><td>R3</td><td colspan="8">DB7[7:0]</td><td colspan="8">DB6[7:0]</td><td colspan="8">DB5[7:0]</td><td colspan="8">DB4[7:0]</td></tr>
<tr><td>...</td><td colspan="8">...</td><td colspan="8">...</td><td colspan="8">...</td><td colspan="8">...</td></tr>
<tr><td>Rn</td><td colspan="8">DBm[7:0]</td><td colspan="8">DBm-1[7:0]</td><td colspan="8">DBm-2[7:0]</td><td colspan="8">DBm-3[7:0]</td></tr>
</table>

R0 Bit 31 - ESI: Error State Indicator

0 : Transmitting node is error active.

1 : Transmitting node is error passive.

R0 Bit 30 - XTD: Extended Identifier

Signals to the Host whether the received frame has a standard or extended identifier.

0 : 11-bit standard identifier.

1 : 29-bit extended identifier.

R0 Bit 29 - RTR: Remote Transmission Request

Signals to the Host whether the received frame is a data frame or a remote frame.

0 : Received frame is a data frame.

1 : Received frame is a remote frame.

Note: There are no remote frames in CAN FD format. In case a CAN FD frame was received (EDL = '1'), bit RTR reflects the state of the reserved bit r1.

R0 Bits 28:0 - ID[28:0]: Identifier

Standard or extended identifier depending on bit XTD. A standard identifier is stored into ID[28:18].

R1 Bit 31 - ANMF: Accepted Non-matching Frame

Acceptance of non-matching frames may be enabled via GFC.ANFS bits (GFC <5:4>) and GFC.ANFE bits (GFC <3:2>).

0 : Received frame matching filter index FIDX.

1 : Received frame did not match any Rx filter element.

R1 Bits 30:24 - FIDX[6:0]: Filter Index

0-127 : Index of matching Rx acceptance filter element (invalid if ANMF = '1').

Note: Range is 0 to SIDFC.LSS bits (SIDFC <23:16>) -1 for standard and 0 to XIDFC.LSE bits (XIDFC <22:16>) -1 for extended.

R1 Bits 23:22 - Reserved

R1 Bit 21 - FDF: FD Format

0 : Standard frame format.

1 : CAN FD frame format (new DLC-coding and CRC).

R1 Bit 20 - BRS: Bit Rate Search

0 : Frame received without bit rate switching.

1 : Frame received with bit rate switching.

R1 Bits 19:16 - DLC[3:0]: Data Length Code

0-8 : CAN + CAN FD: received frame has 0-8 data bytes.

9-15 : CAN: received frame has 8 data bytes.

9-15 : CAN FD: received frame has 12/16/20/24/32/48/64 data bytes.

R1 Bits 15:0 - RXTS[15:0]: Rx Timestamp

Timestamp Counter value captured on start of frame reception. Resolution depending on configuration of the Timestamp Counter Prescaler TSCC.TCP bit (TSCC <19:16>).

R2 Bits 31:24 - DB3[7:0]: Data Byte 3

R2 Bits 23:16 - DB2[7:0]: Data Byte 2

R2 Bits 15:8 - DB1[7:0]: Data Byte 1

R2 Bits 7:0 - DB0[7:0]: Data Byte 0

R3 Bits 31:24 - DB7[7:0]: Data Byte 7

R3 Bits 23:16 - DB6[7:0]: Data Byte 6

R3 Bits 15:8 - DB5[7:0]: Data Byte 5

R3 Bits 7:0 - DB4[7:0]: Data Byte 4

...

Rn Bits 31:24 - DBm[7:0]: Data Byte m

Rn Bits 23:16 - DBm-1[7:0]: Data Byte m-1

Rn Bits 15:8 - DBm-2[7:0]: Data Byte m-2

Rn Bits 7:0 - DBm-3[7:0]: Data Byte m-3

WARNING Depending on the configuration of RXESC, between two and sixteen 32-bit words (Rn = 3 … 17) are used for storage of a CAN message's data field.

39.6.8.3. Tx Buffer Element

The Tx Buffers section can be configured to hold dedicated Tx Buffers as well as a Tx FIFO / Tx Queue. In case that the Tx Buffers section is shared by dedicated Tx buffers and a Tx FIFO / Tx Queue, the dedicated Tx Buffers start at the beginning of the Tx Buffers section followed by the buffers assigned to the Tx FIFO or Tx Queue. The Tx Handler distinguishes between dedicated Tx Buffers and Tx FIFO / Tx Queue by evaluating the Tx Buffer configuration TXBC.TFQS bits (TXBC <29:24>) and TXBC.NDTB bits (TXBC <21:16>). The element size can be configured for storage of CAN FD messages with up to 64 bytes data field via register TXESC.

Table 39-9. Tx Buffer Element

<table>
<tr><th></th><th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td>T0</td><td>ESI</td><td>XTD</td><td>RTR</td><td colspan="29">ID[28:0]</td></tr>
<tr><td>T1</td><td colspan="8">MM[7:0]</td><td>EFC</td><td></td><td>FDF</td><td>BRS</td><td colspan="4">DLC[3:0]</td><td colspan="16"></td></tr>
<tr><td>T2</td><td colspan="8">DB3[7:0]</td><td colspan="8">DB2[7:0]</td><td colspan="8">DB1[7:0]</td><td colspan="8">DB0[7:0]</td></tr>
<tr><td>T3</td><td colspan="8">DB7[7:0]</td><td colspan="8">DB6[7:0]</td><td colspan="8">DB5[7:0]</td><td colspan="8">DB4[7:0]</td></tr>
<tr><td>...</td><td colspan="8">...</td><td colspan="8">...</td><td colspan="8">...</td><td colspan="8">...</td></tr>
<tr><td>Tn</td><td colspan="8">DBm[7:0]</td><td colspan="8">DBm-1[7:0]</td><td colspan="8">DBm-2[7:0]</td><td colspan="8">DBm-3[7:0]</td></tr>
</table>

T0 Bit 31 - ESI: Error State Indicator

0 : ESI bit in CAN FD format depends only on error passive flag.

1 : ESI bit in CAN FD format transmitted recessive.

Note: The ESI bit of the transmit buffer is OR'ed with the error passive flag to decide the value of the ESI bit in the transmitted FD frame. As required by the CAN FD protocol specification, an error active node may optionally transmit the ESI bit recessive, but an error passive node will always transmit the ESI bit recessive.

T0 Bit 30 - XTD: Extended Identifier

0 : 11-bit standard identifier.

1 : 29-bit extended identifier.

T0 Bit 29 - RTR: Remote Transmission Request

0 : Transmit data frame.

1 : Transmit remote frame.

Note: When RTR = ‘1’, the CAN transmits a remote frame according to ISO 11898-1, even if CCCR.CME enables the transmission in CAN FD format.

T0 Bits 28:0 - ID[28:0]: Identifier

Standard or extended identifier depending on bit XTD. A standard identifier is stored into ID[28:18].

T1 Bits 31:24 - MM[7:0]: Message Marker

Written by CPU during Tx Buffer configuration. Copied into Tx Event FIFO element for identification of Tx message status.

T1 Bit 23 - EFC: Event FIFO Control

0 : Don’t store Tx events.

1 : Store Tx events.

T1 Bit 22 - Reserved

TR1 Bit 21 - FDF: FD Format

0 : Frame transmitted in Classic CAN format.

1 : Frame transmitted in CAN FD format.

T1 Bit 20 - BRS: Bit Rate Search

0 : CAN FD frames transmitted without bit rate switching.

1 : CAN FD frames transmitted with bit rate switching.

Note: Bits ESI, FDF, and BRS are only evaluated when CAN FD operation is enabled CCCR.FDOE bit (CCCR <8>) = ‘1’. Bit BRS is only evaluated when in addition CCCR.BRSE bit (CCCR <9>) = ‘1’.

T1 Bits 19:16 - DLC[3:0]: Data Length Code

0-8 : CAN + CAN FD: received frame has 0-8 data bytes.

9-15 : CAN: received frame has 8 data bytes.

9-15 : CAN FD: received frame has 12/16/20/24/32/48/64 data bytes.

T1 Bits 15:0 - Reserved

T2 Bits 31:24 - DB3[7:0]: Data Byte 3

T2 Bits 23:16 - DB2[7:0]: Data Byte 2

T2 Bits 15:8 - DB1[7:0]: Data Byte 1

T2 Bits 7:0 - DB0[7:0]: Data Byte 0

T3 Bits 31:24 - DB7[7:0]: Data Byte 7

T3 Bits 23:16 - DB6[7:0]: Data Byte 6

T3 Bits 15:8 - DB5[7:0]: Data Byte 5

T3 Bits 7:0 - DB4[7:0]: Data Byte 4

...

Tn Bits 31:24 - DBm[7:0]: Data Byte m

Tn Bits 23:16 - DBm-1[7:0]: Data Byte m-1

Tn Bits 15:8 - DBm-2[7:0]: Data Byte m-2

Tn Bits 7:0 - DBm-3[7:0]: Data Byte m-3

Note: Depending on the configuration of TXESC, between two and sixteen 32-bit words (Tn = 3 ... 17) are used for storage of a CAN message’s data field.

39.6.8.4. Tx Event FIFO Element

Each element stores information about transmitted messages. By reading the Tx Event FIFO the Host CPU gets this information in the order the messages were transmitted. Status information about the Tx Event FIFO can be obtained from register TXEFS.

Table 39-10. Tx Event FIFO Element

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E0	ESI	XTD	RTR	ID[28:0]																												
E1	MM[7:0]								ET[1:0]		FDF	BRS	DLC[3:0]				TXTS[15:0]															

E0 Bit 31 - ESI: Error State Indicator

0 : Transmitting node is error active.

1 : Transmitting node is error passive.

E0 Bit 30 - XTD: Extended Identifier

0 : 11-bit standard identifier.

1 : 29-bit extended identifier.

E0 Bit 29 - RTR: Remote Transmission Request

0 : Received frame is a data frame.

1 : Received frame is a remote frame.

E0 Bits 28:0 - ID[28:0]: Identifier

Standard or extended identifier depending on bit XTD. A standard identifier is stored into ID[28:18].

E1 Bits 31:24 - MM[7:0]: Message Marker

Copied from Tx Buffer into Tx Event FIFO element for identification of Tx message status.

E1 Bits 23:22 - ET[1:0]: Event Type

This field defines the event type.

Table 39-11. Event Type

Value	Name	Description
0x0 or 0x3	RES	Reserved
0x1	TXE	Tx event
0x2	TXC	Transmission in spite of cancellation (always set for transmission in DAR mode)

E1 Bit 21 - FDF: FD Format

0 : Standard frame format.

1 : CAN FD frame format (new DLC-coding and CRC).

E1 Bit 20 - BRS: Bit Rate Search

0 : Frame received without bit rate switching.

1 : Frame received with bit rate switching.

E1 Bits 19:16 - DLC[3:0]: Data Length Code

0-8 : CAN + CAN FD: received frame has 0-8 data bytes.

9-15 : CAN: received frame has 8 data bytes.

9-15 : CAN FD: received frame has 12/16/20/24/32/48/64 data bytes.

E1 Bits 15:0 - TXTS[15:0]: Tx Timestamp

Timestamp Counter value captured on start of frame transmission. Resolution depending on configuration of the Timestamp Counter Prescaler TSCC.TCP bit (TSCC <19:16>).

39.6.8.5. Standard Message ID Filter Element

Up to 128 filter elements can be configured for 11-bit standard IDs. When accessing a Standard Message ID Filter element, its address is the Filter List Standard Start Address SIDFC.FLSSA bits (SIDFC <15:0>) plus the index of the filter element (0 … 127).

Table 39-12. Standard Message ID Filter Element

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
S0	SFT[1:0]		SFEC[2:0]			SFID1[10:0]																SFID2[10:0]										

Bits 31:30 - SFT[1:0]: Standard Filter Type

This field defines the standard filter type.

Table 39-13. Standard Filter Type

Value	Name	Description
0x0	RANGE	Range filter from SFID1 to SFID2 (SFID2 >= SFID1)
0x1	DUAL	Dual ID filter for SFID1 or SFID2
0x2	CLASSIC	Classic filter: SFID1 = filter, SFID2 = mask
0x3	RES	Reserved

Bits 29:27 - SFEC[2:0]: Standard Filter Element Configuration

All enabled filter elements are used for acceptance filtering of standard frames. Acceptance filtering stops at the first matching enabled filter element or when the end of the filter list is reached. If SFEC = “100”, “101”, or “110” a match sets interrupt flag IR.HPM (IR <8>) and, if enabled, an interrupt is generated. In this case register HPMS is updated with the status of the priority match.

Table 39-14. Standard Filter Element Configuration

Value	Name	Description
0x0	DISABLE	Disable filter element
0x1	STF0M	Store in Rx FIFO 0 if filter matches
0x2	STF1M	Store in Rx FIFO 1 if filter matches
0x3	REJECT	Reject ID if filter matches
0x4	PRIORITY	Set priority if filter matches.
0x5	PRIF0M	Set priority and store in FIFO 0 if filter matches.
0x6	PRIF1M	Set priority and store in FIFO 1 if filter matches.
0x7	STRXBUF	Store into Rx Buffer or as debug message, configuration of SFT[1:0] ignored.

Bits 26:16 - SFID1[10:0]: Standard Filter ID 1

First ID of standard ID filter element.

When filtering for Rx Buffers or for debug messages this field defines the ID of a standard mesage to be stored. The received identifiers must match exactly, no masking mechanism is used.

Bits 15:11 - Reserved

Bits 10:0 - SFID2[10:0]: Standard Filter ID 2

This bit field has a different meaning depending on the configuration of SFEC.

1. SFEC = “001” ... “110”: Second ID of standard ID filter element.
2. SFEC = “111”: Filter for Rx Buffers or for debug messages.

SFID2[10:9] decides whether the received message is stored into an Rx Buffer or treated as message A, B, or C of the debug message sequence.
00 = Store message into an Rx Buffer
01 = Debug Message A
10 = Debug Message B
11 = Debug Message C

SFID2[8:6] is used to control the filter event pins at the Extension Interface. A ‘1’ at the respective bit position enables generation of a pulse at the related filter event pin with the duration of one CLK_CANx_APB period in case the filter matches.

SFID2[5:0] defines the offset to the Rx Buffer Start Address RXBC.RBSA bits (RXBC <15:0>) for storage of a matching message.

39.6.8.6. Extended Message ID Filter Element

Up to 64 filter elements can be configured for 29-bit extended IDs. When accessing an Extended Message ID Filter element, its address is the Filter List Extended Start Address XIDFC.FLESA bits (XIDFC <15:0>) plus two times the index of the filter element (0…63).

Table 39-15. Extended Message ID Filter Element

<table>
<tr><th></th><th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td>F0</td><td colspan="3">EFEC [2:0]</td><td colspan="29">EFID1[28:0]</td></tr>
<tr><td>F1</td><td colspan="2">EFT [1:0]</td><td></td><td colspan="29">EFID2[28:0]</td></tr>
</table>

F0 Bits 31:29 - EFEC[2:0]: Extended Filter Element Configuration

All enabled filter elements are used for acceptance filtering of extended frames. Acceptance filtering stops at the first matching enabled filter element or when the end of the filter list is reached. If EFEC = “100”, “101”, or “110” a match sets interrupt flag IR.HPM (IR <8>) and, if enabled, an interrupt is generated. In this case register HPMS is updated with the status of the priority match.

Table 39-16. Extended Filter Element Configuration

Value	Name	Description
0x0	DISABLE	Disable filter element.
0x1	STF0M	Store in Rx FIFO 0 if filter matches.
0x2	STF1M	Store in Rx FIFO 1 if filter matches.
0x3	REJECT	Reject ID if filter matches.
0x4	PRIORITY	Set priority if filter matches.
0x5	PRIF0M	Set priority and store in FIFO 0 if filter matches.
0x6	PRIF1M	Set priority and store in FIFO 1 if filter matches.
0x7	STRXBUF	Store into Rx Buffer or as debug message, configuration of EFT[1:0] ignored.

F0 Bits 28:0 - EFID1[28:0]: Extended Filter ID 1

First ID of extended ID filter element.

When filtering for Rx Buffers or for debug messages this field defines the ID of a extended message to be stored. The received identifiers must match exactly, only XIDAM masking mechanism is used.

F1 Bits 31:30 - EFT[1:0]: Extended Filter Type

This field defines the extended filter type.

Table 39-17. Extended Filter Type

Value	Name	Description
0x0	RANGEM	Range filter from EFID1 to EFID2 (EFID2 >= EFID1).
0x1	DUAL	Dual ID filter for EFID1 or EFID2.
0x2	CLASSIC	Classic filter: EFID1 = filter, EFID2 = mask.
0x3	RANGE	Range filter from EFID1 to EFID2 (EFID2 >= EFID1), XIDAM mask not applied.

F1 Bits 28:0 - EFID2[28:0]: Extended Filter ID 2

This bit field has a different meaning depending on the configuration of EFEC.
1) EFEC = “001” ... “110” Second ID of standard ID filter element.
2) EFEC = “111” Filter for Rx Buffers or for debug messages.

EFID2[10:9] decides whether the received message is stored into an Rx Buffer or treated as message A, B, or C of the debug message sequence.

00 = Store message into an Rx Buffer

01 = Debug Message A

10 = Debug Message B

11 = Debug Message C

EFID2[8:6] is used to control the filter event pins at the Extension Interface. A ‘1’ at the respective bit position enables generation of a pulse at the related filter event pin with the duration of one CLK_CANx_AHB period in case the filter matches.

EFID2[5:0] defines the offset to the Rx Buffer Start Address RXBC.RBSA bits (RXBC <15:0>) for storage of a matching message.

39.6.9. Interrupts

The CAN has the following interrupt sources:

- Access to Reserved Address
- Protocol Errors (Data Phase / Arbitration Phase)
- Watchdog Interrupt
- “bus off” Status
- Error Warning & Passive
- Error Logging Overflow
- AHB Bus Error (BERR)
- Message stored to Dedicated Rx Buffer
- Timeout Occurred
- Message RAM Access Failure
- Timestamp Wraparound
- Tx Event FIFO statuses (Element Lost / Full / Watermark Reached / New Entry)
- Tx FIFO Empty

- Transmission Cancellation Finished
- Timestamp Completed
- High Priority Message
- Rx FIFO 1 Statuses (Message Lost / Full / Watermark Reached / New Message)
- Rx FIFO 0 Statuses (Message Lost / Full / Watermark Reached / New Message)

Each interrupt source has an interrupt flag associated with it. The interrupt flag register (IR) is set when the interrupt condition occurs. Each interrupt can be individually enabled by writing '1' or disabled by writing '0' to the corresponding bit in the interrupt enable register (IE). Each interrupt flag can be assigned to one of two interrupt service lines.

An interrupt request is generated when an interrupt flag is set, the corresponding interrupt enable is set, and the corresponding service line enable assigned to the interrupt is set. The interrupt request remains active until the interrupt flag is cleared, the interrupt is disabled, the service line is disabled, or the CAN is reset. Refer to the IR register for details on how to clear interrupt flags. The interrupt request lines are connected to the Nested Vector Interrupt Controller (NVIC). All interrupt requests from the peripheral are sent to the *Nested vector Interrupt Controller NVIC*. Using the CAN FD interrupt requires the interrupt controller to be configured first. The user must read the IR register to determine which interrupt condition is present.

The CAN has one non-maskable interrupt source:

- AHB Bus Error (BERR)

The BERR bit (ERROR<0>) in the error interrupt flag register (ERROR) is set when the interrupt condition occurs. The bus error (BERR) interrupt flag is assigned to the ERROR interrupt service line.

The error interrupt request is generated when the BERR interrupt flag is set. The interrupt request remains active until the interrupt flag is cleared, or the CAN is reset. The interrupt request is sent to the NVIC.

Note: Interrupts must be globally enabled for interrupt requests to be generated.

39.6.10. Sleep Mode Operation

The CAN module can be configured to operate in any idle sleep mode if both AHB and GCLK clocks are available. The CAN interrupts can be used to wake up the device from sleep mode.

The CAN module has its own low power mode that may be used at any time without disabling this module. This is performed by writing one to the Clock Stop Request bit in the CC Control register (CCCR.CSR bit (CCCR <4>) = 1). Once all pending transactions are completed and the idle bus state is detected, the CAN will automatically set the Clock Stop Acknowledge bit (CCCR.CSA bit (CCCR <3>) = 1). The CAN then reverts back to its initial state (CCCR.INIT bit (CCCR <0>) = 1), blocking further transfers.

To exit low power mode, CCCR.CSR bit (CCCR<4>) in CCCR register must be written to 0. Afterwards, the application can restart CAN communication by resetting bit CCCR.INIT bit (CCCR <0>).

After reset, the GCLK_CANx and CLK_CANx_AHB clocks are not requested, except for each APB bus access. However, after the CAN initialization, both GCLK_CANx and CLK_CANx_AHB clocks are requested as long as Clock Stop Request bit in the CC Control register is cleared (CCCR.CSR (CCCR<4>)= 0), and stopped when Clock Stop Request bit in the CC Control register is set (CCCR.CSR (CCCR<4>) = 1) and the Clock Stop Acknowledge bit is set (CCCR.CSA (CCCR<3>)= 1). To limit the wake-up time latency, the CAN clock sources must be enabled in continuous mode (ONDEMAND of respective oscillator must be set zero). For further details, refer to the Clock System chapter.

39.6.11. Synchronization

Due to the asynchronicity between the main clock domain (CLK_CANx_AHB) and the peripheral clock domain (GCLK_CANx) some registers are synchronized when written. When a write-synchronized register is written, the read back value will not be updated until the register has completed synchronization.

The following bits and registers are write-synchronized:

One Initialization bit in CC Control register (CCCR.INIT bit (CCCR <0>)).

39.7. Register Summary

For descriptions and definitions of both Register and bitfield properties, refer to Register Properties.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00 ... 0x03	Reserved									
0x04	ENDN	31:24	ETV[31:24]							
		23:16	ETV[23:16]							
		15:8	ETV[15:8]							
		7:0	ETV[7:0]							
0x08	MRCFG	31:24								
		23:16	OFFSET[7:0]							
		15:8								
		7:0								
0x0C	DBTP	31:24								
		23:16	TDC			DBRP[4:0]				
		15:8				DTSEG1[4:0]				
		7:0	DTSEG2[3:0]				DSJW[3:0]			
0x10	TEST	31:24								
		23:16								
		15:8								
		7:0	RX	TX[1:0]		LBCK				
0x14	RWD	31:24								
		23:16								
		15:8	WDV[7:0]							
		7:0	WDC[7:0]							
0x18	CCCR	31:24								
		23:16								
		15:8		TXP	EFBI	PXHD			BRSE	FDOE
		7:0	TEST	DAR	MON	CSR	CSA	ASM	CCE	INIT
0x1C	NBTP	31:24	NSJW[6:0]							NBRP[8]
		23:16	NBRP[7:0]							
		15:8	NTSEG1[7:0]							
		7:0		NTSEG2[6:0]						
0x20	TSCC	31:24								
		23:16					TCP[3:0]			
		15:8								
		7:0							TSS[1:0]	
0x24	TSCV	31:24								
		23:16								
		15:8	TSC[15:8]							
		7:0	TSC[7:0]							
0x28	TOCC	31:24	TOP[15:8]							
		23:16	TOP[7:0]							
		15:8								
		7:0						TOS[1:0]		ETOC
0x2C	TOCV	31:24								
		23:16								
		15:8	TOC[15:8]							
		7:0	TOC[7:0]							
0x30 ... 0x3F	Reserved									
0x40	ECR	31:24								
		23:16	CEL[7:0]							
		15:8	RP	REC[6:0]						
		7:0	TEC[7:0]							
0x44	PSR	31:24								
		23:16		TDCV[6:0]						
		15:8		PXE	RFDF	RBRS	RESI	DLEC[2:0]		
		7:0	BO	EW	EP	ACT[1:0]		LEC[2:0]		

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x48	TDCR	31:24								
		23:16								
		15:8		TDCO[6:0]						
		7:0		TDCF[6:0]						
0x4C ... 0x4F	Reserved									
0x50	IR	31:24			ARA	PED	PEA	WDI	BO	EW
		23:16	EP	ELO			DRX	TOO	MRAF	TSW
		15:8	TEFL	TEFF	TEFW	TEFN	TFE	TCF	TC	HPM
		7:0	RF1L	RF1F	RF1W	RF1N	RF0L	RF0F	RF0W	RF0N
0x54	IE	31:24			ARAE	PEDE	PEAE	WDIE	BOE	EWE
		23:16	EPE	ELOE			DRXE	TOOE	MRAFE	TSWE
		15:8	TEFLE	TEFFE	TEFWE	TEFNE	TFEE	TCFE	TCE	HPME
		7:0	RF1LE	RF1FE	RF1WE	RF1NE	RF0LE	RF0FE	RF0WE	RF0NE
0x58	ILS	31:24			ARAL	PEDL	PEAL	WDIL	BOL	EWL
		23:16	EPL	ELOL			DRXL	TOOL	MRAFL	TSWL
		15:8	TEFLL	TEFFL	TEFWL	TEFNL	TFEL	TCFL	TCL	HPML
		7:0	RF1LL	RF1FL	RF1WL	RF1NL	RF0LL	RF0FL	RF0WL	RF0NL
0x5C	ILE	31:24								
		23:16								
		15:8								
		7:0							EINT1	EINT0
0x60 ... 0x7F	Reserved									
0x80	GFC	31:24								
		23:16								
		15:8								
		7:0			ANFS[1:0]		ANFE[1:0]		RRFS	RRFE
0x84	SIDFC	31:24								
		23:16	LSS[7:0]							
		15:8	FLSSA[15:8]							
		7:0	FLSSA[7:0]							
0x88	XIDFC	31:24								
		23:16		LSE[6:0]						
		15:8	FLESA[15:8]							
		7:0	FLESA[7:0]							
0x8C ... 0x8F	Reserved									
0x90	XIDAM	31:24				EIDM[28:24]				
		23:16	EIDM[23:16]							
		15:8	EIDM[15:8]							
		7:0	EIDM[7:0]							
0x94	HPMS	31:24								
		23:16								
		15:8	FLST	FIDX[6:0]						
		7:0	MSI[1:0]		BIDX[5:0]					
0x98	NDAT1	31:24	ND31	ND30	ND29	ND28	ND27	ND26	ND25	ND24
		23:16	ND23	ND22	ND21	ND20	ND19	ND18	ND17	ND16
		15:8	ND15	ND14	ND13	ND12	ND11	ND10	ND9	ND8
		7:0	ND7	ND6	ND5	ND4	ND3	ND2	ND1	ND0
0x9C	NDAT2	31:24	ND63	ND62	ND61	ND60	ND59	ND58	ND57	ND56
		23:16	ND55	ND54	ND53	ND52	ND51	ND50	ND49	ND48
		15:8	ND47	ND46	ND45	ND44	ND43	ND42	ND41	ND40
		7:0	ND39	ND38	ND37	ND36	ND35	ND34	ND33	ND32

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0xA0	RXF0C	31:24	F0OM	F0WM[6:0]						
		23:16		F0S[6:0]						
		15:8	F0SA[15:8]							
		7:0	F0SA[7:0]							
0xA4	RXF0S	31:24							RF0L	F0F
		23:16			F0PI[5:0]					
		15:8			F0GI[5:0]					
		7:0		F0FL[6:0]						
0xA8	RXF0A	31:24								
		23:16								
		15:8								
		7:0			F0AI[5:0]					
0xAC	RXBC	31:24								
		23:16								
		15:8	RBSA[15:8]							
		7:0	RBSA[7:0]							
0xB0	RXF1C	31:24	F1OM	F1WM[6:0]						
		23:16		F1S[6:0]						
		15:8	F1SA[15:8]							
		7:0	F1SA[7:0]							
0xB4	RXF1S	31:24	DMS[1:0]						RF1L	F1F
		23:16			F1PI[5:0]					
		15:8			F1GI[5:0]					
		7:0		F1FL[6:0]						
0xB8	RXF1A	31:24								
		23:16								
		15:8								
		7:0			F1AI[5:0]					
0xBC	RXESC	31:24								
		23:16								
		15:8						RBDS[2:0]		
		7:0		F1DS[2:0]				F0DS[2:0]		
0xC0	TXBC	31:24		TFQM	TFQS[5:0]					
		23:16			NDTB[5:0]					
		15:8	TBSA[15:8]							
		7:0	TBSA[7:0]							
0xC4	TXFQS	31:24								
		23:16			TFQF	TFQPI[4:0]				
		15:8				TFGI[4:0]				
		7:0			TFFL[5:0]					
0xC8	TXESC	31:24								
		23:16								
		15:8								
		7:0						TBDS[2:0]		
0xCC	TXBRP	31:24	TRP31	TRP30	TRP29	TRP28	TRP27	TRP26	TRP25	TRP24
		23:16	TRP23	TRP22	TRP21	TRP20	TRP19	TRP18	TRP17	TRP16
		15:8	TRP15	TRP14	TRP13	TRP12	TRP11	TRP10	TRP9	TRP8
		7:0	TRP7	TRP6	TRP5	TRP4	TRP3	TRP2	TRP1	TRP0
0xD0	TXBAR	31:24	AR31	AR30	AR29	AR28	AR27	AR26	AR25	AR24
		23:16	AR23	AR22	AR21	AR20	AR19	AR18	AR17	AR16
		15:8	AR15	AR14	AR13	AR12	AR11	AR10	AR9	AR8
		7:0	AR7	AR6	AR5	AR4	AR3	AR2	AR1	AR0
0xD4	TXBCR	31:24	CR31	CR30	CR29	CR28	CR27	CR26	CR25	CR24
		23:16	CR23	CR22	CR21	CR20	CR19	CR18	CR17	CR16
		15:8	CR15	CR14	CR13	CR12	CR11	CR10	CR9	CR8
		7:0	CR7	CR6	CR5	CR4	CR3	CR2	CR1	CR0
0xD8	TXBTO	31:24	TO31	TO30	TO29	TO28	TO27	TO26	TO25	TO24
		23:16	TO23	TO22	TO21	TO20	TO19	TO18	TO17	TO16
		15:8	TO15	TO14	TO13	TO12	TO11	TO10	TO9	TO8
		7:0	TO7	TO6	TO5	TO4	TO3	TO2	TO1	TO0

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0xDC	TXBCF	31:24	CF31	CF30	CF29	CF28	CF27	CF26	CF25	CF24
		23:16	CF23	CF22	CF21	CF20	CF19	CF18	CF17	CF16
		15:8	CF15	CF14	CF13	CF12	CF11	CF10	CF9	CF8
		7:0	CF7	CF6	CF5	CF4	CF3	CF2	CF1	CF0
0xE0	TXBTIE	31:24	TIE31	TIE30	TIE29	TIE28	TIE27	TIE26	TIE25	TIE24
		23:16	TIE23	TIE22	TIE21	TIE20	TIE19	TIE18	TIE17	TIE16
		15:8	TIE15	TIE14	TIE13	TIE12	TIE11	TIE10	TIE9	TIE8
		7:0	TIE7	TIE6	TIE5	TIE4	TIE3	TIE2	TIE1	TIE0
0xE4	TXBCIE	31:24	CFIE31	CFIE30	CFIE29	CFIE28	CFIE27	CFIE26	CFIE25	CFIE24
		23:16	CFIE23	CFIE22	CFIE21	CFIE20	CFIE19	CFIE18	CFIE17	CFIE16
		15:8	CFIE15	CFIE14	CFIE13	CFIE12	CFIE11	CFIE10	CFIE9	CFIE8
		7:0	CFIE7	CFIE6	CFIE5	CFIE4	CFIE3	CFIE2	CFIE1	CFIE0
0xE8 ... 0xEF	Reserved									
0xF0	TXEFC	31:24			EFWM[5:0]					
		23:16			EFS[5:0]					
		15:8	EFSA[15:8]							
		7:0	EFSA[7:0]							
0xF4	TXEFS	31:24							TEFL	EFF
		23:16				EFPI[4:0]				
		15:8				EFGI[4:0]				
		7:0				EFFL[4:0]				
0xF8	TXEFA	31:24								
		23:16								
		15:8								
		7:0				EFAI[4:0]				
0xFC ... 0xFF	Reserved									
0x0100	ERROR	31:24								
		23:16								
		15:8								
		7:0								BERR

39.7.1. Endian

Name: ENDN
Offset: 0x04
Reset: 0x87654321
Property: Read-only

Table 39-18. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	ETV[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	1	0	0	0	0	1	1	1

Bit	23	22	21	20	19	18	17	16
	ETV[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	1	1	0	0	1	0	1

Bit	15	14	13	12	11	10	9	8
	ETV[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	1	0	0	0	0	1	1

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

Bits 31:0 – ETV[31:0] Endianness Test Value
The endianness test value is 0x87654321

39.7.2. Message RAM Configuration

Name: MRCFG
Offset: 0x08
Reset: 0x00000002
Property: Write-restricted

This register is writable only if CCCR.CCE bit (CCCR <1>) is set.

Table 39-19. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	OFFSET[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
Access								
Reset								

Bits 23:16 – OFFSET[7:0] Message RAM Base Address Offset
This bitfield value represents the 8 bits offset of the memory base address (bits [23:16]).
The base address is calculated following the formula: Base_Address = 0x20000000+OFFSET << 16.

39.7.3. Data Bit Timing and Prescaler

Name: DBTP
Offset: 0x0C
Reset: 0x00000A33
Property: Write-restricted

This register is write-restricted and only writable if both CCCR.CCE bit (CCCR <1>) and CCCR.INIT bit (CCCR <0>) are set.

The CAN bit time may be programmed in the range of 4 to 49 time quanta. The CAN time quantum may be programmed in the range of 1 to 32 GCLK_CANx periods. time quantum (t_q) = (DBRP + 1) mtq. Therefore the length of the bit time is [DTSEG1 + DTSEG2 + 3] tq where DTSEG1 and DTSEG2 are programmed values in the DBTP register.

Note:
With a GCLK_CANx of 8MHz, the reset value 0x00000A33 configures the CAN for a fast bit rate of 500 kBits/s.

The bit rate configured for the CAN FD data phase via the DBTP register must be higher or equal to the bit rate configured for the arbitration phase via the NBTP register.

Table 39-20. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	TDC			DBRP[4:0]				
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
				DTSEG1[4:0]				
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	1	0	1	0

Bit	7	6	5	4	3	2	1	0
	DTSEG2[3:0]				DSJW[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	1	1	0	0	1	1

Bit 23 – TDC Transceiver Delay Compensation

Value	Description
0	Transceiver Delay Compensation disabled.
1	Transceiver Delay Compensation enabled.

Bits 20:16 – DBRP[4:0] Data Baud Rate Prescaler

Value	Description
0x00 – 0x1F	The value by which the GCLK_CANx is divided for generating the bit time quanta. The bit time is built up from a multiple of this quanta. Valid values for the Baud Rate Prescaler are 0 to 31. The actual interpretation by the hardware of this value is such that one more than the value programmed here is used.

Bits 12:8 – DTSEG1[4:0] Fast time segment before sample point

Value	Description
0x00 – 0x1F	Valid values are 0 to 31. The actual interpretation by the hardware of this value is such that one more than the programmed value is used. DTSEG1 is the sum of PROPAGATION TIME SEGMENT (PROP_SEG) and PHASE BUFFER SEGMENT1 (PHASE_SEG1).

Bits 7:4 – DTSEG2[3:0] Data time segment after sample point

Value	Description
0x0 – 0xF	Valid values are 0 to 15. The actual interpretation by the hardware of this value is such that one more than the programmed value is used. DTSEG2 is PHASE BUFFER SEGMENT2 (PHASE_SEG2).

Bits 3:0 – DSJW[3:0] Data (Re)Syncronization Jump Width

Value	Description
0x0 – 0xF	Valid values are 0 to 15. The actual interpretation by the hardware of this value is such that one more than the programmed value is used.

39.7.4. Test

Name: TEST
Offset: 0x10
Reset: 0x00000000
Property: Write-restricted

Table 39-21. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

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

Bit 7 – RX Receive Pin
This bit reflects the actual value of pin CANx_RX. The read value can be interpret as follows.

Value	Description
0	The CAN bus is dominant (CANx_RX = 0).
1	The CAN bus is recessive (CANx_RX = 1).

Bits 6:5 – TX[1:0] Control of Transmit Pin
This field defines the control of the transmit pin.

Value	Name	Description
0x0	CORE	Reset value, CANx_TX controlled by CAN core, updated at the end of CAN bit time.
0x1	SAMPLE	Sample Point can be monitored at pin CANx_TX.
0x2	DOMINANT	Dominant ('0') level at pin CANx_TX.
0x3	RECESSIVE	Recessive ('1') level at pin CANx_TX.

Bit 4 – LBCK Loop Back Mode

Value	Description
0	Loop Back Mode is disabled.
1	Loop Back Mode is enabled.

39.7.5. RAM Watchdog

Name: RWD
Offset: 0x14
Reset: 0x00000000
Property: Write-restricted

This register is write-restricted and writable only if both CCCR.CCE bit (CCCR <1>) and CCCR.INIT bit (CCCR <0>) are set.

The RAM Watchdog monitors the READY output of the Message RAM. A Message RAM access via the CAN's AHB Host Interface starts the Message RAM Watchdog Counter with the value configured by WDC bits (RWD <7:0>). The counter is reloaded with WDC bits (RWD <7:0>) when the Message RAM signals successful completion by activating its READY output. In case there is no response from the Message RAM until the counter has counted down to zero, the counter stops and interrupt IR.WDI bit (IR<26>) is set.

Table 39-22. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
	WDV[7:0]							
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
	WDC[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:8 – WDV[7:0] Watchdog Value
Actual Message RAM Watchdog Counter Value.

Bits 7:0 – WDC[7:0] Watchdog Configuration
Start value of the Message RAM Watchdog Counter. With the reset value of 0x00 the counter is disabled.

39.7.6. CC Control

Name: CCCR
Offset: 0x18
Reset: 0x00000001
Property: Write-restricted

Table 39-23. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
		TXP	EFBI	PXHD			BRSE	FDOE
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
	TEST	DAR	MON	CSR	CSA	ASM	CCE	INIT
Access	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

Bit 14 – TXP Transmit Pause

This bit field is write-restricted and only writable if both CCCR.CCE bit (CCCR <1>) and CCCR.INIT bit (CCCR <0>) are set.

Value	Description
0	Transmit pause disabled.
1	Transmit pause enabled. The CAN pauses for two CAN bit times before starting the next transmission after itself has successfully transmitted a frame.

Bit 13 – EFBI Edge Filtering during Bus Integration

Value	Description
0	Edge filtering is disabled.
1	Two consecutive dominant tq required to detect an edge for hard synchronization.

Bit 12 – PXHD Protocol Exception Handling Disable

Note: When protocol exception handling is disabled, the CAN will transmit an error frame when it detects a protocol exception condition.

Value	Description
0	Protocol exception handling enabled.
1	Protocol exception handling disabled.

Bit 9 – BRSE Bit Rate Switch Enable

Note: When CAN FD operation is disabled (i.e.,CCCR.FDOE bit (CCCR <8>) = 0), BRSE is not evaluated.

Value	Description
0	Bit rate switching for transmissions disabled.
1	Bit rate switching for transmissions enabled.

Bit 8 – FDOE FD Operation Enable

Value	Description
0	FD operation disabled.
1	FD operation enabled.

Bit 7 – TEST Test Mode Enable

This bit field is write-restricted.
Writing a 0 to this field is always allowed.
Writing a 1 to this field is allowed only if both CCCR.CCE bit (CCCR <1>) and CCCR.INIT bit (CCCR <0>) are set.

Value	Description
0	Normal operation. Register TEST holds reset values.
1	Test Mode, write access to register TEST enabled.

Bit 6 – DAR Disable Automatic Retransmission

This bit field is write-restricted and writable only if both CCCR.CCE bit (CCCR <1>) and CCCR.INIT bit (CCCR <0>) are set.

Value	Description
0	Automatic retransmission of messages not transmitted successfully enabled.
1	Automatic retransmission disabled.

Bit 5 – MON Bus Monitoring Mode

This bit field is write-restricted.
Writing a 0 to this field is always allowed.
Writing a 1 to this field is allowed only both CCCR.CCE bit (CCCR <1>) and CCCR.INIT bit (CCCR <0>) are set.

Value	Description
0	Bus Monitoring Mode is disabled.
1	Bus Monitoring Mode is enabled.

Bit 4 – CSR Clock Stop Request

Value	Description
0	No clock stop is requested.
1	Clock stop requested. When clock stop is requested, first CCCR.INIT bit (CCCR <0>) and then the CCCR.CSA bit (CCCR <3>) will be set after all pending transfer requests have been completed and the CAN bus reached idle.

Bit 3 – CSA Clock Stop Acknowledge

Value	Description
0	No clock stop acknowledged.
1	CAN may be set in power down by stopping CLK_CANx_AHB and GCLK_CANx.

Bit 2 – ASM Restricted Operation Mode

This bit field is write-restricted.
Writing a 0 to this field is always allowed.

Writing a 1 to this field is allowed only if both CCCR.CCE bit (CCCR <1>) and CCCR.INIT bit (CCCR <0>) are set.

Value	Description
0	Normal CAN operation.
1	Restricted Operation Mode active.

Bit 1 – CCE Configuration Change Enable

This bit field is write-restricted and only writable if bit field CCCR.INIT bit (CCCR <0>) is set.

Value	Description
0	The CPU has no write access to the protected configuration registers.
1	The CPU has write access to the protected configuration registers (while CCCR.INIT bit (CCCR <0>) =1).

Bit 0 – INIT Initialization

Due to the synchronization mechanism between the two clock domains, there may be a delay until the value written to the INIT bit can be read back. The programmer has to assure that the previous value written to the INIT bit has been accepted by reading the INIT bit before setting the INIT bit to a new value.

Value	Description
0	Normal Operation.
1	Initialization is started.

39.7.7. Nominal Bit Timing and Prescaler

Name: NBTP
Offset: 0x1C
Reset: 0x00000A33
Property: Write-restricted

This register is write-restricted and writable only if both CCCR.CCE bit (CCCR <1>) and CCCR.INIT bit (CCCR <0>) are set.

The CAN bit time may be programmed in the range of 4 to 385 time quanta. The CAN time quantum may be programmed in the range of 1 to 512 GCLK_CANx periods. tq = (NBRP + 1) mtq.

Therefore the length of the bit time is [NTSEG1 + NTSEG2 + 3] tq, where NTSEG1 and NTSEG2 are programmed values in the NBTP register.

Note: With a CAN clock (GCLK_CANx) of 8MHz, the reset value 0x06000A03 configures the CAN for a bit rate of 500 kBits/s.

Table 39-24. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	NSJW[6:0]							NBRP[8]
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	1	1	0

Bit	23	22	21	20	19	18	17	16
	NBRP[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
	NTSEG1[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	1	0	1	0

Bit	7	6	5	4	3	2	1	0
		NTSEG2[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	1	1

Bits 31:25 – NSJW[6:0] Nominal (Re)Syncronization Jump Width

Value	Description
`0x00 – 0x7F`	Valid values are 0 to 127. The actual interpretation by the hardware of this value is such that one more than the programmed value is used.

Bits 24:16 – NBRP[8:0] Nominal Baud Rate Prescaler

Value	Description
`0x000 – 0x1FF`	The value by which the GCLK_CANx is divided for generating the bit time quanta. The bit time is built up from a multiple of this quanta. Valid values for the Baud Rate Prescaler are 0 to 511. The actual interpretation by the hardware of this value is such that one more than the value programmed here is used.

Bits 15:8 – NTSEG1[7:0] Nominal Time segment before sample point

Value	Description
`0x00 – 0x7F`	Valid values are 1 to 255. The actual interpretation by the hardware of this value is such that one more than the programmed value is used. NTSEG1 is the sum of PROPAGATION TIME SEGMENT (PROP_SEG) and PHASE BUFFER SEGMENT1 (PHASE_SEG1).

Bits 6:0 – NTSEG2[6:0] Time segment after sample point

Value	Description
`0x00 – 0x7F`	Valid values are 0 to 127. The actual interpretation by the hardware of this value is such that one more than the programmed value is used. NTSEG2 is PHASE BUFFER SEGMENT2 (PHASE_SEG2).

39.7.8. Timestamp Counter Configuration

Name: TSCC
Offset: 0x20
Reset: 0x00000000
Property: Write-restricted

This register is write-restricted and writable only if both CCCR.CCE bit (CCCR <1>) and CCCR.INIT bit (CCCR <0>) are set.

Table 39-25. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

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

Bits 19:16 – TCP[3:0] Timestamp Counter Prescaler

Value	Description
`0x0 - 0xF`	Configures the timestamp and timeout counters time unit in multiples of CAN bit times [1…16]. The actual interpretation by the hardware of this value is such that one more than the value programmed here is used.

Bits 1:0 – TSS[1:0] Timestamp Select
This field defines the timestamp counter selection.

Value	Name	Description
`0x0 or 0x3`	ZERO	Timestamp counter value always 0x0000.
`0x1`	INC	Timestamp counter value incremented by TCP.
`0x2`	-	Reserved

39.7.9. Timestamp Counter Value

Name: TSCV
Offset: 0x24
Reset: 0x00000000
Property: Read-only

Notes:

1. A write access to TSCV clears the Timestamp Counter value.
2. A “wrap around” is a change of the Timestamp Counter value from non-zero to zero not caused by the write access to TSCV.

Table 39-26. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
	TSC[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
	TSC[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – TSC[15:0] Timestamp Counter

The internal Timestamp Counter value is captured on start of frame (both Rx and Tx). When bitfield TSCC.TSS (TSCC<1:0>) = 0x1, the Timestamp Counter is incremented in multiples of CAN bit times [1...16] depending on the configuration of bitfield TSCC.TCP (TSCC<19:16>) . A wrap around sets interrupt flag IR.TSW (IR<16>) .

39.7.10. Timeout Counter Configuration

Name: TOCC
Offset: 0x28
Reset: 0xFFFF0000
Property: Write-restricted

This register is write-restricted and writable if both CCCR.CCE bit (CCCR <1>) and CCCR.INIT bit (CCCR <0>) are set.

Table 39-27. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	TOP[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	23	22	21	20	19	18	17	16
	TOP[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

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

Bits 31:16 – TOP[15:0] Timeout Period

Start value of the Timeout Counter (down-counter). Configures the Timeout Period.

Bits 2:1 – TOS[1:0] Timeout Select

When operating in Continuous mode, a write to TOCV register presets the counter to the value configured by bit TOCC.TOP (TOCC<31:16>) and continues down-counting. When the Timeout Counter is controlled by one of the FIFOs, an empty FIFO presets the counter to the value configured by bit TOCC.TOP (TOCC<31:16>). Down-counting is started when the first FIFO element is stored.

Value	Name	Description
`0x0`	CONT	Continuous operation.
`0x1`	TXEF	Timeout controlled by TX Event FIFO.
`0x2`	RXF0	Timeout controlled by Rx FIFO 0.
`0x3`	RXF1	Timeout controlled by Rx FIFO 1.

Bit 0 – ETOC Enable Timeout Counter

Value	Description
`0`	Timeout Counter disabled.

Value	Description
1	Timeout Counter enabled.

39.7.11. Timeout Counter Value

Name: TOCV
Offset: 0x2C
Reset: 0x0000FFFF
Property: -

Note: A write access to TOCV reloads the Timeout Counter with the value of bit TOCC.TOP (TOCC<31:16>).

Table 39-28. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
	TOC[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
	TOC[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 – TOC[15:0] Timeout Counter
The Timeout Counter is decremented in multiples of CAN bit times [1…16] depending on the configuration of the TSCC.TCP bit (TSCC<19:16>). When decremented to zero, interrupt flag (IR.TOO bit (IR<18>)) is set and the Timeout Counter is stopped. Start and reset/restart conditions are configured via TOCC.TOS bit (TOCC<2:1>).

39.7.12. Error Counter

Name: ECR
Offset: 0x40
Reset: 0x00000000
Property: Read-only

Note: When CCCR.ASM bit (CCCR<2>) is set, the CAN protocol controller does not increment TEC and REC when a CAN protocol error is detected, but CEL is still incremented.

Table 39-29. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	CEL[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
	RP	REC[6:0]						
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
	TEC[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 23:16 – CEL[7:0] CAN Error Logging
The counter is incremented each time when a CAN protocol error causes the Transmit Error Counter or Receive Error Counter to be incremented. It is reset by read access to CEL. The counter stops at 0xFF; and the next increment of TEC or REC sets interrupt flag IR.ELO bit (IR<22>).

Bit 15 – RP Receive Error Passive

Bits 14:8 – REC[6:0] Receive Error Counter
Actual state of the Receive Error Counter, values between 0 and 127.

Bits 7:0 – TEC[7:0] Transmit Error Counter
Actual state of the Transmit Error Counter, values between 0 and 255.

39.7.13. Protocol Status

Name: PSR
Offset: 0x44
Reset: 0x00000707
Property: Read-only

Notes:

1. When a frame in CAN FD format has reached the data phase with BRS flag set, the next CAN event (error or valid frame) will be shown in PSR.DLEC bit field (PSR<10:8>) instead of LEC. An error in a fixed stuff bit of a CAN FD CRC sequence will be shown as a Form Error, not Stuff Error.
2. The 'bus off' recovery sequence (see CAN Specification Rev. 2.0 or ISO 11898-1) cannot be shortened by setting or resetting CCCR.INIT bit (CCCR <0>). If the device goes 'bus off', it will set CCCR.INIT bit (CCCR <0>) of its own accord, stopping all bus activities. Once CCCR.INIT bit (CCCR <0>) has been cleared by the CPU, the device will wait for 129 occurrences of Bus Idle (129 * 11 consecutive recessive bits) before resuming normal operation. At the end of the'bus off' recovery sequence, the Error Management Counters will be reset. During the wait time after the resetting of CCCR.INIT bit (CCCR <0>), each time a sequence of 11 recessive bits is monitored, a Bit0 Error code is written to PSR.LEC bit field (PSR <2:0>). This enables the CPU to readily check the status of CAN bus (whether bus is stuck at dominant level or continuously disturbed). ECR.REC bit-field (ECR<14:8>) is used to count these sequences.

Table 39-30. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
		TDCV[6:0]						
Access		R	R	R	R	R	R	R
Reset		0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
		PXE	RFDF	RBRS	RESI	DLEC[2:0]		
Access		R	R	R	R	R	R	R
Reset		0	0	0	0	1	1	1

Bit	7	6	5	4	3	2	1	0
	BO	EW	EP	ACT[1:0]		LEC[2:0]		
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	1	1	1

Bits 22:16 – TDCV[6:0] Transmitter Delay Compensation Value

Value	Description
`0x00 - 0x7F`	Position of the secondary sample point, defined by the sum of the measured delay from CANx_TX to CANx_RX and TDCR.TDCO bit-field (TDCR <14:8>). The SSP position is, in the data phase, the number of mtq between the start of the transmitted bit and the secondary sample point. Valid values are 0 to 127 mtq.

Bit 14 – PXE Protocol Exception Event

This field is cleared on read access. A recessive “reserved bit” following a recessive FDF bit is the example of Protocol Exception.

Value	Description
0	No protocol exception event occurred since last read access.
1	Protocol exception event occurred.

Bit 13 – RFDF Received a CAN FD Message

This field is cleared on read access.

Value	Description
0	Since this bit was reset by the CPU, no CAN FD message has been received.
1	Message in CAN FD format with FDF flag set has been received. This bit is set independent of acceptance filtering.

Bit 12 – RBRS BRS flag of last received CAN FD Message

This field is cleared on read access.

Value	Description
0	Last received CAN FD message did not have its BRS flag set.
1	Last received CAN FD message had its BRS flag set. This bit is set together with RFDF, independent of acceptance filtering.

Bit 11 – RESI ESI flag of last received CAN FD Message

This field is cleared on read access.

Value	Description
0	Last received CAN FD message did not have its ESI flag set.
1	Last received CAN FD message had its ESI flag set.

Bits 10:8 – DLEC[2:0] Data Last Error Code

Type of last error that occurred in the data phase of a CAN FD format frame with its BRS flag set. Coding is the same as for LEC. This field will be cleared to zero when a CAN FD format frame with its BRS flag set has been transferred (reception or transmission) without error.

Bit 7 – BO 'bus off' Status

Value	Description
0	The CAN is not 'bus off' state.
1	The CAN is in 'bus off' state.

Bit 6 – EW Error Warning Status

Value	Description
0	Both error counters are below the Error Warning limit of 96.
1	At least one of the error counter has reached the Error Warning limit of 96.

Bit 5 – EP Error Passive

Value	Description
0	The CAN is in the ‘error active’ state. It normally takes part in bus communication and sends an active error flag when an error has been detected.
1	The CAN is in the 'error passive' state.

Bits 4:3 – ACT[1:0] Activity

Monitors the module’s CAN communication state.

Value	Name	Description
0x0	SYNC	Node is synchronizing on CAN communication.
0x1	IDLE	Node is neither receiver nor transmitter.
0x2	RX	Node is operating as receiver.
0x3	TX	Node is operating as transmitter.

Bits 2:0 – LEC[2:0] Last Error Code

The LEC indicates the type of the last error to occur on the CAN bus. This field will be cleared to '0' when a message has been transferred (reception or transmission) without error.
This field is set on read access.

Value	Name	Description
0x0	NONE	No Error: No error occurred since LEC has been reset by successful reception or transmission.
0x1	STUFF	Stuff Error: More than 5 equal bits in a sequence have occurred in a part of a received message where this is not allowed.
0x2	FORM	Form Error: A fixed format part of a received frame has the wrong format.
0x3	ACK	Ack Error: The message transmitted by the CAN was not acknowledged by another node.
0x4	BIT1	Bit1 Error: During the transmission of a message (with the exception of the arbitration field), the device wanted to send a recessive level (bit of logical value '1'), but the monitored bus was dominant.
0x5	BIT0	Bit0 Error: During the transmission of a message (or acknowledge bit, or active error flag, or overload flag), the device wanted to send a dominant level (data or identifier bit logical value '0'), but the monitored bus value was recessive. During 'bus off' recovery this status is set each time a sequence of 11 recessive bits have been monitored. This enables the CPU to monitor the proceeding of the 'bus off' recovery sequence (indicating the bus is not stuck at dominant or continuously disturbed).
0x6	CRC	CRC Error: The CRC checksum of a received message was incorrect. The CRC of an incoming message does not match with the CRC calculated from the received data.
0x7	NC	No Change: Any read access to the Protocol Status Register re-initializes the LEC to '7'. When the LEC shows the value '7', no CAN bus event was detected since the last CPU read access to the Protocol Status Register.

39.7.14. Transmitter Delay Compensation

Name: TDCR
Offset: 0x48
Reset: 0x00000000
Property: Write-restricted

This register is write-restricted and only writable if bit fields CCCR.CCE bit (CCCR <1>) = 1 and CCCR.INIT bit (CCCR <0>) = 1.

Table 39-31. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
		TDCO[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

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

Bits 14:8 – TDCO[6:0] Transmitter Delay Compensation Offset

Value	Description
0x00 – 0x7F	Offset value defining the distance between the measured delay from CANx_TX to CANx_RX and the secondary sample point. Valid values are 0 to 127 mtq.

Bits 6:0 – TDCF[6:0] Transmitter Delay Compensation Filter Window Length

Value	Description
0x00 – 0x7F	Defines the minimum value for the SSP position, dominant edges on CANx_RX that would result in an earlier SSP position are ignored for transmitter delay measurement. The feature is enabled when TDCF is configured to a value greater than TDCO. Valid values are 0 to 127 mtq.

39.7.15. Interrupt

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

Note: Interrupt flags must be cleared and then read back to confirm the clear before exiting the ISR to avoid double interrupts.

The flags are set when one of the listed conditions is detected (edge-sensitive). A flag is cleared by writing a 1 to the corresponding bit field. Writing a 0 has no effect. A hard reset will clear the register.

Table 39-32. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
			ARA	PED	PEA	WDI	BO	EW
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
	EP	ELO			DRX	TOO	MRAF	TSW
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
	TEFL	TEFF	TEFW	TEFN	TFE	TCF	TC	HPM
Access	R/W	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
	RF1L	RF1F	RF1W	RF1N	RF0L	RF0F	RF0W	RF0N
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 29 – ARA Access to Reserved Address

Value	Description
0	No access to reserved address occurred.
1	Access to reserved address occurred.

Bit 28 – PED Protocol Error in Data Phase

Value	Description
0	No protocol error in data phase detected.
1	Protocol error in data phase detected (i.e., PSR.DLEC bits (PSR<10:8>) != 0 and PSR.DLEC bits (PSR<10:8>) != 0).

Bit 27 – PEA Protocol Error in Arbitration Phase

Value	Description
0	No protocol error in arbitration phase detected.
1	Protocol error in arbitration phase detected (i.e., PSR.LEC bits (PSR<2:0>) != 0 and PSR.LEC bits (PSR<2:0>) != 0).

Bit 26 – WDI Watchdog Interrupt

Value	Description
0	No Message RAM Watchdog event occurred.
1	Message RAM Watchdog event due to missing READY.

Bit 25 – BO 'bus off' Status

Value	Description
0	'bus off' status unchanged.
1	'bus off' status changed.

Bit 24 – EW Error Warning Status

Value	Description
0	Error Warning status unchanged.
1	Error Warning status changed.

Bit 23 – EP Error Passive

Value	Description
0	Error Passive status unchanged.
1	Error Passive status changed.

Bit 22 – ELO Error Logging Overflow

Value	Description
0	CAN Error Logging Counter did not overflow.
1	Overflow of CAN Error Logging Counter occurred.

Bit 19 – DRX Message stored in a Dedicated Rx Buffer
The flag is set whenever a received message has been stored into a dedicated Rx Buffer.

Value	Description
0	No Rx Buffer updated.
1	At least one received message stored into a Rx Buffer.

Bit 18 – TOO Timeout Occurred

Value	Description
0	No timeout.
1	Timeout reached.

Bit 17 – MRAF Message RAM Access Failure
The flag is set, when the Rx Handler:

- has not completed acceptance filtering or storage of an accepted message until the arbitration field of the following message has been received. In this case acceptance filtering or message storage is aborted and the Rx Handler starts processing of the following message.
- was not able to write a message to the Message RAM. In this case message storage is aborted.

In both cases the FIFO put index is not updated. The New Data flag for a dedicated Rx Buffer is not set, a partly stored message is overwritten when the next message is stored to this location.

The flag is also set when the Tx Handler was not able to read a message from the Message RAM in time. In this case message transmission is aborted. In case of a Tx Handler access failure the CAN is switched into Restricted Operation Mode. To leave Restricted Operation Mode, the Host CPU must clear CCCR.ASM bit (CCCR <2>).

Value	Description
0	No Message RAM access failure occurred.
1	Message RAM access failure occurred.

Bit 16 – TSW Timestamp Wraparound

Value	Description
0	No timestamp counter wrap-around.
1	Timestamp counter wrapped around.

Bit 15 – TEFL Tx Event FIFO Element Lost

Value	Description
0	No Tx Event FIFO element lost.
1	Tx Event FIFO element lost, also set after write attempt to Tx Event FIFO of size zero.

Bit 14 – TEFF Tx Event FIFO Full

Value	Description
0	Tx Event FIFO not full.
1	Tx Event FIFO full.

Bit 13 – TEFW Tx Event FIFO Watermark Reached

Value	Description
0	Tx Event FIFO fill level below watermark.
1	Tx Event FIFO fill level reached watermark.

Bit 12 – TEFN Tx Event FIFO New Entry

Value	Description
0	Tx Event FIFO unchanged.
1	Tx Handler wrote Tx Event FIFO element.

Bit 11 – TFE Tx FIFO Empty

Value	Description
0	Tx FIFO not-empty.
1	Tx FIFO empty.

Bit 10 – TCF Transmission Cancellation Finished

Value	Description
0	Transmission cancellation not finished.
1	Transmission cancellation finished.

Bit 9 – TC Timestamp Completed

Value	Description
0	No transmission completed.
1	Transmission completed.

Bit 8 – HPM High Priority Message

Value	Description
0	No high priority message received.
1	High priority message received.

Bit 7 – RF1L Rx FIFO 1 Message Lost

Value	Description
0	No Rx FIFO 1 message lost.
1	Rx FIFO 1 message lost. also set after write attempt to Rx FIFO 1 of size zero.

Bit 6 – RF1F Rx FIFO 1 Full

Value	Description
0	Rx FIFO 1 not full.
1	Rx FIFO 1 full.

Bit 5 – RF1W Rx FIFO 1 Watermark Reached

Value	Description
0	Rx FIFO 1 fill level below watermark.
1	Rx FIFO 1 fill level reached watermark.

Bit 4 – RF1N Rx FIFO 1 New Message

Value	Description
0	No new message written to Rx FIFO 1.
1	New message written to Rx FIFO 1.

Bit 3 – RF0L Rx FIFO 0 Message Lost

Value	Description
0	No Rx FIFO 0 message lost.
1	Rx FIFO 0 message lost. also set after write attempt to Rx FIFO 0 of size zero.

Bit 2 – RF0F Rx FIFO 0 Full

Value	Description
0	Rx FIFO 0 not full.
1	Rx FIFO 0 full.

Bit 1 – RF0W Rx FIFO 0 Watermark Reached

Value	Description
0	Rx FIFO 0 fill level below watermark.
1	Rx FIFO 0 fill level reached watermark.

Bit 0 – RF0N Rx FIFO 0 New Message

Value	Description
0	No new message written to Rx FIFO 0.
1	New message written to Rx FIFO 0.

39.7.16. Interrupt Enable

Name: IE
Offset: 0x54
Reset: 0x00000000
Property: -

The settings in the Interrupt Enable register determine which status changes in the Interrupt Register will be signalled on an interrupt line.

Table 39-33. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
			ARAE	PEDE	PEAE	WDIE	BOE	EWE
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
	EPE	ELOE			DRXE	TOOE	MRAFE	TSWE
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
	TEFLE	TEFFE	TEFWE	TEFNE	TFEE	TCFE	TCE	HPME
Access	R/W	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
	RF1LE	RF1FE	RF1WE	RF1NE	RF0LE	RF0FE	RF0WE	RF0NE
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 29 – ARAE Access to Reserved Address Interrupt Enable

Value	Description
0	Interrupt disabled.
1	Interrupt enabled.

Bit 28 – PEDE Protocol Error in Data Phase Interrupt Enable

Value	Description
0	Interrupt disabled.
1	Interrupt enabled.

Bit 27 – PEAE Protocol Error in Arbitration Phase Interrupt Enable

Value	Description
0	Interrupt disabled.
1	Interrupt enabled.

Bit 26 – WDIE Watchdog Interrupt Enable

Value	Description
0	Interrupt disabled.
1	Interrupt enabled.

Bit 25 – BOE 'bus off' Status Interrupt Enable

Value	Description
0	Interrupt disabled.
1	Interrupt enabled.

Bit 24 – EWE Error Warning Status Interrupt Enable

Value	Description
0	Interrupt disabled.
1	Interrupt enabled.

Bit 23 – EPE Error Passive Interrupt Enable

Value	Description
0	Interrupt disabled.
1	Interrupt enabled.

Bit 22 – ELOE Error Logging Overflow Interrupt Enable

Value	Description
0	Interrupt disabled.
1	Interrupt enabled.

Bit 19 – DRXE Message stored to Dedicated Rx Buffer Interrupt Enable

Value	Description
0	Interrupt disabled.
1	Interrupt enabled.

Bit 18 – TOOE Timeout Occurred Interrupt Enable

Value	Description
0	Interrupt disabled.
1	Interrupt enabled.

Bit 17 – MRAFE Message RAM Access Failure Interrupt Enable

Value	Description
0	Interrupt disabled.
1	Interrupt enabled.

Bit 16 – TSWE Timestamp Wraparound Interrupt Enable

Value	Description
0	Interrupt disabled.
1	Interrupt enabled.

Bit 15 – TEFLE Tx Event FIFO Event Lost Interrupt Enable

Value	Description
0	Interrupt disabled.
1	Interrupt enabled.

Bit 14 – TEFFE Tx Event FIFO Full Interrupt Enable

Value	Description
0	Interrupt disabled.
1	Interrupt enabled.

Bit 13 – TEFWE Tx Event FIFO Watermark Reached Interrupt Enable

Value	Description
0	Interrupt disabled.
1	Interrupt enabled.

Bit 12 – TEFNE Tx Event FIFO New Entry Interrupt Enable

Value	Description
0	Interrupt disabled.
1	Interrupt enabled.

Bit 11 – TFEE Tx FIFO Empty Interrupt Enable

Value	Description
0	Interrupt disabled.
1	Interrupt enabled.

Bit 10 – TCFE Transmission Cancellation Finished Interrupt Enable

Value	Description
0	Interrupt disabled.
1	Interrupt enabled.

Bit 9 – TCE Transmission Completed Interrupt Enable

Value	Description
0	Interrupt disabled.
1	Interrupt enabled.

Bit 8 – HPME High Priority Message Interrupt Enable

Value	Description
0	Interrupt disabled.
1	Interrupt enabled.

Bit 7 – RF1LE Rx FIFO 1 Message Lost Interrupt Enable

Value	Description
0	Interrupt disabled.
1	Interrupt enabled.

Bit 6 – RF1FE Rx FIFO 1 Full Interrupt Enable

Value	Description
0	Interrupt disabled.
1	Interrupt enabled.

Bit 5 – RF1WE Rx FIFO 1 Watermark Reached Interrupt Enable

Value	Description
0	Interrupt disabled.
1	Interrupt enabled.

Bit 4 – RF1NE Rx FIFO 1 New Message Interrupt Enable

Value	Description
0	Interrupt disabled.
1	Interrupt enabled.

Bit 3 – RF0LE Rx FIFO 0 Message Lost Interrupt Enable

Value	Description
0	Interrupt disabled.
1	Interrupt enabled.

Bit 2 – RF0FE Rx FIFO 0 Full Interrupt Enable

Value	Description
0	Interrupt disabled.
1	Interrupt enabled.

Bit 1 – RF0WE Rx FIFO 0 Watermark Reached Interrupt Enable

Value	Description
0	Interrupt disabled.
1	Interrupt enabled.

Bit 0 – RF0NE Rx FIFO 0 New Message Interrupt Enable

Value	Description
0	Interrupt disabled.
1	Interrupt enabled.

39.7.17. Interrupt Line Select

Name: ILS
Offset: 0x58
Reset: 0x00000000
Property: -

The Interrupt Line Select register assigns an interrupt generated by a specific interrupt flag from IR to one of the two module interrupt lines.

Table 39-34. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
			ARAL	PEDL	PEAL	WDIL	BOL	EWL
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
	EPL	ELOL			DRXL	TOOL	MRAFL	TSWL
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
	TEFLL	TEFFL	TEFWL	TEFNL	TFEL	TCFL	TCL	HPML
Access	R/W	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
	RF1LL	RF1FL	RF1WL	RF1NL	RF0LL	RF0FL	RF0WL	RF0NL
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 29 – ARAL Access to Reserved Address Interrupt Line

Value	Description
0	Interrupt assigned to CAN interrupt line 0.
1	Interrupt assigned to CAN interrupt line 1.

Bit 28 – PEDL Protocol Error in Data Phase Interrupt Line

Value	Description
0	Interrupt assigned to CAN interrupt line 0.
1	Interrupt assigned to CAN interrupt line 1.

Bit 27 – PEAL Protocol Error in Arbitration Phase Interrupt Line

Value	Description
0	Interrupt assigned to CAN interrupt line 0.
1	Interrupt assigned to CAN interrupt line 1.

Bit 26 – WDIL Watchdog Interrupt Line

Value	Description
0	Interrupt assigned to CAN interrupt line 0.
1	Interrupt assigned to CAN interrupt line 1.

Bit 25 – BOL 'bus off' Status Interrupt Line

Value	Description
0	Interrupt assigned to CAN interrupt line 0.
1	Interrupt assigned to CAN interrupt line 1.

Bit 24 – EWL Error Warning Status Interrupt Line

Value	Description
0	Interrupt assigned to CAN interrupt line 0.
1	Interrupt assigned to CAN interrupt line 1.

Bit 23 – EPL Error Passive Interrupt Line

Value	Description
0	Interrupt assigned to CAN interrupt line 0.
1	Interrupt assigned to CAN interrupt line 1.

Bit 22 – ELOL Error Logging Overflow Interrupt Line

Value	Description
0	Interrupt assigned to CAN interrupt line 0.
1	Interrupt assigned to CAN interrupt line 1.

Bit 19 – DRXL Message stored to Dedicated Rx Buffer Interrupt Line

Value	Description
0	Interrupt assigned to CAN interrupt line 0.
1	Interrupt assigned to CAN interrupt line 1.

Bit 18 – TOOL Timeout Occurred Interrupt Line

Value	Description
0	Interrupt assigned to CAN interrupt line 0.
1	Interrupt assigned to CAN interrupt line 1.

Bit 17 – MRAFL Message RAM Access Failure Interrupt Line

Value	Description
0	Interrupt assigned to CAN interrupt line 0.
1	Interrupt assigned to CAN interrupt line 1.

Bit 16 – TSWL Timestamp Wraparound Interrupt Line

Value	Description
0	Interrupt assigned to CAN interrupt line 0.
1	Interrupt assigned to CAN interrupt line 1.

Bit 15 – TEFLL Tx Event FIFO Event Lost Interrupt Line

Value	Description
0	Interrupt assigned to CAN interrupt line 0.
1	Interrupt assigned to CAN interrupt line 1.

Bit 14 – TEFFL Tx Event FIFO Full Interrupt Line

Value	Description
0	Interrupt assigned to CAN interrupt line 0.
1	Interrupt assigned to CAN interrupt line 1.

Bit 13 – TEFWL Tx Event FIFO Watermark Reached Interrupt Line

Value	Description
0	Interrupt assigned to CAN interrupt line 0.
1	Interrupt assigned to CAN interrupt line 1.

Bit 12 – TEFNL Tx Event FIFO New Entry Interrupt Line

Value	Description
0	Interrupt assigned to CAN interrupt line 0.
1	Interrupt assigned to CAN interrupt line 1.

Bit 11 – TFEL Tx FIFO Empty Interrupt Line

Value	Description
0	Interrupt assigned to CAN interrupt line 0.
1	Interrupt assigned to CAN interrupt line 1.

Bit 10 – TCFL Transmission Cancellation Finished Interrupt Line

Value	Description
0	Interrupt assigned to CAN interrupt line 0.
1	Interrupt assigned to CAN interrupt line 1.

Bit 9 – TCL Transmission Completed Interrupt Line

Value	Description
0	Interrupt assigned to CAN interrupt line 0.
1	Interrupt assigned to CAN interrupt line 1.

Bit 8 – HPML High Priority Message Interrupt Line

Value	Description
0	Interrupt assigned to CAN interrupt line 0.
1	Interrupt assigned to CAN interrupt line 1.

Bit 7 – RF1LL Rx FIFO 1 Message Lost Interrupt Line

Value	Description
0	Interrupt assigned to CAN interrupt line 0.
1	Interrupt assigned to CAN interrupt line 1.

Bit 6 – RF1FL Rx FIFO 1 Full Interrupt Line

Value	Description
0	Interrupt assigned to CAN interrupt line 0.
1	Interrupt assigned to CAN interrupt line 1.

Bit 5 – RF1WL Rx FIFO 1 Watermark Reached Interrupt Line

Value	Description
0	Interrupt assigned to CAN interrupt line 0.
1	Interrupt assigned to CAN interrupt line 1.

Bit 4 – RF1NL Rx FIFO 1 New Message Interrupt Line

Value	Description
0	Interrupt assigned to CAN interrupt line 0.
1	Interrupt assigned to CAN interrupt line 1.

Bit 3 – RF0LL Rx FIFO 0 Message Lost Interrupt Line

Value	Description
0	Interrupt assigned to CAN interrupt line 0.
1	Interrupt assigned to CAN interrupt line 1.

Bit 2 – RF0FL Rx FIFO 0 Full Interrupt Line

Value	Description
0	Interrupt assigned to CAN interrupt line 0.
1	Interrupt assigned to CAN interrupt line 1.

Bit 1 – RF0WL Rx FIFO 0 Watermark Reached Interrupt Line

Value	Description
0	Interrupt assigned to CAN interrupt line 0.
1	Interrupt assigned to CAN interrupt line 1.

Bit 0 – RF0NL Rx FIFO 0 New Message Interrupt Line

Value	Description
0	Interrupt assigned to CAN interrupt line 0.
1	Interrupt assigned to CAN interrupt line 1.

39.7.18. Interrupt Line Enable

Name: ILE
Offset: 0x5C
Reset: 0x00000000
Property: -

Table 39-35. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
							EINT1	EINT0
Access							R/W	R/W
Reset							0	0

Bit 1 – EINT1 Enable Interrupt Line 1

Value	Description
0	CAN interrupt line 1 disabled.
1	CAN interrupt line 1 enabled.

Bit 0 – EINT0 Enable Interrupt Line 0

Value	Description
0	CAN interrupt line 0 disabled.
1	CAN interrupt line 0 enabled.

39.7.19. Global Filter Configuration

Name: GFC
Offset: 0x80
Reset: 0x00000000
Property: Write-restricted

This register is write-restricted and only writable if bit fields CCCR.CCE bit (CCCR <1>) = 1 and CCCR.INIT bit (CCCR <0>) = 1.

Table 39-36. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			ANFS[1:0]		ANFE[1:0]		RRFS	RRFE
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bits 5:4 – ANFS[1:0] Accept Non-matching Frames Standard
Defines how received messages with 11-bit IDs that do not match any element of the filter list are treated.

Value	Name	Description
`0x0`	RXF0	Accept in Rx FIFO 0.
`0x1`	RXF1	Accept in Rx FIFO 1.
`0x1x`	REJECT	Reject

Bits 3:2 – ANFE[1:0] Accept Non-matching Frames Extended
Defines how received messages with 29-bit IDs that do not match any element of the filter list are treated.

Value	Name	Description
`0x0`	RXF0	Accept in Rx FIFO 0.
`0x1`	RXF1	Accept in Rx FIFO 1.
`0x1x`	REJECT	Reject

Bit 1 – RRFS Reject Remote Frames Standard

Value	Description
0	Filter remote frames with 11-bit standard IDs.
1	Reject all remote frames with 11-bit standard IDs.

Bit 0 – RRFE Reject Remote Frames Extended

Value	Description
0	Filter remote frames with 29-bit extended IDs.
1	Reject all remote frames with 29-bit extended IDS.

39.7.20. Standard ID Filter Configuration

Name: SIDFC
Offset: 0x84
Reset: 0x00000000
Property: Write-restricted

This register is write-restricted and only writable if bit fields CCCR.CCE bit (CCCR <1>) = 1 and CCCR.INIT bit (CCCR <0>) = 1.

Table 39-37. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	LSS[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
	FLSSA[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
	FLSSA[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 – LSS[7:0] List Size Standard

Value	Description
`0`	No standard Message ID filter.
`1 – 128`	Number of standard Message ID filter elements.
`> 128`	Values greater than 128 are interpreted as 128.

Bits 15:0 – FLSSA[15:0] Filter List Standard Start Address
Start address of standard Message ID filter list. When the CAN module addresses the Message RAM it addresses 32-bit words, not single bytes. The configurable start addresses are 32-bit word addresses, i.e. only bits 15 to 2 are evaluated, the two least significant bits are ignored. Bits 1 to 0 will always be read back as “00”.

39.7.21. Extended ID Filter Configuration

Name: XIDFC
Offset: 0x88
Reset: 0x00000000
Property: Write-restricted

Table 39-38. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
		LSE[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
	FLESA[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
	FLESA[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 22:16 – LSE[6:0] List Size Extended

Value	Description
`0`	No extended Message ID filter.
`1 - 64`	Number of Extended Message ID filter elements.
`> 64`	Values greater than 64 are interpreted as 64.

Bits 15:0 – FLESA[15:0] Filter List Extended Start Address

Start address of extended Message ID filter list. When the CAN module addresses the Message RAM it addresses 32-bit words, not single bytes. The configurable start addresses are 32-bit word addresses, i.e. only bits 15 to 2 are evaluated, the two least significant bits are ignored. Bits 1 to 0 will always be read back as “00”.

39.7.22. Extended ID AND Mask

Name: XIDAM
Offset: 0x90
Reset: 0x1FFFFFFF
Property: Write-restricted

This register is write-restricted and only writable if bit fields CCCR.CCE bit (CCCR <1>) = 1 and CCCR.INIT bit (CCCR <0>) = 1.

Table 39-39. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
				EIDM[28:24]				
Access				R/W	R/W	R/W	R/W	R/W
Reset				1	1	1	1	1

Bit	23	22	21	20	19	18	17	16
	EIDM[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
	EIDM[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
	EIDM[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 28:0 – EIDM[28:0] Extended ID Mask
For acceptance filtering of extended frames the Extended ID AND Mask is ANDed with the Message ID of a received frame. Intended for masking of 29-bit IDs in SAE J1939. With the reset value of all bits set to one the mask is not active.

39.7.23. High Priority Message Status

Name: HPMS
Offset: 0x94
Reset: 0x00000000
Property: Read-only

This register is updated every time a Message ID filter element configured to generate a priority event matches. This can be used to monitor the status of incoming high priority messages and to enable fast access to these messages.

Table 39-40. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
	FLST	FIDX[6:0]						
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
	MSI[1:0]		BIDX[5:0]					
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 15 – FLST Filter List
Indicates the filter list of the matching filter element.

Value	Description
0	Standard Filter List.
1	Extended Filter List.

Bits 14:8 – FIDX[6:0] Filter Index
Index of matching filter element. Range is 0 to SIDFC.LSS bits (SIDFC <23:16>) bits (XIDFC <22:16>) - 1 (standard) or XIDFC.LSE bits (XIDFC <22:16>)- 1 (extended).

Bits 7:6 – MSI[1:0] Message Storage Indicator
This field defines the message storage information to a FIFO.

Value	Name	Description
0x0	NONE	No FIFO selected.
0x1	LOST	FIFO message lost.
0x2	FIFO0	Message stored in FIFO 0.

Value	Name	Description
0x3	FIFO1	Message stored in FIFO 1.

Bits 5:0 – BIDX[5:0] Buffer Index
Index of Rx FIFO element to which the message was stored. Only valid when MSI[1] = 1.

39.7.24. New Data 1

Name: NDAT1
Offset: 0x98
Reset: 0x00000000
Property: -

Table 39-41. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	ND31	ND30	ND29	ND28	ND27	ND26	ND25	ND24
Access	R/W	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
	ND23	ND22	ND21	ND20	ND19	ND18	ND17	ND16
Access	R/W	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
	ND15	ND14	ND13	ND12	ND11	ND10	ND9	ND8
Access	R/W	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
	ND7	ND6	ND5	ND4	ND3	ND2	ND1	ND0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – NDn New Data n [n = 0..31]

The register holds the New Data flags of Rx Buffers 0 to 31. The flags are set when the respective Rx Buffer has been updated from a received frame. The flags remain set until the Host clears them. A flag is cleared by writing 1 to the corresponding bit position. Writing a 0 has no effect. A hard reset will clear the register.

39.7.25. New Data 2

Name: NDAT2
Offset: 0x9C
Reset: 0x00000000
Property: -

Table 39-42. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	ND63	ND62	ND61	ND60	ND59	ND58	ND57	ND56
Access	R/W	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
	ND55	ND54	ND53	ND52	ND51	ND50	ND49	ND48
Access	R/W	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
	ND47	ND46	ND45	ND44	ND43	ND42	ND41	ND40
Access	R/W	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
	ND39	ND38	ND37	ND36	ND35	ND34	ND33	ND32
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – NDn New Data [n = 32..63]

The register holds the New Data flags of Rx Buffers 32 to 63. The flags are set when the respective Rx Buffer has been updated from a received frame. The flags remain set until the Host clears them. A flag is cleared by writing 1 to the corresponding bit position. Writing a 0 has no effect. A hard reset will clear the register.

39.7.26. Rx FIFO 0 Configuration

Name: RXF0C
Offset: 0xA0
Reset: 0x00000000
Property: Write-restricted

Table 39-43. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	F0OM	F0WM[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	23	22	21	20	19	18	17	16
		F0S[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
	F0SA[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
	F0SA[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 31 – F0OM FIFO 0 Operation Mode

FIFO 0 can be operated in blocking or in overwrite mode.

Value	Description
0	FIFO 0 blocking mode.
1	FIFO 0 overwrite mode.

Bits 30:24 – F0WM[6:0] Rx FIFO 0 Watermark

Value	Description
0	Watermark interrupt disabled.
1 - 64	Level for Rx FIFO 0 watermark interrupt (IR.RF0W bit (IR <1>)).
>64	Watermark interrupt disabled.

Bits 22:16 – F0S[6:0] Rx FIFO 0 Size

The Rx FIFO 0 elements are indexed from 0 to F0S - 1.

Value	Description
0	No Rx FIFO 0
1 - 64	Number of Rx FIFO 0 elements.
>64	Values greater than 64 are interpreted as 64.

Bits 15:0 – F0SA[15:0] Rx FIFO 0 Start Address

Start address of Rx FIFO 0 in Message RAM. When the CAN module addresses the Message RAM it addresses 32-bit words, not single bytes. The configurable start addresses are 32-bit word addresses, i.e. only bits 15 to 2 are evaluated, the two least significant bits are ignored. Bits 1 to 0 will always be read back as “00”.

39.7.27. Rx FIFO 0 Status

Name: RXF0S
Offset: 0xA4
Reset: 0x00000000
Property: Read-only

Table 39-44. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
							RF0L	F0F
Access							R	R
Reset							0	0

Bit	23	22	21	20	19	18	17	16
			F0PI[5:0]					
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
			F0GI[5:0]					
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
		F0FL[6:0]						
Access		R	R	R	R	R	R	R
Reset		0	0	0	0	0	0	0

Bit 25 – RF0L Rx FIFO 0 Message Lost

This bit is a copy of interrupt flag IR.RF0L bit (IR <3>). When IR.RF0L bit (IR <3>) is reset, this bit is also reset.

Overwriting the oldest message when RXF0C.F0OM bit (RXF0C <31>) = '1' will not set this flag.

Value	Description
0	No Rx FIFO 0 message lost.
1	Rx FIFO 0 message lost, also set after write attempt to Rx FIFO 0 of size zero.

Bit 24 – F0F Rx FIFO 0 Full

Value	Description
0	Rx FIFO 0 not full.
1	Rx FIFO 0 full.

Bits 21:16 – F0PI[5:0] Rx FIFO 0 Put Index

Rx FIFO 0 write index pointer, range 0 to 63.

Bits 13:8 – F0GI[5:0] Rx FIFO 0 Get Index

Rx FIFO 0 read index pointer, range 0 to 63.

Bits 6:0 – F0FL[6:0] Rx FIFO 0 Fill Level
Number of elements stored in Rx FIFO 0, range 0 to 64.

39.7.28. Rx FIFO 0 Acknowledge

Name: RXF0A
Offset: 0xA8
Reset: 0x00000000
Property: -

Table 39-45. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			F0AI[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 – F0AI[5:0] Rx FIFO 0 Acknowledge Index
After the Host has read a message or a sequence of messages from Rx FIFO 0 it has to write the buffer index of the last element read from Rx FIFO 0 to F0AI. This will set the Rx FIFO 0 Get Index RXF0S.F0GI bits (RXF0S <13:8>) to F0AI + 1 and update the FIFO 0 Fill Level RXF0S.F0FL bit (RXF0S <25>).

39.7.29. Rx Buffer Configuration

Name: RXBC
Offset: 0xAC
Reset: 0x00000000
Property: Write-restricted

This register is write-restricted and only writable if bit fields CCCR.CCE bit (CCCR <1>) = 1 and CCCR.INIT bit (CCCR <0>) = 1.

Table 39-46. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
	RBSA[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
	RBSA[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 – RBSA[15:0] Rx Buffer Start Address
Configures the start address of the Rx Buffers section in the Message RAM. Also used to reference debug message A,B,C. When the CAN module addresses the Message RAM it addresses 32-bit words, not single bytes. The configurable start addresses are 32-bit word addresses, i.e. only bits 15 to 2 are evaluated, the two least significant bits are ignored. Bits 1 to 0 will always be read back as “00”.

39.7.30. Rx FIFO 1 Configuration

Name: RXF1C
Offset: 0xB0
Reset: 0x00000000
Property: Write-restricted

This register is write-restricted and only writable if bit fields CCCR.CCE bit (CCCR <1>) = 1 and CCCR.INIT bit (CCCR <0>) = 1.

Table 39-47. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	F1OM	F1WM[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	23	22	21	20	19	18	17	16
		F1S[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
	F1SA[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
	F1SA[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 31 – F1OM FIFO 1 Operation Mode
FIFO 1 can be operated in blocking or in overwrite mode.

Value	Description
0	FIFO 1 blocking mode.
1	FIFO 1 overwrite mode.

Bits 30:24 – F1WM[6:0] Rx FIFO 1 Watermark

Value	Description
0	Watermark interrupt disabled.
1 - 64	Level for Rx FIFO 1 watermark interrupt (IR.RF1W bit (IR <5>)).
>64	Watermark interrupt disabled.

Bits 22:16 – F1S[6:0] Rx FIFO 1 Size
The Rx FIFO 1 elements are indexed from 0 to F1S - 1.

Value	Description
0	No Rx FIFO 1

Value	Description
1 - 64	Number of Rx FIFO 1 elements.
>64	Values greater than 64 are interpreted as 64.

Bits 15:0 – F1SA[15:0] Rx FIFO 1 Start Address
Start address of Rx FIFO 1 in Message RAM. When the CAN module addresses the Message RAM it addresses 32-bit words, not single bytes. The configurable start addresses are 32-bit word addresses, i.e. only bits 15 to 2 are evaluated, the two least significant bits are ignored. Bits 1 to 0 will always be read back as “00”.

39.7.31. Rx FIFO 1 Status

Name: RXF1S
Offset: 0xB4
Reset: 0x00000000
Property: Read-only

Table 39-48. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	DMS[1:0]						RF1L	F1F
Access	R	R					R	R
Reset	0	0					0	0

Bit	23	22	21	20	19	18	17	16
			F1PI[5:0]					
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
			F1GI[5:0]					
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
		F1FL[6:0]						
Access		R	R	R	R	R	R	R
Reset		0	0	0	0	0	0	0

Bits 31:30 – DMS[1:0] Debug Message Status
This field defines the debug message status.

Value	Name	Description
0x0	IDLE	Idle state, wait for reception of debug messages, DMA request is cleared.
0x1	DBGA	Debug message A received.
0x2	DBGB	Debug message A, B received.
0x3	DBGC	Debug message A, B, C received, DMA request is set.

Bit 25 – RF1L Rx FIFO 1 Message Lost
This bit is a copy of interrupt flag IR.RF1L bit (IR <7>). When IR.RF1L bit (IR <7>) is reset, this bit is also reset.
Overwriting the oldest message when RXF1C.F1OM bit (RXF1C <31>) = '1' will not set this flag.

Value	Description
0	No Rx FIFO 1 message lost.
1	Rx FIFO 1 message lost, also set after write attempt to Rx FIFO 1 of size zero.

Bit 24 – F1F Rx FIFO 1 Full

Value	Description
0	Rx FIFO 1 not full.

Value	Description
1	Rx FIFO 1 full.

Bits 21:16 – F1PI[5:0] Rx FIFO 1 Put Index
Rx FIFO 1 write index pointer, range 0 to 63.

Bits 13:8 – F1GI[5:0] Rx FIFO 1 Get Index
Rx FIFO 1 read index pointer, range 0 to 63.

Bits 6:0 – F1FL[6:0] Rx FIFO 1 Fill Level
Number of elements stored in Rx FIFO 1, range 0 to 64.

39.7.32. Rx FIFO 1 Acknowledge

Name: RXF1A
Offset: 0xB8
Reset: 0x00000000
Property: -

Table 39-49. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			F1AI[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 – F1AI[5:0] Rx FIFO 1 Acknowledge Index
After the Host has read a message or a sequence of messages from Rx FIFO 1 it has to write the buffer index of the last element read from Rx FIFO 1 to F1AI. This will set the Rx FIFO 1 Get Index RXF1S.F1GI bits (RXF1S <13:8>) to F1AI + 1 and update the FIFO 1 Fill Level RXF1S.F1FL bits (RXF1S <6:0>).

39.7.33. Rx Buffer / FIFO Element Size Configuration

Name: RXESC
Offset: 0xBC
Reset: 0x00000000
Property: Write-restricted

This register is write-restricted and only writable if bit fields CCCR.CCE bit (CCCR <1>) = 1 and CCCR.INIT bit (CCCR <0>) = 1.

Configures the number of data bytes belonging to an Rx Buffer / Rx FIFO element. Data field sizes >8 bytes are intended for CAN FD operation only.

Table 39-50. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
						RBDS[2:0]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	7	6	5	4	3	2	1	0
		F1DS[2:0]				F0DS[2:0]		
Access		R/W	R/W	R/W		R/W	R/W	R/W
Reset		0	0	0		0	0	0

Bits 10:8 – RBDS[2:0] Rx Buffer Data Field Size

In case the data field size of an accepted CAN frame exceeds the data field size configured for the matching Rx Buffer, only the number of bytes as configured by RXESC are stored to the Rx Buffer element. The rest of the frame's data field is ignored.

Value	Name	Description
0x0	DATA8	8 byte data field.
0x1	DATA12	12 byte data field.
0x2	DATA16	16 byte data field.
0x3	DATA20	20 byte data field.
0x4	DATA24	24 byte data field.
0x5	DATA32	32 byte data field.
0x6	DATA48	48 byte data field.
0x7	DATA64	64 byte data field.

Bits 6:4 – F1DS[2:0] Rx FIFO 1 Data Field Size
In case the data field size of an accepted CAN frame exceeds the data field size configured for the matching Rx FIFO 1, only the number of bytes as configured by RXESC are stored to the Rx FIFO 1 element. The rest of the frame's data field is ignored.

Value	Name	Description
0x0	DATA8	8 byte data field.
0x1	DATA12	12 byte data field.
0x2	DATA16	16 byte data field.
0x3	DATA20	20 byte data field.
0x4	DATA24	24 byte data field.
0x5	DATA32	32 byte data field.
0x6	DATA48	48 byte data field.
0x7	DATA64	64 byte data field.

Bits 2:0 – F0DS[2:0] Rx FIFO 0 Data Field Size
In case the data field size of an accepted CAN frame exceeds the data field size configured for the matching Rx FIFO 0, only the number of bytes as configured by RXESC are stored to the Rx FIFO 0 element. The rest of the frame's data field is ignored.

Value	Name	Description
0x0	DATA8	8 byte data field.
0x1	DATA12	12 byte data field.
0x2	DATA16	16 byte data field.
0x3	DATA20	20 byte data field.
0x4	DATA24	24 byte data field.
0x5	DATA32	32 byte data field.
0x6	DATA48	48 byte data field.
0x7	DATA64	64 byte data field.

39.7.34. Tx Buffer Configuration

Name: TXBC
Offset: 0xC0
Reset: 0x00000000
Property: Write-restricted

This register is write-restricted and only writable if bit fields CCCR.CCE bit (CCCR <1>) = 1 and CCCR.INIT bit (CCCR <0>) = 1.

Note: Be aware that the sum of TFQS and NDTB may not be greater than 32. There is no check for erroneous configurations. The Tx Buffers section in the Message RAM starts with the dedicated Tx Buffers.

Table 39-51. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
		TFQM	TFQS[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	23	22	21	20	19	18	17	16
			NDTB[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
	TBSA[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
	TBSA[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 30 – TFQM Tx FIFO/Queue Mode

Value	Description
0	Tx FIFO operation.
1	Tx Queue operation.

Bits 29:24 – TFQS[5:0] Transmit FIFO/Queue Size

Value	Description
0	No Tx FIFO/Queue.
1 - 32	Number of Tx Buffers used for Tx FIFO/Queue.
>32	Values greater than 32 are interpreted as 32.

Bits 21:16 – NDTB[5:0] Number of Dedicated Transmit Buffers

Value	Description
0	No Tx FIFO/Queue.
1 – 32	Number of Tx Buffers used for Tx FIFO/Queue.
>32	Values greater than 32 are interpreted as 32.

Bits 15:0 – TBSA[15:0] Tx Buffers Start Address
Start address of Tx Buffers section in Message RAM. When the CAN module addresses the Message RAM it addresses 32-bit words, not single bytes. The configurable start addresses are 32-bit word addresses, i.e. only bits 15 to 2 are evaluated, the two least significant bits are ignored. Bits 1 to 0 will always be read back as “00”.

39.7.35. Tx FIFO/Queue Status

Name: TXFQS
Offset: 0xC4
Reset: 0x00000000
Property: Read-only

Note: In case of mixed configurations where dedicated Tx Buffers are combined with a Tx FIFO or a Tx Queue, the Put and Get Indexes indicate the number of the Tx Buffer starting with the first dedicated Tx Buffers. Example: For a configuration of 12 dedicated Tx Buffers and a Tx FIFO of 20 Buffers a Put Index of 15 points to the fourth buffer of the Tx FIFO.

Table 39-52. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
			TFQF	TFQPI[4:0]				
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
				TFGI[4:0]				
Access				R	R	R	R	R
Reset				0	0	0	0	0

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

Bit 21 – TFQF Tx FIFO/Queue Full

Value	Description
0	Tx FIFO/Queue not full.
1	Tx FIFO/Queue full.

Bits 20:16 – TFQPI[4:0] Tx FIFO/Queue Put Index
Tx FIFO/Queue write index pointer, range 0 to 31.

Bits 12:8 – TFGI[4:0] Tx FIFO/Queue Get Index
Tx FIFO read index pointer, range 0 to 31. Read as zero when Tx Queue operation is configured (TXBC.TFQM bit (TXBC <30>) = ‘1’).

Bits 5:0 – TFFL[5:0] Tx FIFO Free Level
Number of consecutive free Tx FIFO elements starting from TFGI, range 0 to 32. Read as zero when Tx Queue operation is configured (TXBC.TFQM bit (TXBC <30>) = ‘1’).

39.7.36. Tx Buffer Element Size Configuration

Name: TXESC
Offset: 0xC8
Reset: 0x00000000
Property: Write-restricted

This register is write-restricted and only writable if bit fields CCCR.CCE bit (CCCR <1>) = 1 and CCCR.INIT bit (CCCR <0>) = 1.

Configures the number of data bytes belonging to a Tx Buffer element. Data field sizes >8 bytes are intended for CAN FD operation only.

Table 39-53. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

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

Bits 2:0 – TBDS[2:0] Tx Buffer Data Field Size
In case the data length code DLC of a Tx Buffer element is configured to a value higher than the Tx Buffer data field size TXESC.TBDS (TXESC <2:0>), the bytes not defined by the Tx Buffer are transmitted as “0xCC” (padding bytes).

Value	Name	Description
`0x0`	DATA8	8 byte data field.
`0x1`	DATA12	12 byte data field.
`0x2`	DATA16	16 byte data field.
`0x3`	DATA20	20 byte data field.
`0x4`	DATA24	24 byte data field.
`0x5`	DATA32	32 byte data field.
`0x6`	DATA48	48 byte data field.
`0x7`	DATA64	64 byte data field.

39.7.37. Tx Buffer Request Pending

Name: TXBRP
Offset: 0xCC
Reset: 0x00000000
Property: Read-only

Note: TXBRP bits which are set while a Tx scan is in progress are not considered during this particular Tx scan. In case a cancellation is requested for such a Tx Buffer, this Add Request is canceled immediately, the corresponding TXBRP bit is reset.

Table 39-54. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	TRP31	TRP30	TRP29	TRP28	TRP27	TRP26	TRP25	TRP24
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
	TRP23	TRP22	TRP21	TRP20	TRP19	TRP18	TRP17	TRP16
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
	TRP15	TRP14	TRP13	TRP12	TRP11	TRP10	TRP9	TRP8
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
	TRP7	TRP6	TRP5	TRP4	TRP3	TRP2	TRP1	TRP0
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, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – TRPn Transmission Request Pending

Each Tx Buffer has its own Transmission Request Pending bit.
The bits are reset after a requested transmission has completed or has been cancelled via register TXBCR.
TXBRP bits are set only for those Tx Buffers configured via TXBC. After a TXBRP bit has been set, a Tx scan is started to check for the pending Tx request with the highest priority (Tx Buffer with lowest Message ID).
A cancellation request resets the corresponding transmission request pending bit of register TXBRP. In case a transmission has already been started when a cancellation is requested, this is done at the end of the transmission, regardless whether the transmission was successful or not. The cancellation request bits are reset directly after the corresponding TXBRP bit has been reset.
After a cancellation has been requested, a finished cancellation is signaled via TXBCF

- after successful transmission together with the corresponding TXBTO bit
- when the transmission has not yet been started at the point of cancellation
- when the transmission has been aborted due to lost arbitration

- when an error occurred during frame transmission

In DAR mode all transmissions are automatically canceled if they are not successful. The corresponding TXBCF bit is set for all unsuccessful transmissions.

Value	Description
0	No transmission request pending.
1	Transmission request pending.

39.7.38. Tx Buffer Add Request

Name: TXBAR
Offset: 0xD0
Reset: 0x00000000
Property: -

Note: If an add request is applied for a Tx Buffer with pending transmission request (corresponding TXBRP bit is already set), this add request is ignored.

Table 39-55. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	AR31	AR30	AR29	AR28	AR27	AR26	AR25	AR24
Access	R/W	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
	AR23	AR22	AR21	AR20	AR19	AR18	AR17	AR16
Access	R/W	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
	AR15	AR14	AR13	AR12	AR11	AR10	AR9	AR8
Access	R/W	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
	AR7	AR6	AR5	AR4	AR3	AR2	AR1	AR0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – ARn Add Request

Each Tx Buffer has its own Add Request bit.
Writing a ‘1’ will set the corresponding Add Request bit; writing a ‘0’ has no impact. This enables the Host to set transmission requests for multiple Tx Buffers with one write to TXBAR. TXBAR bits are set only for those Tx Buffers configured via TXBC. When no Tx scan is running, the bits are reset immediately, else the bits remain set until the Tx scan process has completed.

39.7.39. Tx Buffer Cancellation Request

Name: TXBCR
Offset: 0xD4
Reset: 0x00000000
Property: -

Table 39-56. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	CR31	CR30	CR29	CR28	CR27	CR26	CR25	CR24
Access	R/W	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
	CR23	CR22	CR21	CR20	CR19	CR18	CR17	CR16
Access	R/W	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
	CR15	CR14	CR13	CR12	CR11	CR10	CR9	CR8
Access	R/W	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
	CR7	CR6	CR5	CR4	CR3	CR2	CR1	CR0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – CRn Cancellation Request

Each Tx Buffer has its own Cancellation Request bit.
Writing a '1' will set the corresponding Cancellation Request bit; writing a '0' has no impact. This enables the Host to set cancellation requests for multiple Tx Buffers with one write to TXBCR. TXBCR bits are set only for those Tx Buffers configured via TXBC. The bits remain set until the corresponding bit of TXBRP is reset.

Value	Description
0	No cancellation pending.
1	Cancellation pending.

39.7.40. Tx Buffer Transmission Occurred

Name: TXBTO
Offset: 0xD8
Reset: 0x00000000
Property: Read-only

Table 39-57. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	TO31	TO30	TO29	TO28	TO27	TO26	TO25	TO24
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
	TO23	TO22	TO21	TO20	TO19	TO18	TO17	TO16
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
	TO15	TO14	TO13	TO12	TO11	TO10	TO9	TO8
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
	TO7	TO6	TO5	TO4	TO3	TO2	TO1	TO0
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, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – TOn Transmission Occurred

Each Tx Buffer has its own Transmission Occurred bit.
The bits are set when the corresponding TXBRP bit is cleared after a successful transmission.
The bits are reset when a new transmission is requested by writing ‘1’ to the corresponding bit of register TXBAR.

39.7.41. Tx Buffer Cancellation Finished

Name: TXBCF
Offset: 0xDC
Reset: 0x00000000
Property: Read-only

Table 39-58. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	CF31	CF30	CF29	CF28	CF27	CF26	CF25	CF24
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
	CF23	CF22	CF21	CF20	CF19	CF18	CF17	CF16
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
	CF15	CF14	CF13	CF12	CF11	CF10	CF9	CF8
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
	CF7	CF6	CF5	CF4	CF3	CF2	CF1	CF0
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, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – CFn Cancellation Finished

Each Tx Buffer has its own Cancellation Finished bit.

The bits are set when the corresponding TXBRP bit is cleared after a cancellation was requested via TXBCR. In case the corresponding TXBRP bit was not set at the point of cancellation, CF is set immediately.

The bits are reset when a new transmission is requested by writing ‘1’ to the corresponding bit of register TXBAR.

39.7.42. Tx Buffer Transmission Interrupt Enable

Name: TXBTIE
Offset: 0xE0
Reset: 0x00000000
Property: -

Table 39-59. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	TIE31	TIE30	TIE29	TIE28	TIE27	TIE26	TIE25	TIE24
Access	R/W	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
	TIE23	TIE22	TIE21	TIE20	TIE19	TIE18	TIE17	TIE16
Access	R/W	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
	TIE15	TIE14	TIE13	TIE12	TIE11	TIE10	TIE9	TIE8
Access	R/W	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
	TIE7	TIE6	TIE5	TIE4	TIE3	TIE2	TIE1	TIE0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – TIEn Transmission Interrupt Enable

Each Tx Buffer has its own Transmission Interrupt Enable bit.

Value	Description
0	Transmission interrupt disabled.
1	Transmission interrupt enabled.

39.7.43. Tx Buffer Cancellation Finished Interrupt Enable

Name: TXBCIE
Offset: 0xE4
Reset: 0x00000000
Property: -

Table 39-60. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	CFIE31	CFIE30	CFIE29	CFIE28	CFIE27	CFIE26	CFIE25	CFIE24
Access	R/W	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
	CFIE23	CFIE22	CFIE21	CFIE20	CFIE19	CFIE18	CFIE17	CFIE16
Access	R/W	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
	CFIE15	CFIE14	CFIE13	CFIE12	CFIE11	CFIE10	CFIE9	CFIE8
Access	R/W	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
	CFIE7	CFIE6	CFIE5	CFIE4	CFIE3	CFIE2	CFIE1	CFIE0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – CFIEn Cancellation Finished Interrupt Enable

Each Tx Buffer has its own Cancellation Finished Interrupt Enable bit.

Value	Description
0	Cancellation finished interrupt disabled.
1	Cancellation finished interrupt enabled.

39.7.44. Tx Event FIFO Configuration

Name: TXEFC
Offset: 0xF0
Reset: 0x00000000
Property: Write-restricted

This register is write-restricted and only writable if bit fields CCCR.CCE bit (CCCR <1>) = 1 and CCCR.INIT bit (CCCR <0>) = 1.

Table 39-61. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
			EFWM[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
			EFS[5:0]					
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	EFSA[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
	EFSA[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 29:24 – EFWM[5:0] Event FIFO Watermark

Value	Description
0	Watermark interrupt disabled.
1 – 32	Level for Tx Event FIFO watermark interrupt (IR.TEFW bit (IR <13>)).
>32	Watermark interrupt disabled.

Bits 21:16 – EFS[5:0] Event FIFO Size
The Tx Event FIFO elements are indexed from 0 to EFS - 1.

Value	Description
0	Tx Event FIFO disabled
1 – 32	Number of Tx Event FIFO elements.
>32	Values greater than 32 are interpreted as 32.

Bits 15:0 – EFSA[15:0] Event FIFO Start Address
Start address of Tx Event FIFO in Message RAM. When the CAN module addresses the Message RAM it addresses 32-bit words, not single bytes. The configurable start addresses are 32-bit word

addresses, i.e. only bits 15 to 2 are evaluated, the two least significant bits are ignored. Bits 1 to 0 will always be read back as “00”.

39.7.45. Tx Event FIFO Status

Name: TXEFS
Offset: 0xF4
Reset: 0x00000000
Property: Read-only

Table 39-62. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
							TEFL	EFF
Access							R	R
Reset							0	0

Bit	23	22	21	20	19	18	17	16
				EFPI[4:0]				
Access				R	R	R	R	R
Reset				0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
				EFGI[4:0]				
Access				R	R	R	R	R
Reset				0	0	0	0	0

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

Bit 25 – TEFL Tx Event FIFO Element Lost

This bit is a copy of interrupt flag IR.TEFL bit (IR <15>). When IR.TEFL bit (IR <15>) is reset, this bit is also reset.

Value	Description
0	No Tx Event FIFO element lost.
1	Tx Event FIFO element lost, also set after write attempt to Tx Event FIFO of size zero.

Bit 24 – EFF Event FIFO Full

Value	Description
0	Tx Event FIFO not full.
1	Tx Event FIFO full.

Bits 20:16 – EFPI[4:0] Event FIFO Put Index

Tx Event FIFO write index pointer, range 0 to 31.

Bits 12:8 – EFGI[4:0] Event FIFO Get Index

Tx Event FIFO read index pointer, range 0 to 31.

Bits 4:0 – EFFL[4:0] Event FIFO Fill Level
Number of elements stored in Tx Event FIFO, range 0 to 32.

39.7.46. Tx Event FIFO Acknowledge

Name: TXEFA
Offset: 0xF8
Reset: 0x00000000
Property: -

Table 39-63. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

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

Bits 4:0 – EFAI[4:0] Event FIFO Acknowledge Index
After the Host has read an element or a sequence of elements from the Tx Event FIFO it has to write the index of the last element read from Tx Event FIFO to EFAI. This will set the Tx Event FIFO Get Index TXEFS.EFGI bits (TXEFS <12:8>) to EFAI + 1 and update the FIFO 0 Fill Level TXEFS.EFFL bits (TXEFS <4:0>).

39.7.47. Error Interrupt Flag

Name: ERROR
Offset: 0x100
Reset: 0x00000000
Property: -

Note: Interrupt flags must be cleared and then read back to confirm the clear before exiting the ISR to avoid double interrupts.

Table 39-64. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

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

Bit 0 – BERR AHB Bus Error Detection
The flag is set when an AHB bus error is detected. The flag is cleared by writing a 1 to the corresponding bit field. Writing a 0 has no effect. A hard reset will clear the register. When the bit is set, the Error non-maskable interrupt is generated.

Value	Description
0	No bus error detection.
1	Bus error detection.

40. External Bus Interface (EBI)

40.1. Overview

The External Bus Interface, only available on 100-pin and 144-pin devices, is designed to provide data transfer between external devices and the embedded Memory Controller of an Arm-based microcontroller.

The EBI module generates the signals that control the access to the external memory devices or peripheral devices. It has 4 chip selects, a 24-bit address bus, and a configurable 8-bit or 16-bit data bus. Separate read and write control signals allow for direct memory and peripheral interfacing. Read and write signal waveforms are fully adjustable.

The EBI module can manage wait requests from external devices to extend access times. The EBI module provides an automatic Slow clock mode. In Slow clock mode, it switches from user-programmed waveforms to slow-rate specific waveforms on read and write signals. It also supports asynchronous burst read in Page mode access for page sizes up to 32 bytes.

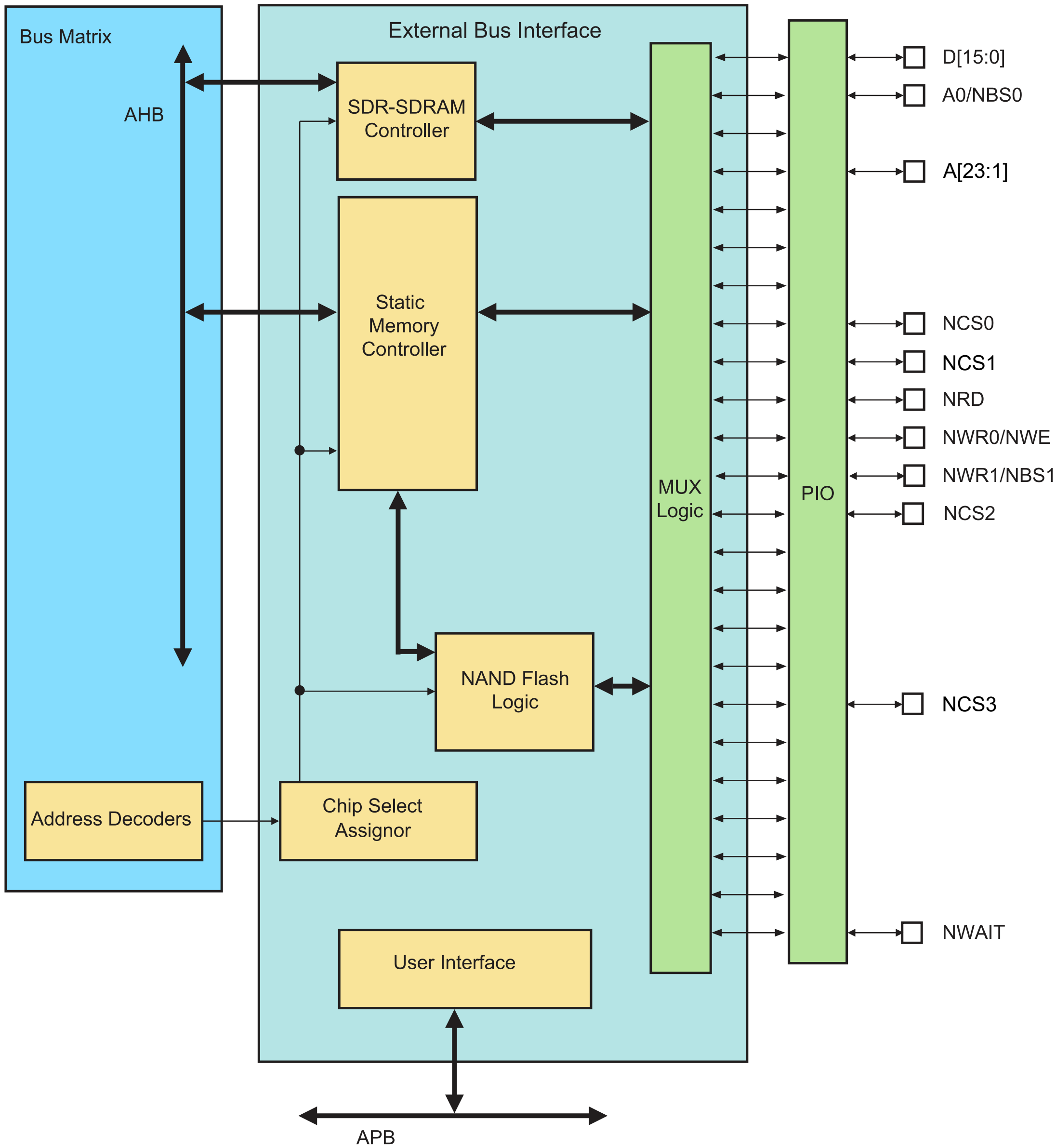
40.2. Features

The following are key features of the EBI module:

- Up to 24-bit Address Bus (up to 16 Mbytes linear per chip select)
- Up to four Chip Selects
- 8-bit or 16-bit Data Bus
- Word, Halfword, Byte Transfers
- Byte Write or Byte Select Lines
- Programmable Setup, And Hold Time for Read Signals per Chip Select
- Programmable Setup, And Hold Time for Write Signals per Chip Select
- Programmable Data Float Time per Chip Select
- Compliant with most LCD Module interfaces as well as FLASH, SRAM and DRAM
- External Wait Request
- Automatic Switch to Slow Clock Mode
- Asynchronous Read in Page Mode Supported: Page Size Ranges from 4 to 32 Bytes

40.3. EBI Block Diagram

Figure 40-1. Organization of the External Bus Interface



40.4. I/O Lines Description

Table 40-1. EBI I/O Lines Description

Name	Description	Type	Active Level
EBI_NCS[3:0]	Static Memory Controller Chip Select Lines	Output	Low
EBI_NRD	Read Signal	Output	Low
EBI_NWR0	Write 0 Signal	Output	Low
EBI_NBS0	Byte 0 Select Signal	Output	Low
EBI_NBS1	Byte 1 Select Signal	Output	Low
EBI_NWR1	Write 1 Signal	Output	Low
EBI_NWE	Write Enable Signal	Output	Low

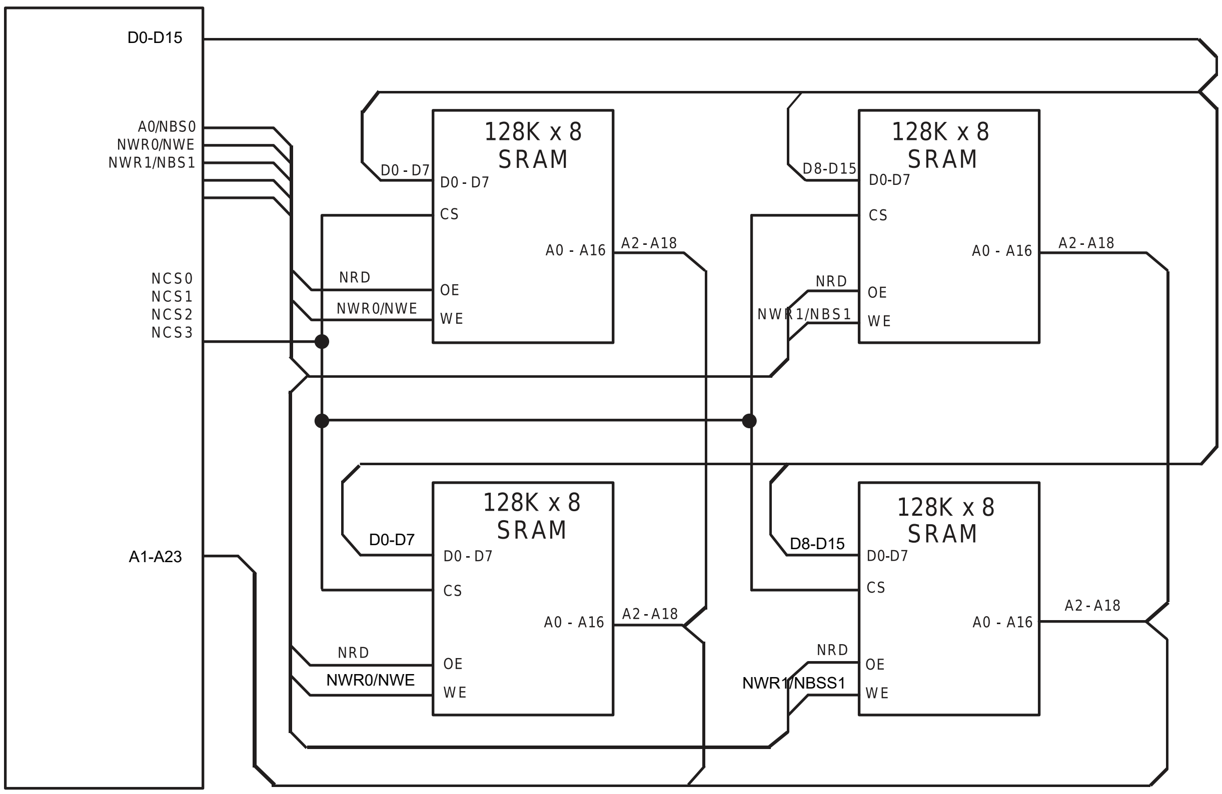
Table 40-1. EBI I/O Lines Description (continued)

Name	Description	Type	Active Level
EBI_A[23:0]	Address Bus	Output	–
EBI_D[15:0]	Data Bus	I/O	–
EBI_NWAIT	External Wait Signal	Input	Low

40.5. Application Example

40.5.1. Hardware Interface

Figure 40-2. EBI Interface Connections to External Memory Devices Example



40.6. Peripheral Dependencies

Peripheral Name	Base Address	MCLK AHBx/APBx Clock Enable Mask Bit	PAC Peripheral Identifier (PAC.WRCTRL.PERIDx)	Power Domain
EBI	0x4502_C000	AHB : MCLK.CLKMSK0[21] APB : MCLK.CLKMSK3[18]	58	VDDREG

40.6.1. I/O Lines

The pins used for interfacing the External Bus Interface may be multiplexed with the PIO lines. The programmer must first program the PIO controller to assign the External Bus Interface pins to their peripheral function. If I/O lines of the External Bus Interface are not used by the application, they can be used for other purposes by the PIO Controller.

40.7. Functional Description

The EBI transfers data between the internal AHB Bus (handled by the Bus Matrix) and the external memories or peripheral devices. It controls the waveforms and the parameters of the external address, data and control buses. It is composed of the following elements:

- Static Memory Controller(SMC)
- A chip select assignment feature that assigns an AHB address space to the external devices

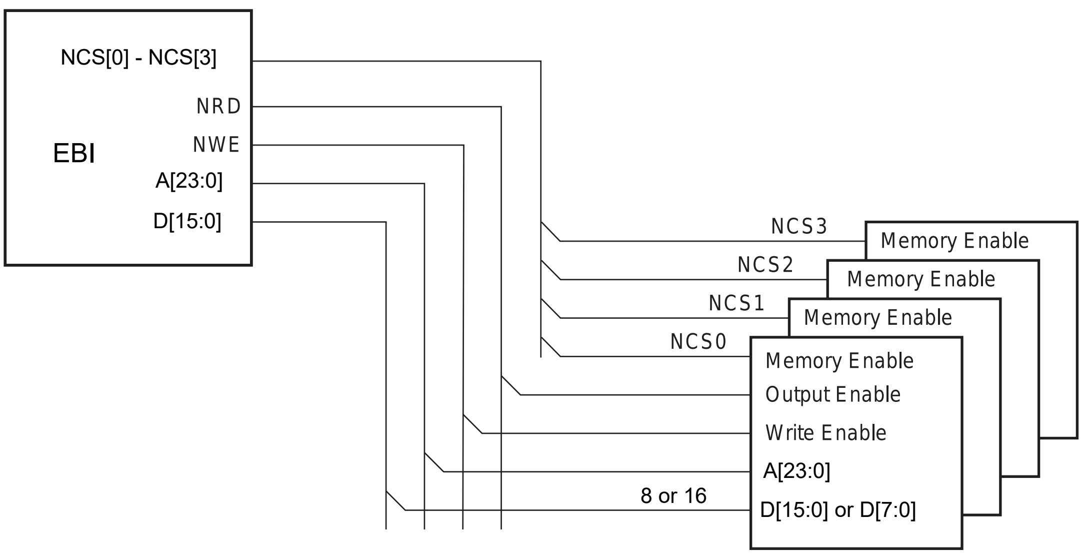
40.7.1. External Memory Mapping

The EBI provides up to 24 address lines, A[23:0]. This allows each chip select line to address up to 16 Mbytes of memory.

If the physical memory device connected on one chip select is smaller than 16 Mbytes, it wraps around and appears to be repeated within this space. The EBI handles any valid access to the memory device within the page, as shown in the following figure.

The LSB of A[23:0], A0, is only significant for 8 bit memories and not used for 16bit bit wide memories, A[23:1].

Figure 40-3. Memory Connections for Four External Devices



40.7.2. Connection to External Devices

40.7.2.1. Data Bus Width

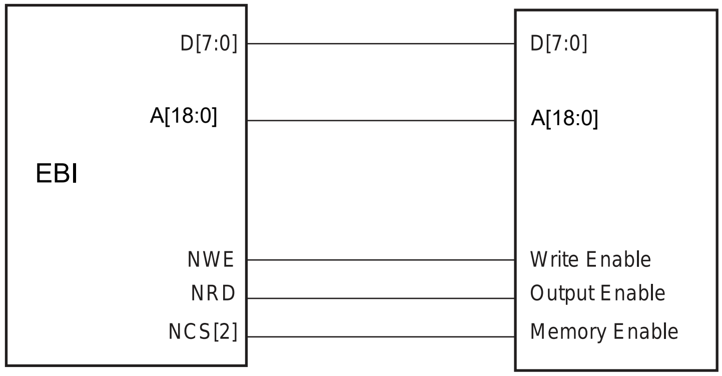
A data bus width of 8 or 16 bits can be selected for each chip select. This option is con-trolled by the field DBW in SMC_MODE (Mode Register) for the corresponding chip select.

40.7.2.2. Byte Write or Byte Select Access

Each chip select with an 8 or 16 bit data bus can operate with one of two different types of write access: byte write or byte select access. This is controlled by the BAT field of the SMC_MODE register for the corresponding chip select.

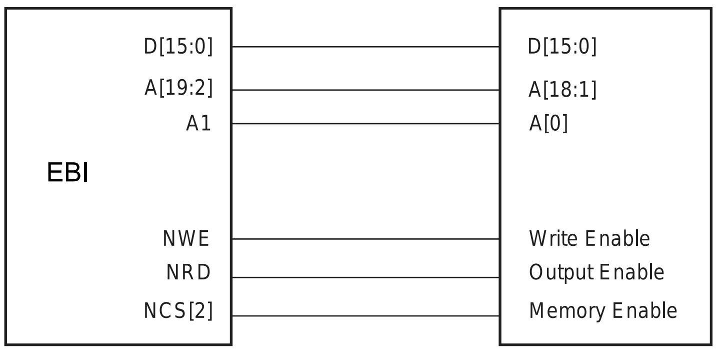
The following figure shows how to connect a 512K x 8-bit memory on NCS2.

Figure 40-4. Memory Connection for an 8-bit Data Bus



The following figure shows how to connect a 512K x 16-bit memory on NCS2.

Figure 40-5. Memory Connection for a 16-bit Data Bus



40.7.2.2.1. Byte Write Access

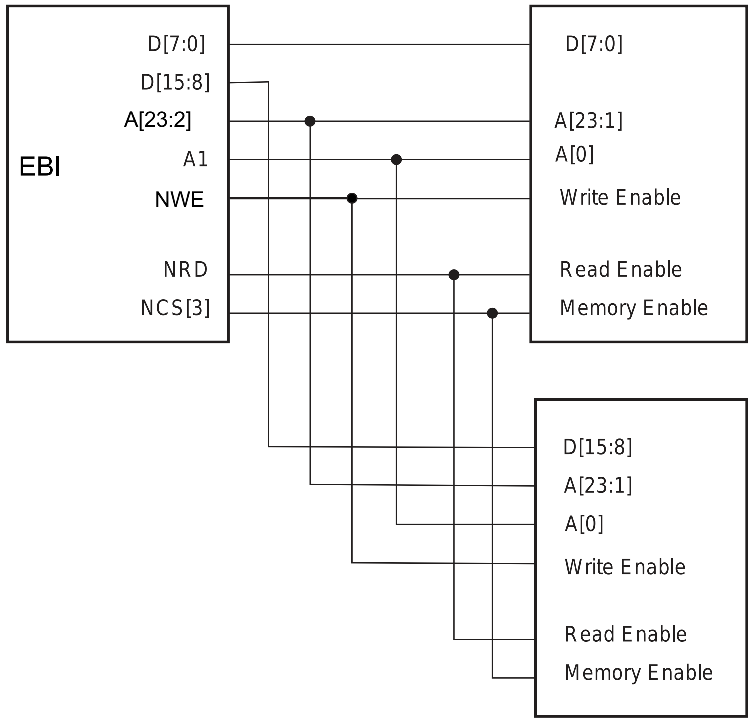
Byte write access supports one byte write signal per byte of the data bus and a single read signal.

Note that the EBI does not allow boot in Byte Write Access mode.

- For 16-bit devices: the EBI provides NWR0 and NWR1 write signals for respectively byte0 (lower byte) and byte1 (upper byte) of a 16-bit bus. One single read signal (NRD) is provided.

Byte Write Access is used to connect 2 x 8-bit devices as a 16-bit memory.

Figure 40-6. Connection of a 2x8-bit Devices on a 16-bit Bus



40.7.2.2.2. Byte Select Access

In this mode, read/write operations can be enabled/disabled at a byte level. One byte-select line per byte of the data bus is provided. One NRD and one NWE signal control read and write.

- For 16-bit devices: the EBI provides NBS0 and NBS1 selection signals for respectively byte0 (lower byte) and byte1 (upper byte) of a 16-bit bus.

Byte Select Access is used to connect one 16-bit device.

40.7.3. Signal Multiplexing

Depending on the byte access type (BAT), only the byte write signals or the byte select signals are used. To save I/Os at the external bus interface, control signals at the EBI interface are multiplexed. The following table shows signal multiplexing depending on the data bus width and the byte access type.

For32-bit devices, bits A0 and A1 are unused. For 16-bit devices, bit A0 of address is unused. When the Byte Select option is selected, NWR1 to NWR3 are unused. When the Byte Write option is selected, NBS0 to NBS3 are unused.

Table 40-2. EBI Multiplexed Signal Translation

Signal Name	16-bit Bus		8-bit Bus
Device Type	1 x 16-bit	2 x 8-bit	1 x 8-bit
Byte Access Type (BAT)	Byte Select	Byte Write	–
NBS0_A0	NBS0	–	A0
NWE_NWR0	NWE	NWR0	NWE
NBS1_NWR1	NBS1	NWR1	–

40.7.4. Standard Read and Write Protocols

In the following sections, the byte access type is not considered. Byte select lines (NBS0 and NBS1) always have the same timing as the address bus. NWE represents either the NEW signal in byte select access type or one of the byte write lines (NWR0 to NWR1) in byte write access type. NWR0 to NWR1 have the same timings and protocol as NWE. In the same way, NCS represents one of the NCS[0..NB_CS-1] chip select lines.

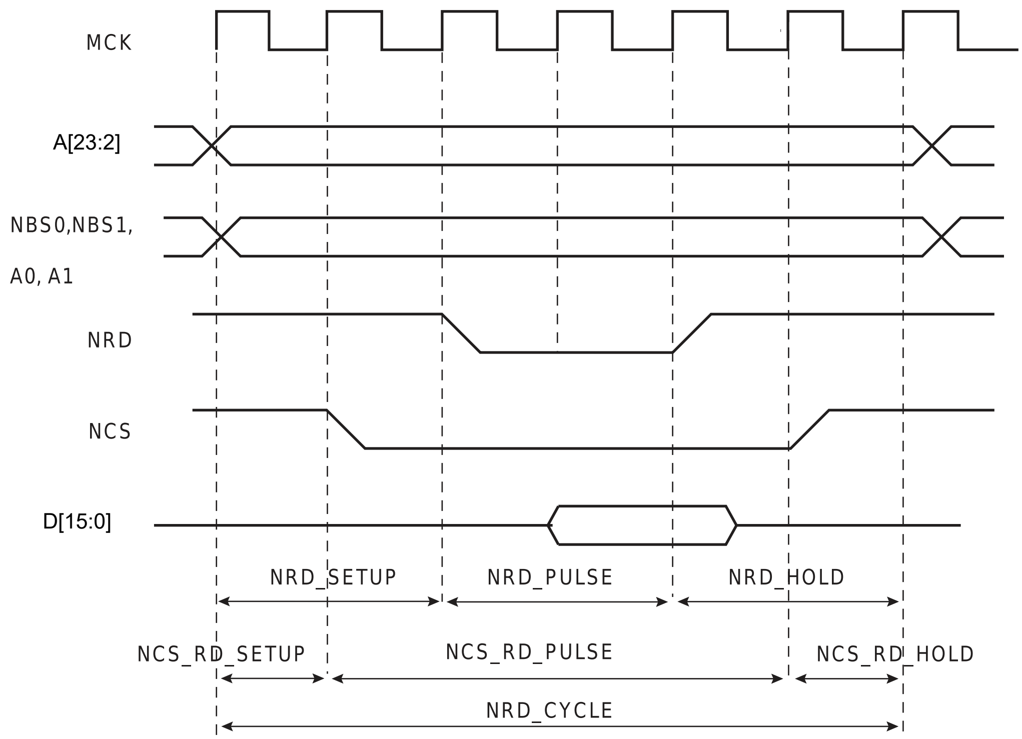
40.7.4.1. Read Waveforms

The read cycle is shown in the following figure. The read cycle starts with the address setting on the memory address bus:

{A[23:2], A1, A0} for 8-bit devices

{A[23:2], A1} for 16-bit devices

Figure 40-7. Standard Read Cycle



40.7.4.2. NRD Waveform

The NRD signal is characterized by a setup timing, a pulse width and a hold timing.

- **nrdsetup:** NRD setup time is defined as the setup of address before the NRD falling edge
- **nrdpulse:** NRD pulse length is the time between NRD falling edge and NRD rising edge
- **nrdhold:** NRD hold time is defined as the hold time of address after the NRD rising edge

40.7.4.3. NCS Waveform

The NCS signal can be divided into a setup time, pulse length and hold time:

- **ncsrd setup:** NCS setup time is defined as the setup time of address before the NCS falling edge
- **ncsrd pulse:** NCS pulse length is the time between NCS falling edge and NCS rising edge
- **ncsrd hold:** NCS hold time is defined as the hold time of address after the NCS rising edge

40.7.4.4. Read Cycle

The NRD CYCLE time is defined as the total duration of the read cycle, i.e., from the time where address is set on the address bus to the point where address may change. The total read cycle time is defined as:

NRD CYCLE = NRD SETUP + NRD PULSE + NRD HOLD, as well as

NRD CYCLE = NCS RD SETUP + NCS RD PULSE + NCS RD HOLD

All NRD and NCS timings are defined separately for each chip select as an integer number of Host Clock cycles. The NRD CYCLE field is common to both the NRD and NCS signals, therefore the timing period is of the same duration.

NRD CYCLE, NRD SETUP, and NRD PULSE implicitly define the NRD HOLD value as:

NRD HOLD = NRD CYCLE - NRD SETUP - NRD PULSE

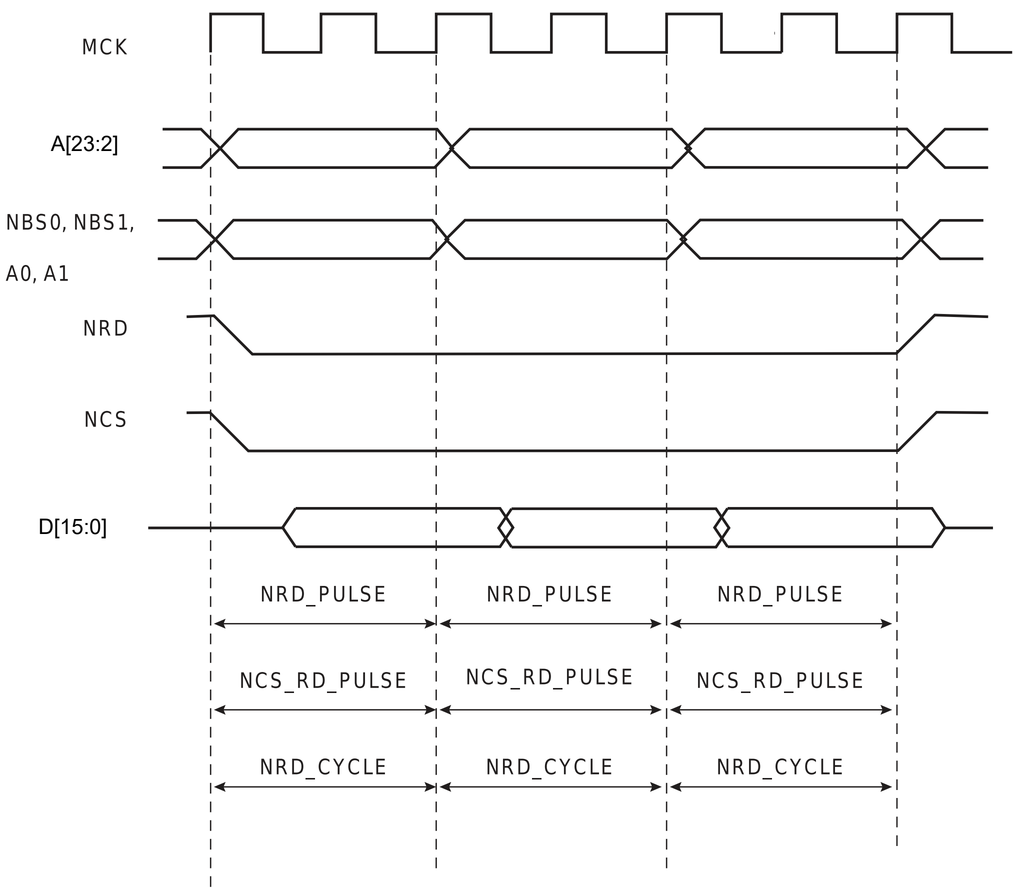
NRD CYCLE, NCS RD SETUP, and NCS RD PULSE implicitly define the NCS RD HOLD value as:

NCS RD HOLD = NRD CYCLE - NCS RD SETUP - NCS RD PULSE

40.7.4.5. Null Delay Setup and Hold

If null setup and hold parameters are programmed for NRD and/or NCS, NRD and NCS remain active continuously in case of consecutive read cycles in the same memory. This is shown in the following figure.

Figure 40-8. No Setup, No Hold on NRD and NCS Read Signals



40.7.4.6. Null Pulse

Programming a null pulse is not permitted. The pulse must be at least set to 1. A null value leads to unpredictable behavior.

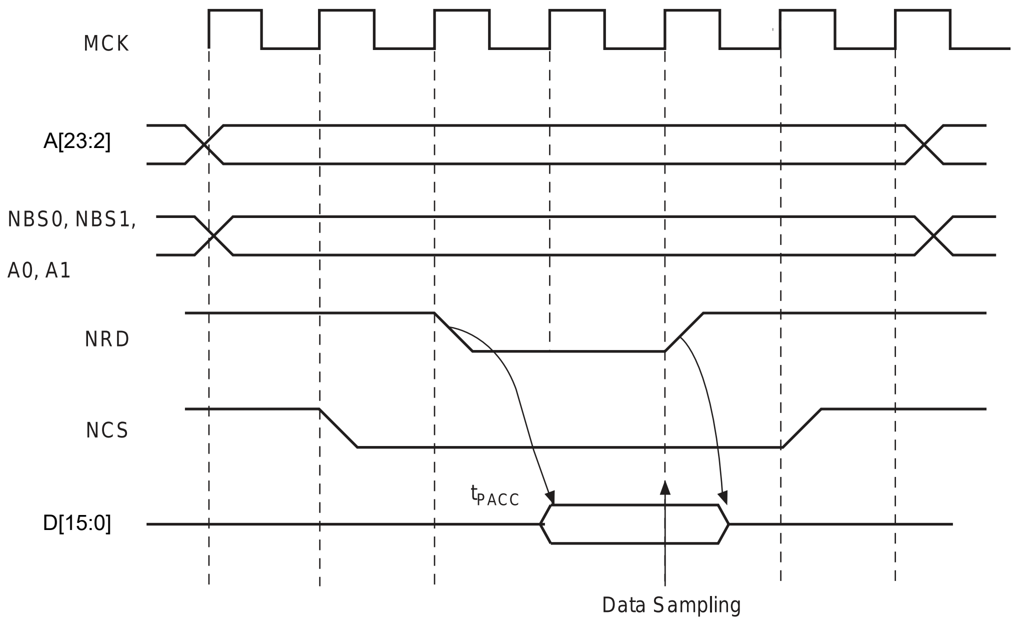
40.7.5. Read Mode

As NCS and NRD waveforms are defined independently of one other, the EBI needs to know when the read data is available on the data bus. The SMC does not compare NCS and NRD timings to know which signal rises first. The READ MODE bit in the SMC MODE register of the corresponding chip select indicates which signal of NRD and NCS controls the read operation.

40.7.5.1. Read is Controlled by NRD (EBI MODE.READ MODE = 1)

The following figure shows the waveforms of a read operation of a typical asynchronous RAM. The read data is available tPACC after the falling edge of NRD, and turns to 'Z' after the rising edge of NRD. In this case, SMC MODE.READ MODE must be set to 1 (read is controlled by NRD), to indicate that data is available with the rising edge of NRD. The EBI samples the read data internally on the rising edge of Host Clock that generates the rising edge of NRD, whatever the programmed waveform of NCS may be.

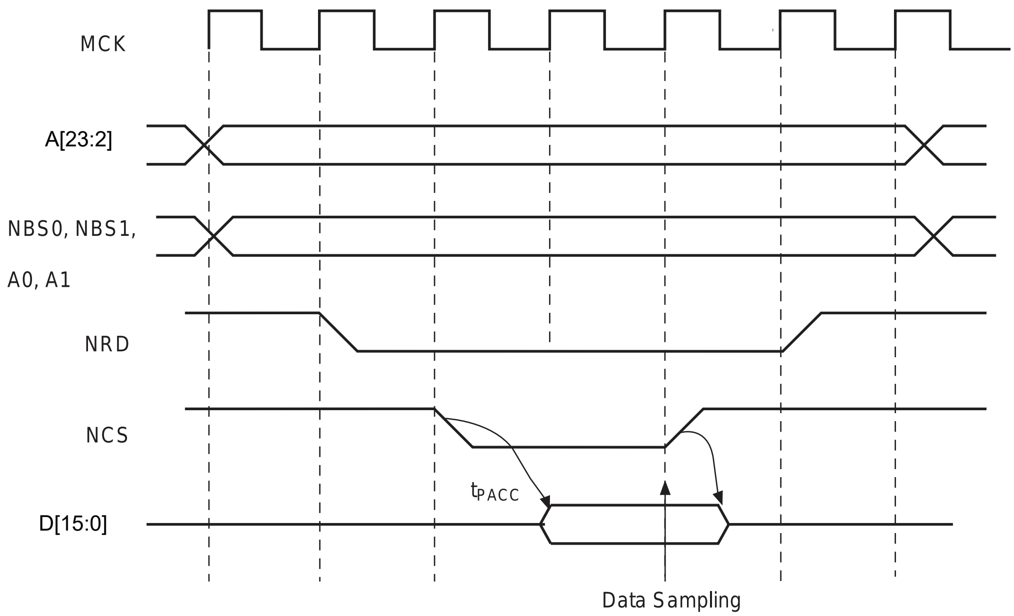
Figure 40-9. SMC MODE.READ MODE=1: Data is Sampled by EBI before the Rising Edge of NRD



40.7.5.2. Read is Controlled by NCS (EBI M0DE.READ M0DE = 0)

The following figure shows the typical read cycle of an LCD module. The read data is valid tPACC after the falling edge of the NCS signal and remains valid until the rising edge of NCS. Data must be sampled when NCS is raised. In this case, the EBI MODE.READ MODE must be set to 0 (read is controlled by NCS): the EBI internally samples the data on the rising edge of Host Clock that generates the rising edge of NCS, whatever the programmed waveform of NRD may be.

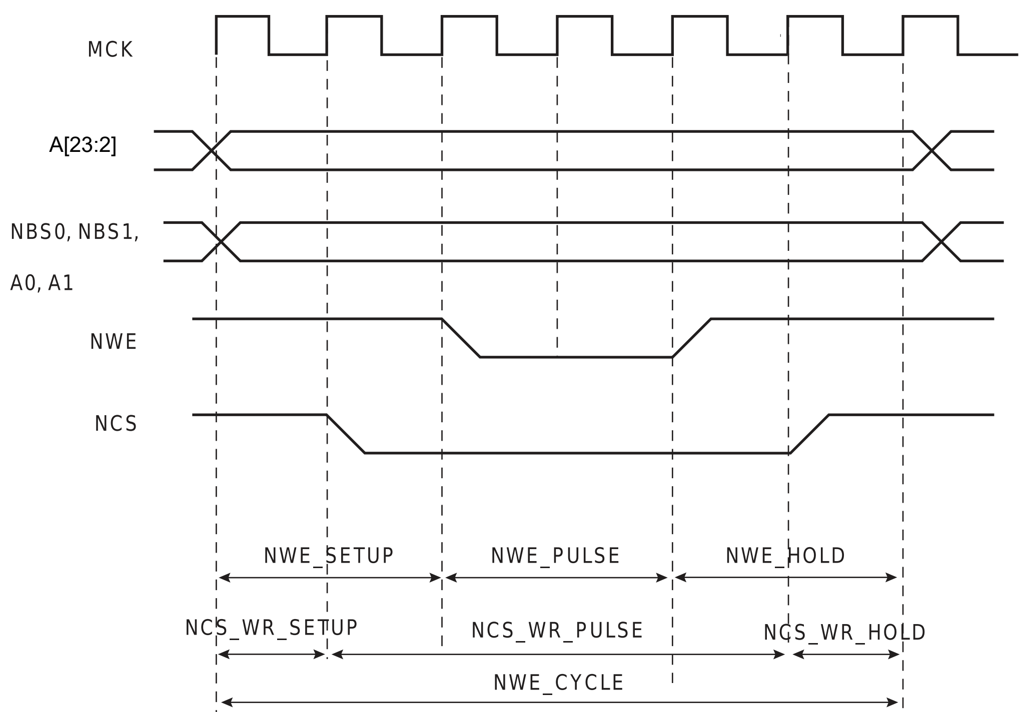
Figure 40-10. SMC MODE.READ MODE = 0: Data is Sampled by EBI before the Rising Edge of NCS



40.7.6. Write Waveforms

The write protocol is similar to the read protocol. It is depicted in the following figure. The write cycle starts with the address setting on the memory address bus.

Figure 40-11. Write Cycle



40.7.6.1. NWE Waveforms

The NWE signal is characterized by a setup timing, a pulse width and a hold timing.

- **NWE SETUP:** The NWE setup time is defined as the setup of address and data before the NWE falling edge

- **NWEPULSE:** The NWE pulse length is the time between NWE falling edge and NWE rising edge
- **NWE HOLD:** The NWE hold time is defined as the hold time of address and data after the NWE rising edge

40.7.6.2. NCS Waveforms

The NCS signal waveforms in write operation are not the same that those applied in read operations, but are separately defined:

- **ncs wr setup:** The NCS setup time is defined as the setup time of address before the NCS falling edge
- **ncs wr pulse:** The NCS pulse length is the time between NCS falling edge and NCS rising edge
- **ncs wr hold:** The NCS hold time is defined as the hold time of address after the NCS rising edge

40.7.6.3. Write Cycle

The write cycle time is defined as the total duration of the write cycle; that is, from the time where address is set on the address bus to the point where address may change. The total write cycle time is defined as:

NWE CYCLE = NWE SETUP + NWE PULSE + NWE HOLD,

as well as

NWE CYCLE = NCS WR SETUP + NCS WR PULSE + NCS WR HOLD

All NWE and NCS (write) timings are defined separately for each chip select as an integer number of Host Clock cycles. The NWE CYCLE field is common to both the NWE and NCS signals, therefore the timing period is of the same duration.

NWE CYCLE, NWE SETUP, and NWE PULSE implicitly define the NWE HOLD value as:

NWE HOLD = NWE CYCLE - NWE SETUP - NWE PULSE

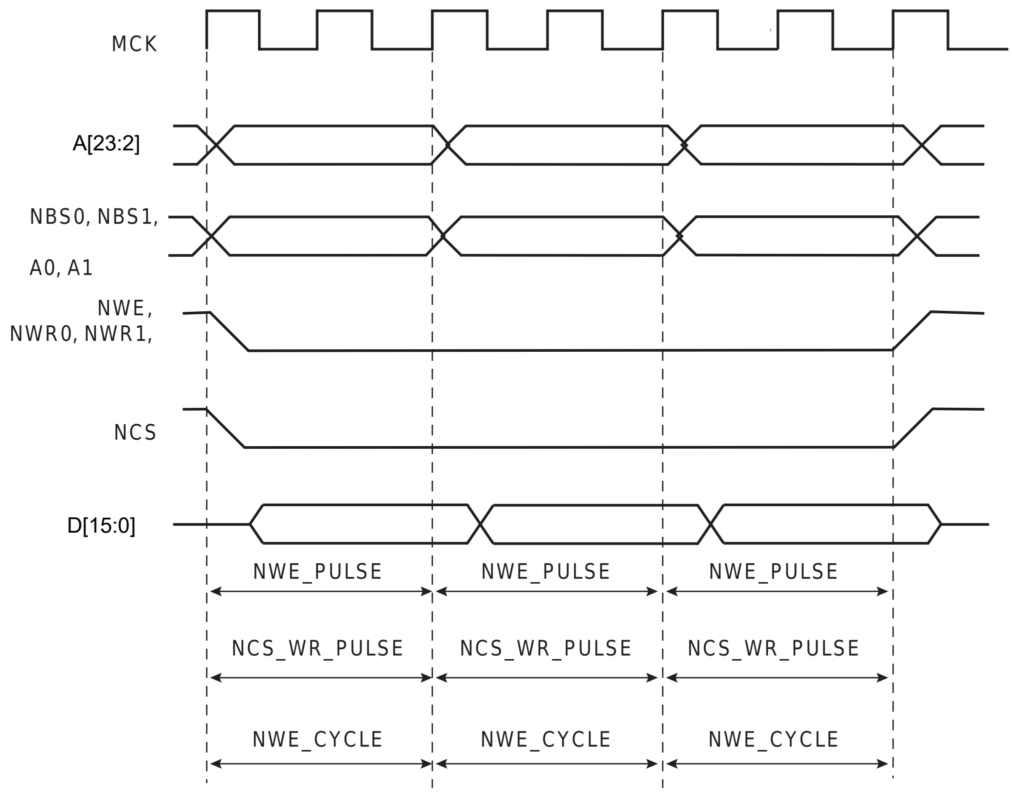
NWE CYCLE, NCS WR SETUP, and NCS WR PULSE implicitly define the NCS WR HOLD value as:

NCS WR HOLD = NWE CYCLE - NCS WR SETUP - NCS WR PULSE

40.7.6.4. Null Delay Setup and Hold

If null setup parameters are programmed for NWE and/or NCS, NWE and/or NCS remain active continuously in case of consecutive write cycles in the same memory (see the following figure). However, for devices that perform write operations on the rising edge of NWE or NCS, such as SRAM, either a setup or a hold must be programmed.

Figure 40-12. Null Setup and Hold Values of NCS and NWE in Write Cycle



40.7.6.5. Null Pulse

Programming null pulse is not permitted. Pulse must be at least set to 1. A null value leads to unpredictable behavior.

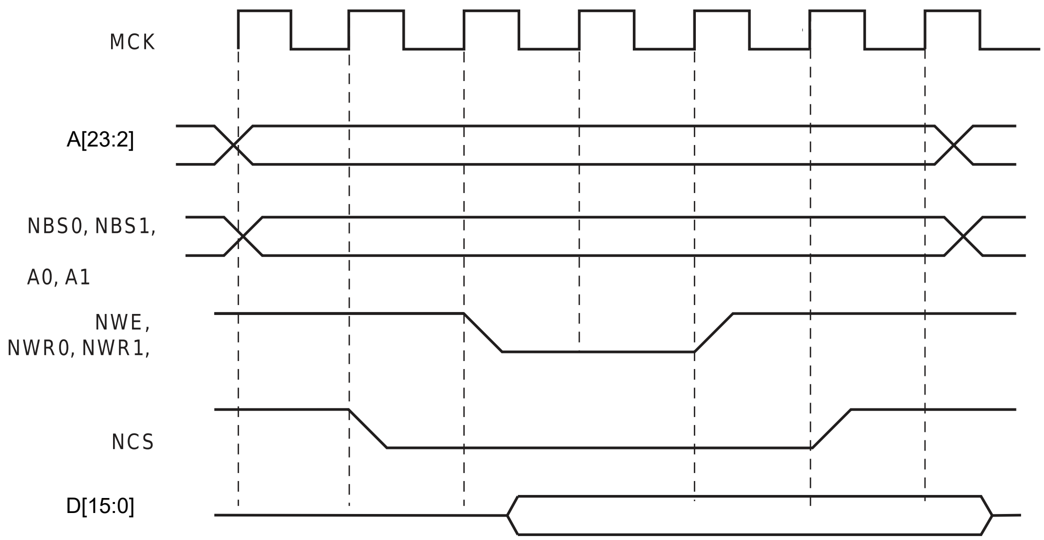
40.7.7. Write Mode

The bit WRITE MODE in the SMC MODE register of the corresponding chip select indicates which signal controls the write operation.

40.7.7.1. Write is Controlled by NWE (SMC.M0DE.WRITE M0DE = 1)

The following figure shows the waveforms of a write operation with SMC MODE.WRITE MODE set. The data is put on the bus during the pulse and hold steps of the NWE signal. The internal data buffers are switched to Output mode after the NWE SETUP time, and remain in this mode until the end of the write cycle, regardless of the programmed waveform on NCS.

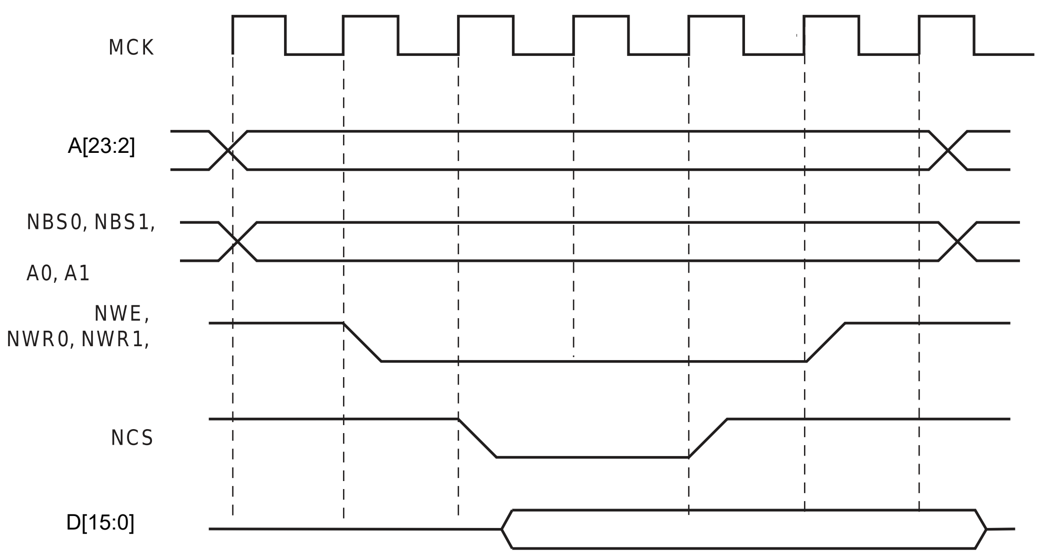
Figure 40-13. SMC MODE.WRITE MODE = 1. Write Operation is Controlled by NWE



40.7.7.2. Write is Controlled by NCS (SMC.MODE.WRITE MODE = 0)

The following figure shows the waveforms of a write operation with SMC MODE.WRITE MODE cleared. The data is put on the bus during the pulse and hold steps of the NCS signal. The internal data buffers are switched to Output mode after the NCS WR SETUP time, and remain in this mode until the end of the write cycle, regardless of the programmed waveform on NWE.

Figure 40-14. WRITE MODE = 0. Write Operation is Controlled by NCS



40.7.8. Register Write Protection

To prevent any single software error that may corrupt EBI behavior, the registers listed below can be write-protected by setting the WPEN bit in the SMC Write Protection Mode register (SMC WPMR).

If a write access in a write-protected register is detected, the WPVS flag in the SMC Write Protection Status register (SMC WPSR) is set and the field WPVSRC indicates in which register the write access has been attempted.

The WPVS flag is automatically cleared after reading the SSMC WPSR. The following registers can be write-protected:

- SETUP Register
- PULSE Register
- CYCLE Register
- MODE Register
- Off-chip Memory Scrambling Register

40.7.9. Coding Timing Parameters

All timing parameters are defined for one chip select and are grouped together in one register according to their type.

The SMC SETUP register groups the definition of all setup parameters:

- NRDSETUP
- NCS RDSETUP
- NWE SETUP
- NCS WR SETUP

The SMC PULSE register groups the definition of all pulse parameters:

- NRDPULSE
- NCS RDPULSE
- NWE PULSE
- NCS WR PULSE

The SMC CYCLE register groups the definition of all cycle parameters:

- NRDCYCLE
- NWE CYCLE

The following table shows how the timing parameters are coded and their permitted range.

Table 40-3. Coding and Range of Timing Parameters

Coded Value	Number of Bits	Effective Value	Permitted Range	
			Coded Value	Effective Value
setup [5:0]	6	128 x setup[5] + setup[4:0]	0 :: 31	0 :: 128+31
pulse [6:0]	7	256 x pulse[6] + pulse[5:0]	0 :: 63	0 :: 256+63
cycle [8:0]	9	256 x cycle[8:7] + cycle[6:0]	0 :: 127	0 :: 256+127
				0 :: 512+127
				0 :: 768+127

40.7.10. Usage Restriction

The SMC does not check the validity of the user-programmed parameters. If the sum of SETUP and PULSE parameters is larger than the corresponding CYCLE parameter, this leads to unpredictable behavior of the SMC.

- For read operations:

Null but positive setup and hold of address and NRD and/or NCS can not be guaranteed at the memory interface because of the propagation delay of theses signals through external logic and pads. If positive setup and hold values must be verified, then it is strictly recommended to program non-null values so as to cover possible skews between address, NCS and NRD signals.

- For write operations:

If a null hold value is programmed on NWE, the EBI can guarantee a positive hold of address and NCS signal after the rising edge of NWE. This is true for SMC MODE.WRITE MODE = 1 only. See *Early Read Wait State*.

- For read and write operations:

A null value for pulse parameters is forbidden and may lead to unpredictable behavior.

In read and write cycles, the setup and hold time parameters are defined in reference to the address bus. For external devices that require setup and hold time between NCS and NRD signals (read), or between NCS and NWE signals (write), these setup and hold times must be converted into setup and hold times in reference to the address bus.

40.7.11. Automatic Wait States

Under certain circumstances, the EBI automatically inserts idle cycles between accesses to avoid bus contention or operation conflict.

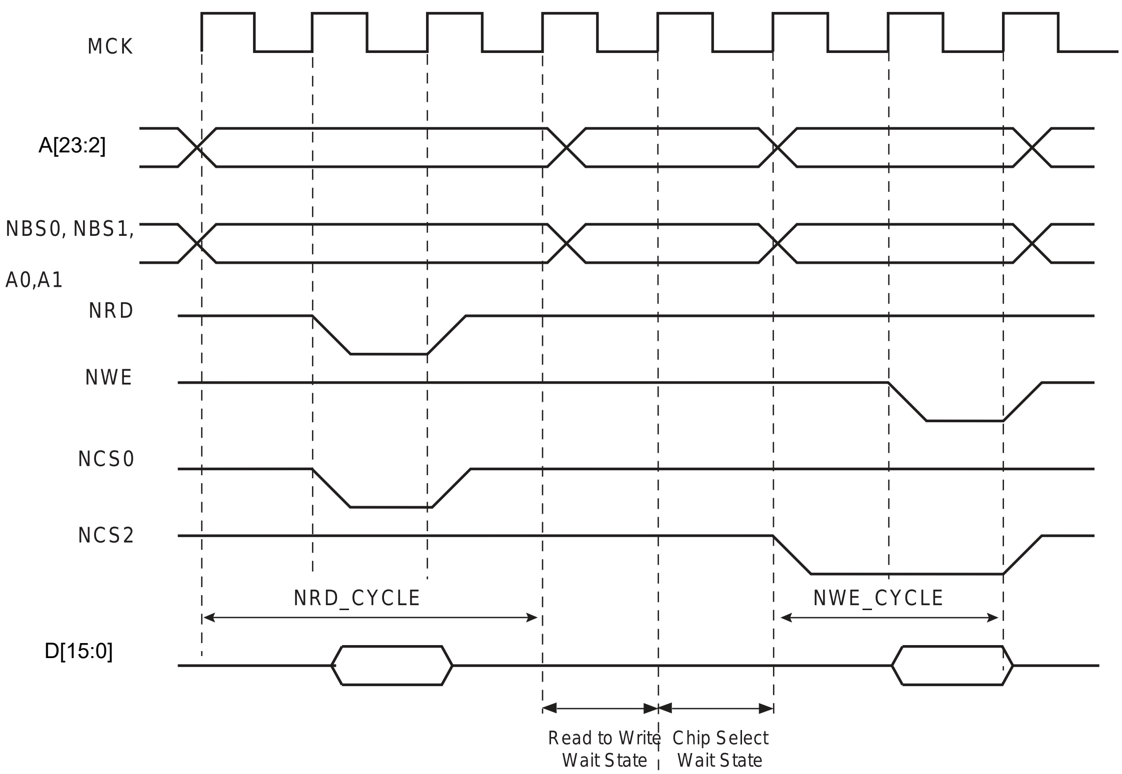
40.7.11.1. Chip Select Wait States

The EBI always inserts an idle cycle between two transfers on separate chip selects. This idle cycle ensures that there is no bus contention between the deactivation of one device and the activation of the next one.

During chip select wait state, all control lines are turned inactive: NBS0 to NBS1, NWR0 to NWR1, NCS[0..NB_CS-1], NRD lines are all set to 1.

The following figure illustrates a chip select wait state between access on Chip Select 0 and Chip Select 2.

Figure 40-15. Chip Select Wait State Between a Read Access on NSC0 and a Write Access on NCS2



40.7.11.2. Early Read Wait State

In some cases, the EBI inserts a wait state cycle between a write access and a read access to allow time for the write cycle to end before the subsequent read cycle begins. This wait state is not generated in addition to a chip select wait state. The early read cycle therefore only occurs between a write and read access to the same memory device (same chip select).

An early read wait state is automatically inserted if at least one of the following conditions is valid:

- If the write controlling signal has no hold time and the read controlling signal has no setup time
- In NCS Write controlled mode (SMC MODE.WRITE MODE = 0), if there is no hold timing on the NCS signal and the NCS RD SETUP parameter is set to 0, regardless of the Read mode. The write operation must end with a NCS rising edge. Without an Early Read Wait State, the write operation could not complete properly.
- In NWE controlled mode (SMC MODE.WRITE MODE = 1) and if there is no hold timing (NWE HOLD = 0), the feedback of the write control signal is used to control address, data, and chip select lines. If the external write control signal is not inactivated as expected due to load capacitances, an Early Read Wait State is inserted and address, data and control signals are maintained one more cycle.

Figure 40-16. Early Read Wait State: Write with no Hold Followed by Read with no Setup

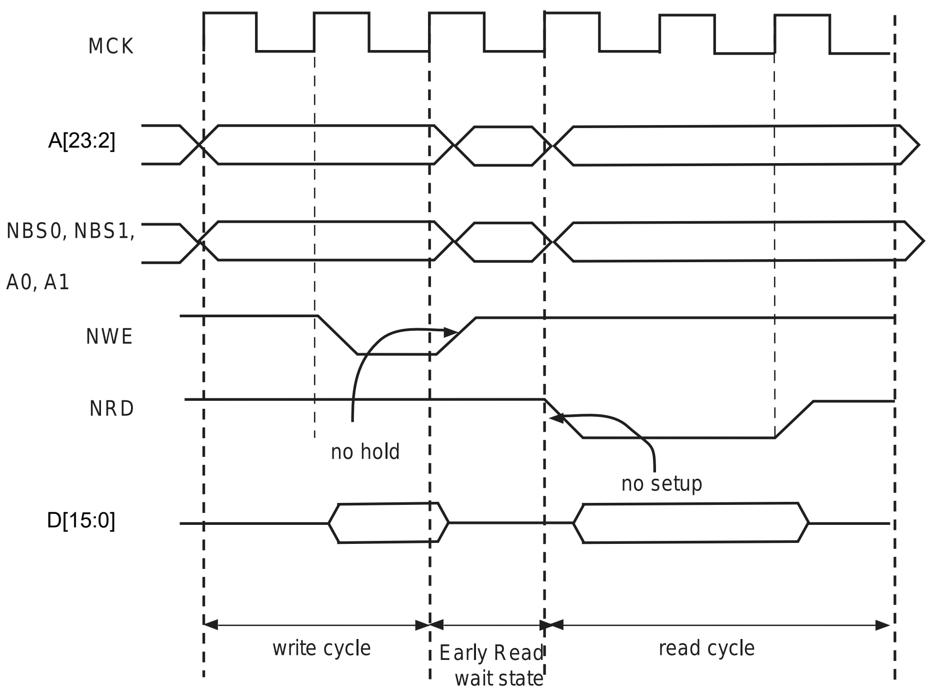


Figure 40-17. Early Read Wait State: NCS Controlled Write with no Hold Followed by a Read with no NCS Setup

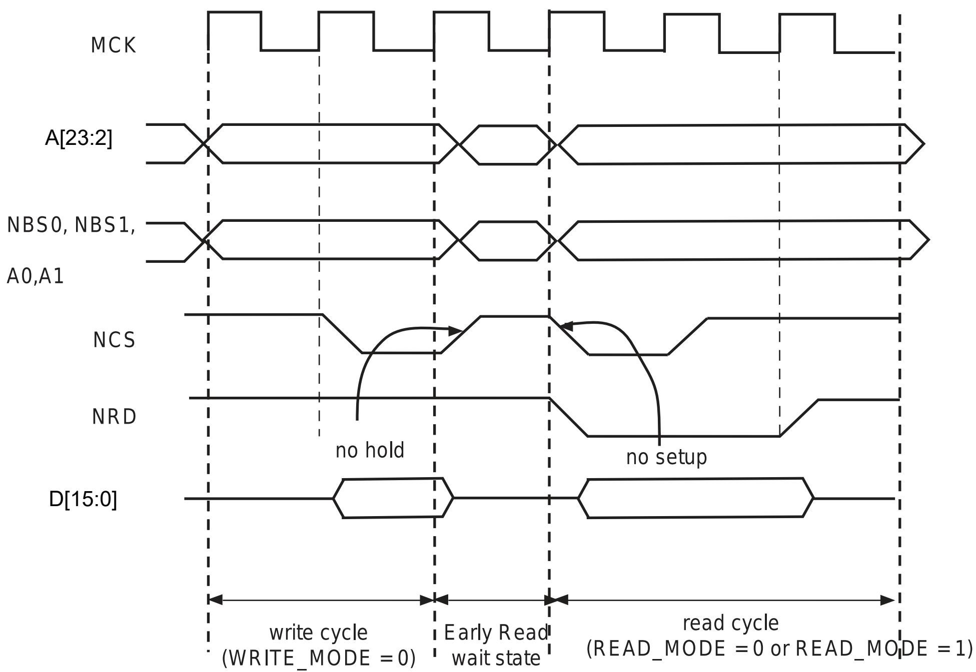
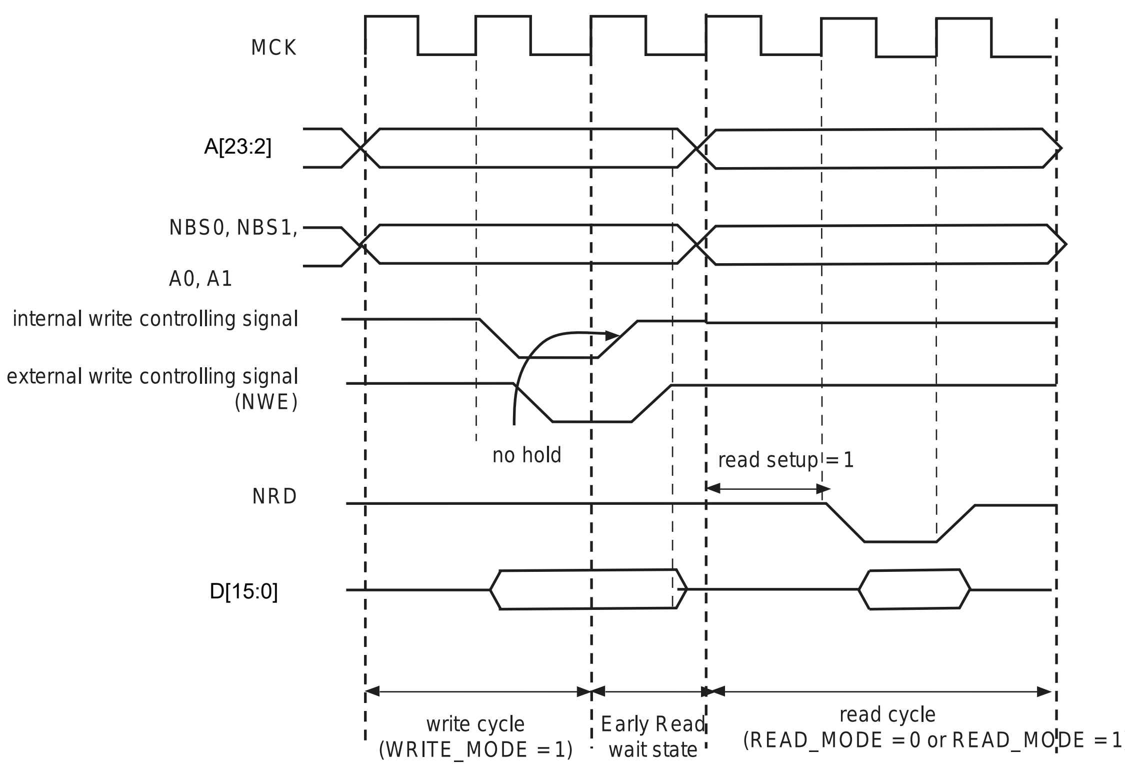


Figure 40-18. Early Read Wait State: NWE-Controlled Write with no Hold Followed by a Read with One Set-up Cycle



40.7.11.3. Reload User Configuration Wait State

The user may change any of the configuration parameters by writing the SMC user interface.

When detecting that a new user configuration has been written in the user interface, the EBI inserts a wait state before starting the next access. This "reload user configuration wait state" is used by the EBI to load the new set of parameters to apply to next accesses.

The reload configuration wait state is not applied in addition to the chip select wait state. If accesses before and after re-programming the user interface are made to different devices (chip selects), then one single chip select wait state is applied.

On the other hand, if accesses before and after writing the user interface are made to the same device, a reload configuration wait state is inserted, even if the change does not concern the current chip select.

40.7.11.3.1. User Procedure

To insert a reload configuration wait state, the SMC detects a write access to any SMC MODE register of the user interface. If the user only modifies timing registers (SMC SETUP, SMC PULSE, SMC CYCLE registers) in the user interface, he must validate the modification by writing the SMC MODE, even if no change was made on the mode parameters.

The user must not change the configuration parameters of an EBI chip select (Setup, Pulse, Cycle, Mode) if accesses are performed on this CS during the modification. Any change of the chip select parameters, while fetching the code from a memory connected on this CS, may lead to unpredictable behavior. The instructions used to modify the parameters of an EBI chip select can be executed from the internal RAM or from a memory connected to another CS.

40.7.11.3.2. Slow Clock Mode Transition

A reload configuration wait state is also inserted when the Slow Clock mode is entered or exited, after the end of the current transfer. For more information, see *Slow Clock Mode*.

40.7.11.4. Read to Write Wait State

Due to an internal mechanism, a wait cycle is always inserted between consecutive read and write SMC accesses.

This wait cycle is referred to as a read to write wait state in this document.

This wait cycle is applied in addition to chip select and reload user configuration wait states when they are to be inserted.

40.7.12. Data Float Wait States

Some memory devices are slow to release the external bus. For such devices, it is necessary to add wait states (data float wait states) after a read access:

- Before starting a read access to a different external memory
- Before starting a write access to the same device or to a different external one

The data float output time (t_{DF}) for each external memory device is programmed in the SMC MODE.TDF CYCLES field for the corresponding chip select. The value of

SMC MODE.TDF CYCLES indicates the number of data float wait cycles (between 0 and 15) before the external device releases the bus, and represents the time allowed for the data output to go to high impedance after the memory is disabled.

Data float wait states do not delay internal memory accesses. Therefore, a single access to an external memory with long t_{DF} will not slow down the execution of a program from internal memory.

The data float wait states management depends on SMC MODE.READ MODE and the SMC MODE.TDF MODE fields for the corresponding chip select.

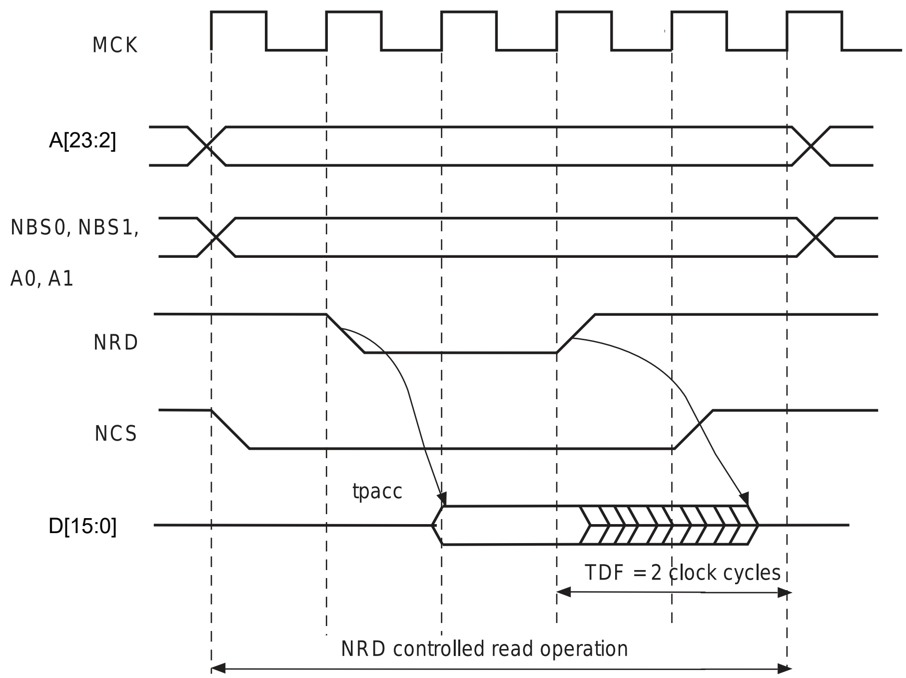
40.7.12.1. SMC M0DE.READ M0DE

Setting SMC MODE.READ MODE to 1 indicates to the SMC that the NRD signal is responsible for turning off the tri-state buffers of the external memory device. The Data Float Period then begins after the rising edge of the NRD signal and lasts SMC MODE.TDF CYCLES MCK cycles.

When the read operation is controlled by the NCS signal (SMC MODE.READ MODE = 0), the TDF field gives the number of MCK cycles during which the data bus remains busy after the rising edge of NCS.

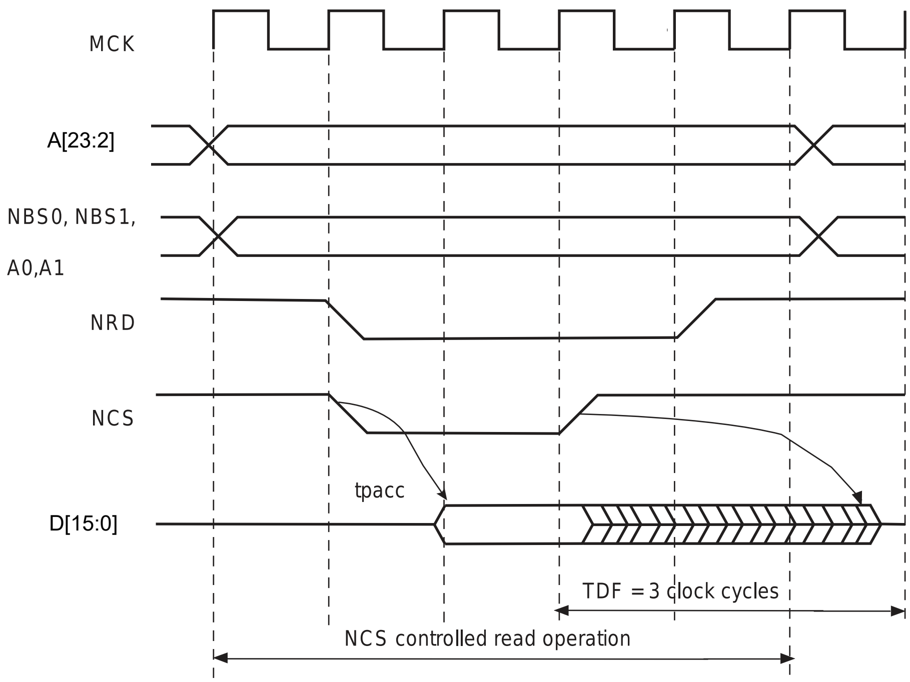
The following figure illustrates the Data Float Period in NRD-controlled mode (SMC MODE.READ MODE =1), assuming a data float period of 2 cycles (SMC MODE.TDF CYCLES = 2).

Figure 40-19. TDF Period in NRD Controlled Read Access (TDF = 2)



The following figure shows the read operation when controlled by NCS (SMC MODE.READ MODE = 0) and SMC MODE.TDF CYCLES = 3.

Figure 40-20. TDF, (i.e. Data Float Time), Period in NCS Controlled Read 0peration (TDF = 3)



40.7.12.2. TDF, (i.e. Data Float Time), Optimization Enabled (SMC M0DE.TDF M0DE = 1)

When the TDF_MODE of the SMC_MODE register is set to 1 (TDF data float optimization is enabled),the SMC takes advantage of the setup period of the next access to optimize the number of wait states cycle to insert.

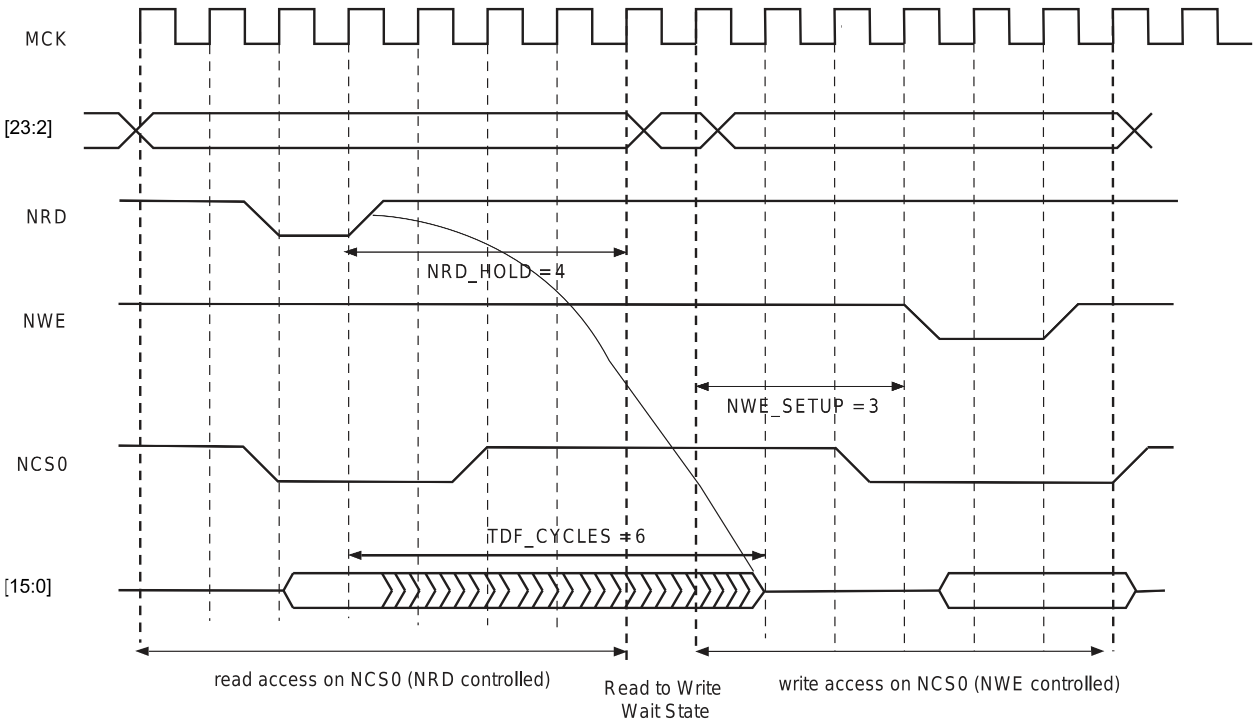
The following figure shows a read access controlled by NRD, followed by a write access controlled by NWE, on Chip Select 0. Chip Select 0 has been programmed with:

NRD_HOLD = 4; READ_MODE = 1 (NRD controlled)

NWE_SETUP = 3; WRITE_MODE = 1 (NWE controlled)

TDF_CYCLES = 6; TDF_MODE= 1 (optimization enabled).

Figure 40-21. TDF, (i.e. Data Float Time), Optimization: No TDF Wait States are Inserted if the TDF Period is over when the Next Access Begins



40.7.12.3. TDF, (i.e. Data Float Time), Optimization Disabled (SMC M0DE.TDF M0DE = 0)

When optimization is disabled, TDF Wait states are inserted at the end of the read transfer, so that the data float period is ended when the second access begins. If the hold period of the read1 controlling signal overlaps the data float period, no additional TDF Wait states will be inserted.

The following figures illustrate the cases:

- read access followed by a read access on another Chip Select
- read access followed by a write access on another Chip Select
- read access followed by a write access on the same Chip Select, with no TDF optimization

Figure 40-22. TDF Optimization Disabled (TDF Mode = 0): TDF Wait States Between Two Read Accesses on Different Chip Selects

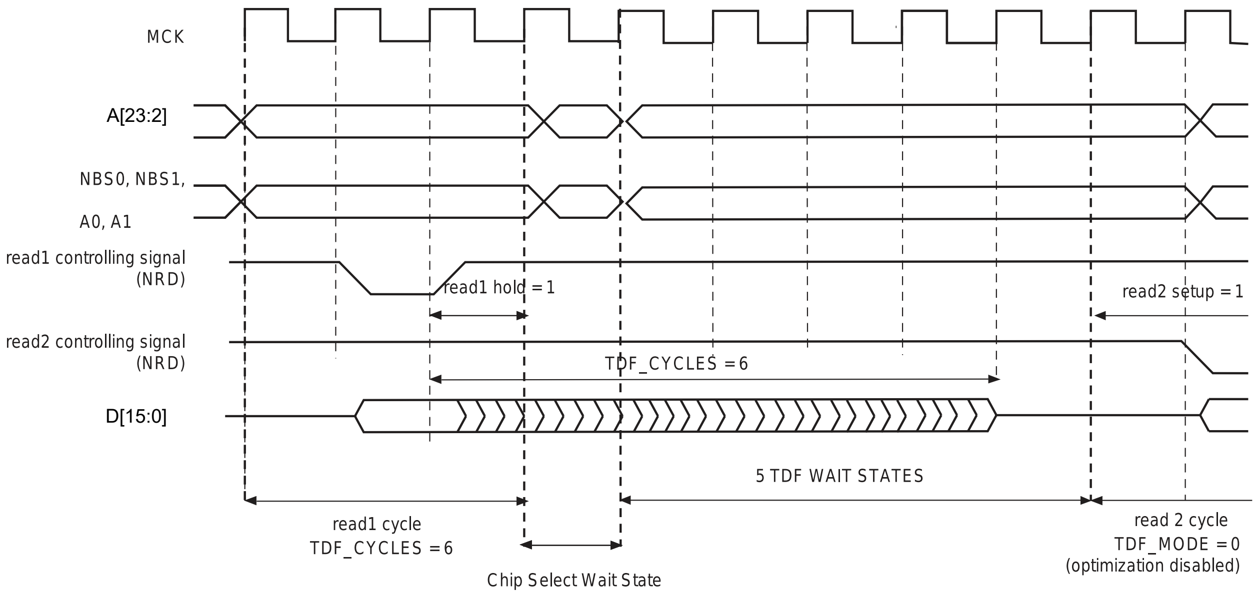


Figure 40-23. TDF Mode = 0: TDF Wait States Between a Read and a Write Access on Different Chip Selects

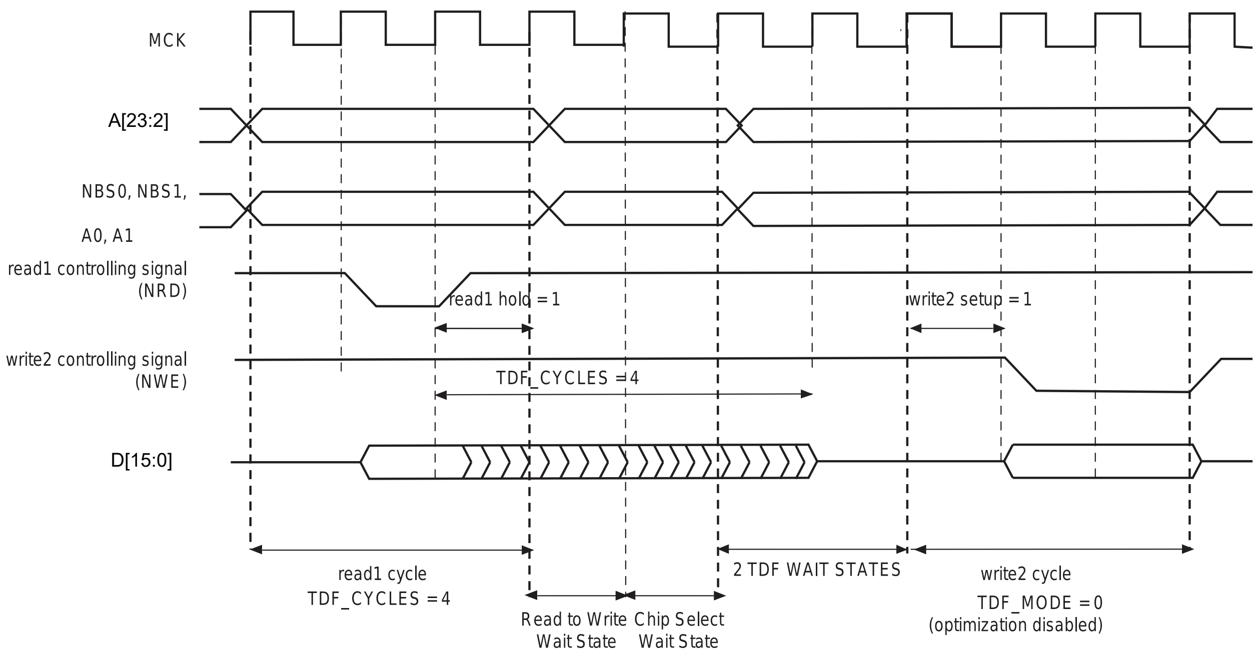
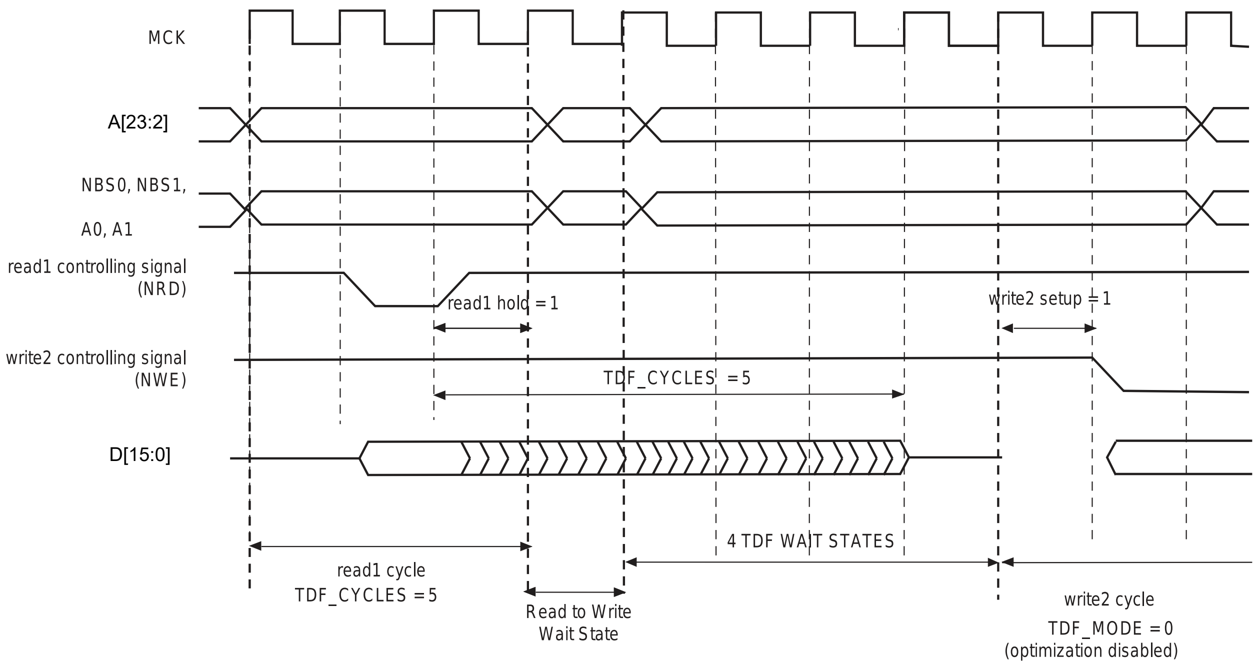


Figure 40-24. TDF Mode = 0: TDF Wait States Between Read and Write Accesses on the Same Chip Select



40.7.13. External Wait

Any access can be extended by an external device using the NWAIT input signal of the EBI. The EXNW_MODE field of the SMC_MODE register on the corresponding chip select must be set to either to ‘10’ (frozen mode) or ‘11’ (ready mode). When the EXNW_MODE is set to ‘00’ (disabled), the NWAIT signal is simply ignored on the corresponding chip select. The NWAIT signal delays the read or write operation in regards to the read or write controlling signal, depending on the read and write modes of the corresponding chip select.

40.7.13.1. Restriction

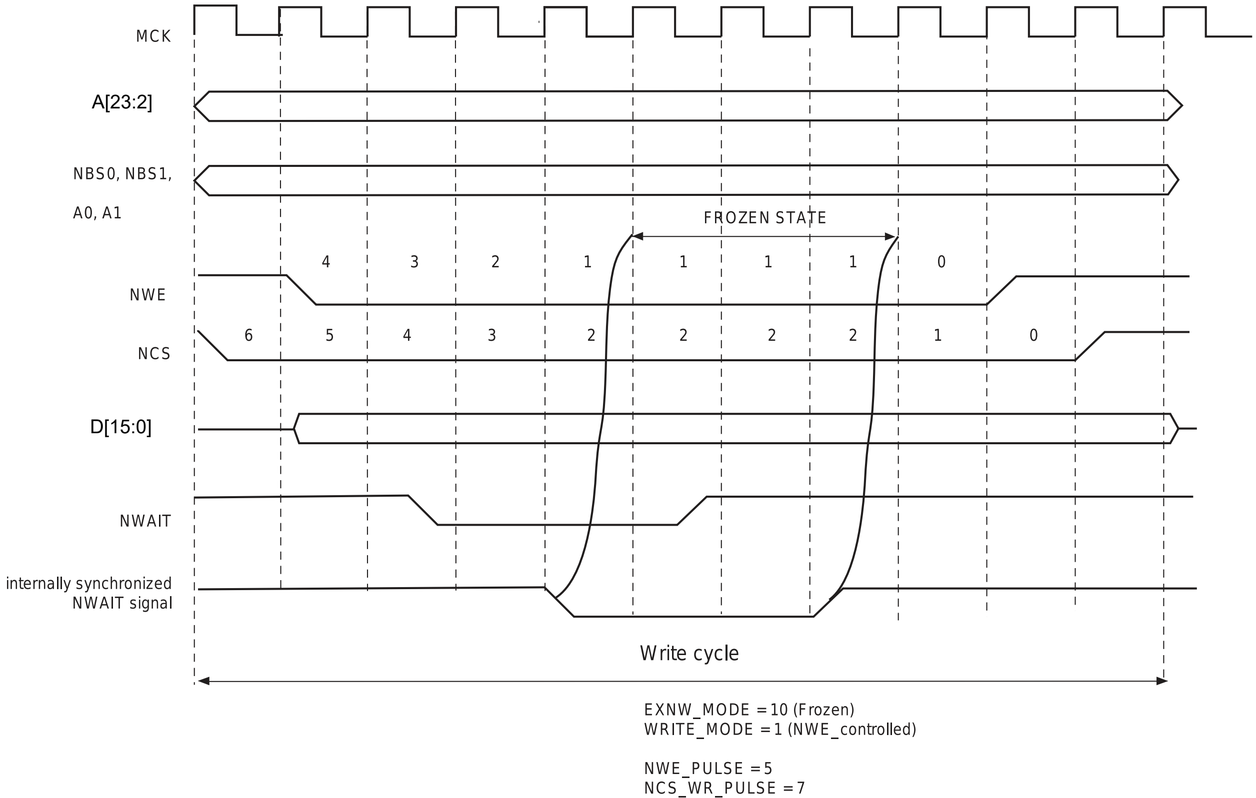
When one of the EXNW_MODE is enabled, it is mandatory to program at least one hold cycle for the read/write controlling signal. For that reason, the NWAIT signal cannot be used in Page Mode, or in Slow Clock Mode .

The NWAIT signal is assumed to be a response of the external device to the read/write request of the SMC. Then NWAIT is examined by the SMC only in the pulse state of the read or write controlling signal. The assertion of the NWAIT signal outside the expected period has no impact on SMC behavior

40.7.13.2. Frozen Mode

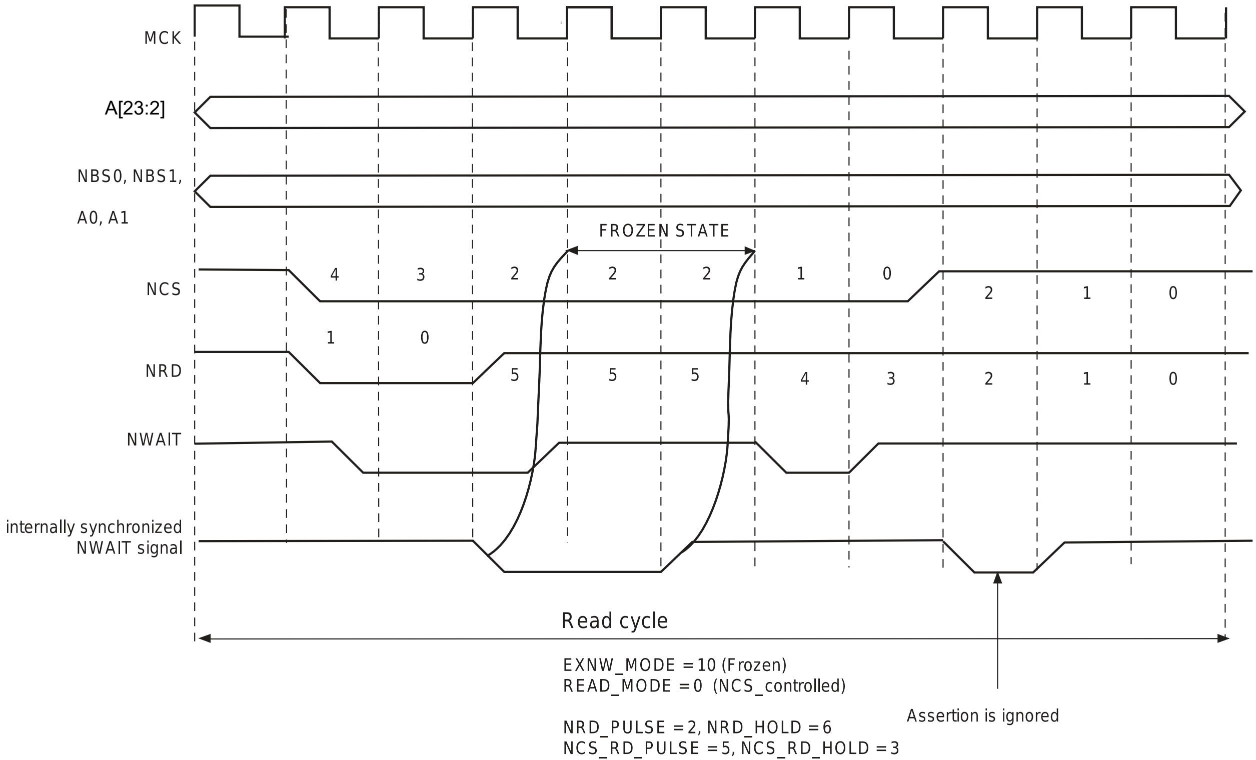
When the external device asserts the NWAIT signal (active low), and after internal synchronization of this signal, the EBI state is frozen, i.e., SMC internal counters are frozen, and all control signals remain unchanged. When the resynchronized NWAIT signal is deasserted, the EBI completes the access, resuming the access from the point where it was stopped. This mode must be selected when the external device uses the NWAIT signal to delay the access and to freeze the EBI.

Figure 40-25. Write Access with NWAIT Assertion in Frozen Mode (SMC M0DE.EXNW M0DE = 10)



The assertion of the NWAIT signal outside the expected period is ignored as illustrated in the following figure.

Figure 40-26. Read Access with NWAIT Assertion in Frozen Mode (SMC M0DE.EXNW M0DE = 10)



40.7.13.3. Ready Mode

In Ready mode (EXNW MODE = 11), the EBI behaves differently. Normally, the EBI begins the access by down counting the setup and pulse counters of the read/write controlling signal. In the last cycle of the pulse phase, the resynchronized NWAIT signal is examined.

If asserted, the EBI suspends the access. After deassertion, the access is completed and the hold step of the access is performed.

This mode must be selected when the external device uses deassertion of the NWAIT signal to indicate its ability to complete the read or write operation.

If the NWAIT signal is deasserted before the end of the pulse, or asserted after the end of the pulse of the controlling read/write signal, it will have impact on the access length.

Figure 40-27. NWAIT Assertion in Write Access: Ready Mode (SMC M0DE.EXNW M0DE = 11)

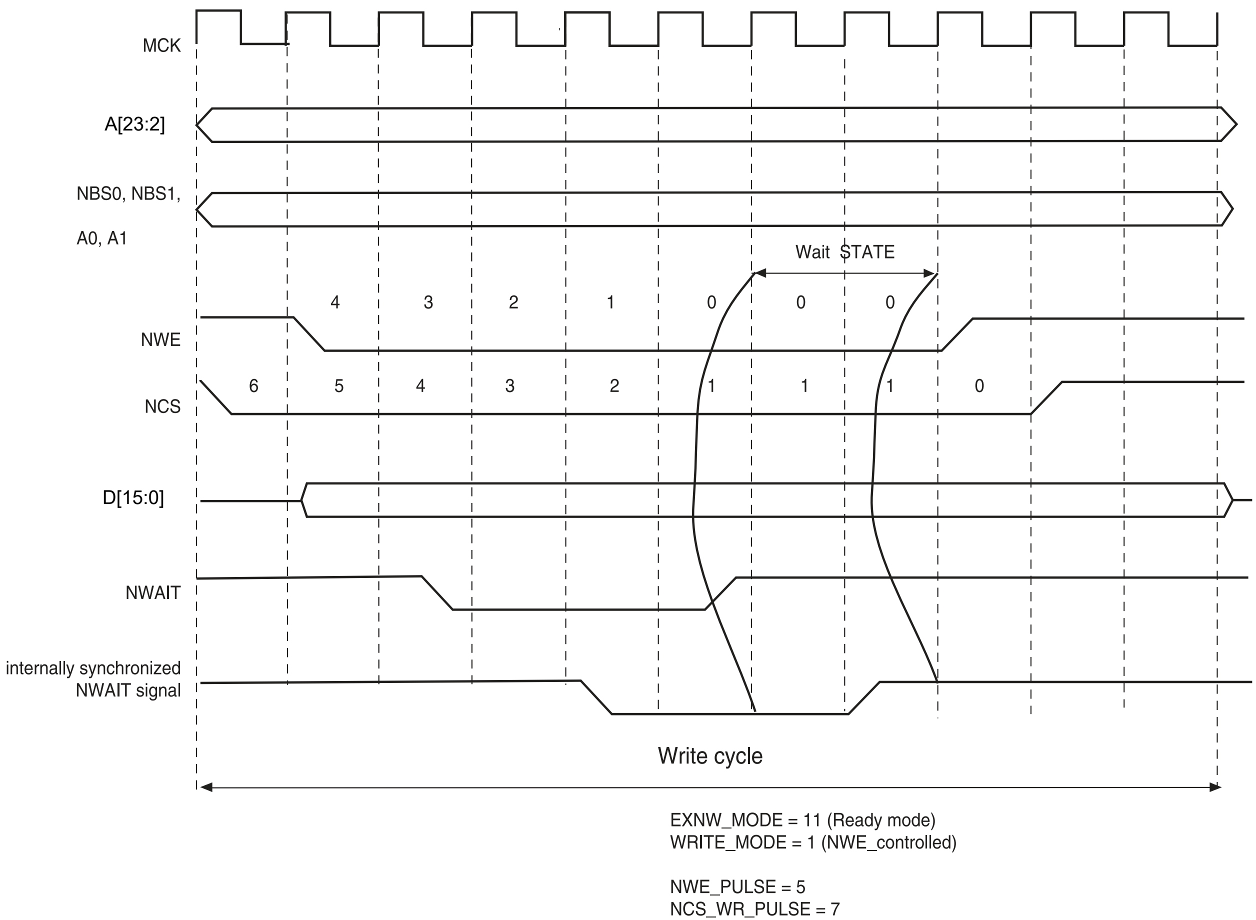
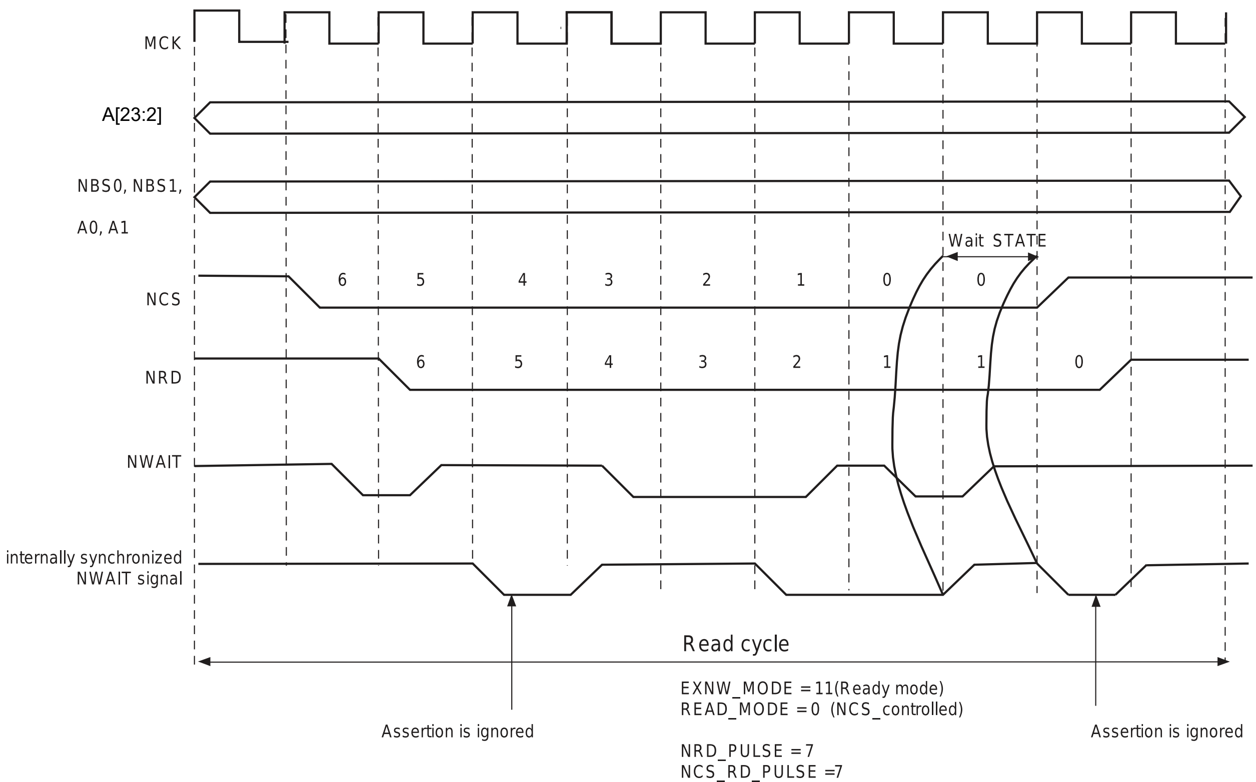


Figure 40-28. NWAIT Assertion in Read Access: Ready Mode (SMC M0DE.EXNW M0DE = 11)



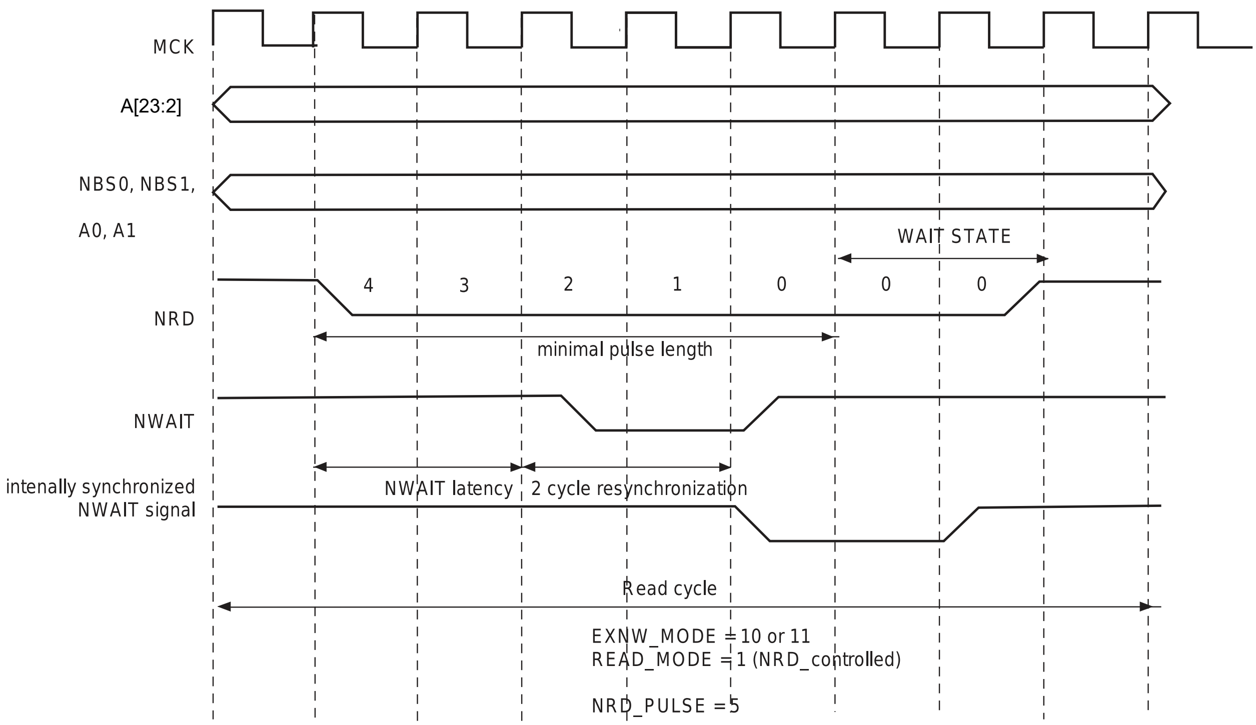
40.7.13.4. NWAIT Latency and Read/Write Timings

There may be a latency between the assertion of the read/write controlling signal and the assertion of the NWAIT signal by the device. The programmed pulse length of the read/write controlling signal must be at least equal to this latency plus the 2 cycles of resynchronization + one cycle. Otherwise, the SMC may enter the hold state of the access without detecting the NWAIT signal assertion. This is true in Frozen mode as well as in Ready mode. This is illustrated in the following figure.

When EXNW MODE is enabled (ready or frozen), the user must program a pulse length of the read and write controlling signal of at least:

Minimal pulse length = NWAIT latency + 2 resynchronization cycles + 1 cycle

Figure 40-29. NWAIT Latency



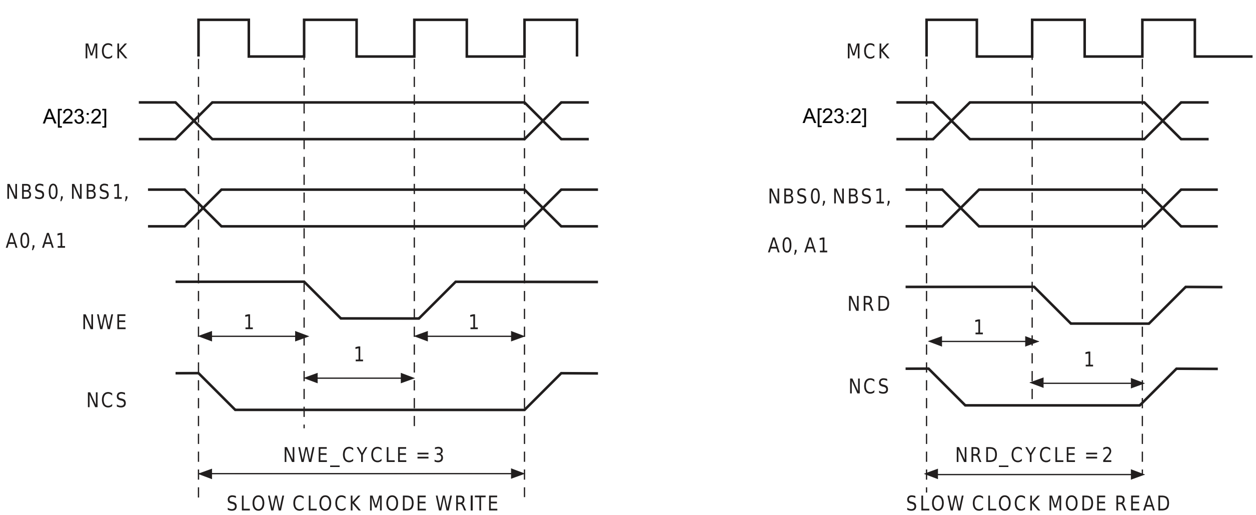
40.7.14. Slow Clock Mode

The SMC is able to automatically apply a set of "Slow clock mode" read/write waveforms when an internal signal driven by the Power Management Controller is asserted because MCK has been turned to a very slow clock rate (typically 32 kHz clock rate). In this mode, the user-programmed waveforms are ignored and the Slow clock mode waveforms are applied. This mode is provided so as to avoid reprogramming the User Interface with appropriate waveforms at a very slow clock rate. When activated, the Slow clock mode is active on all chip selects.

40.7.14.1. Slow Clock Mode Waveforms

The following figure illustrates the read and write operations in Slow Clock mode. They are valid on all Chip Selects.

Figure 40-30. Read/Write Cycles in Slow Clock Mode



The following table indicates the value of read and write parameters in Slow Clock mode.

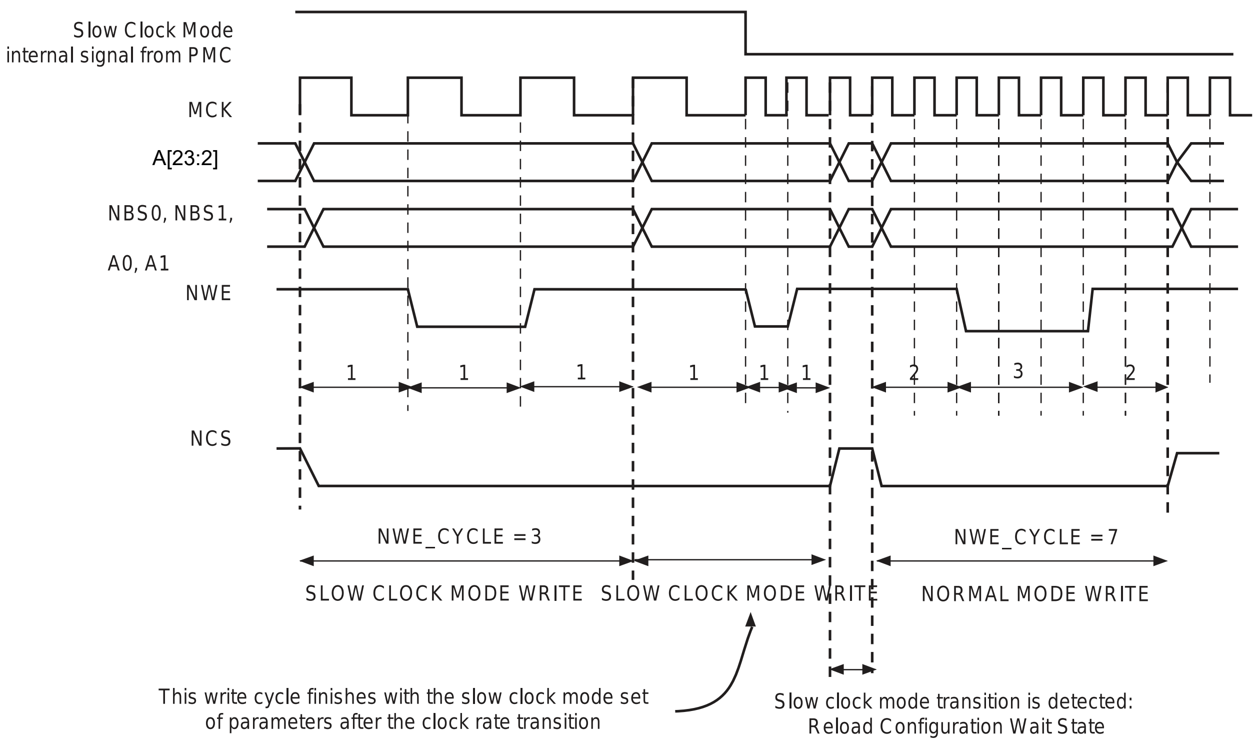
Table 40-4. Read and Write Timing Parameters in Slow Clock Mode

Read Parameters	Duration (cycles)	Write Parameters	Duration (cycles)
NRD SETUP	1	NWE SETUP	1
NRD PULSE	1	NWE PULSE	1
NCS RD SETUP	0	NCS WR SETUP	0
NCS RD PULSE	2	NCS WR PULSE	3
NRD CYCLE	2	NWE CYCLE	3

40.7.14.2. Switching from (to) Slow Clock Mode to (from) Normal Mode

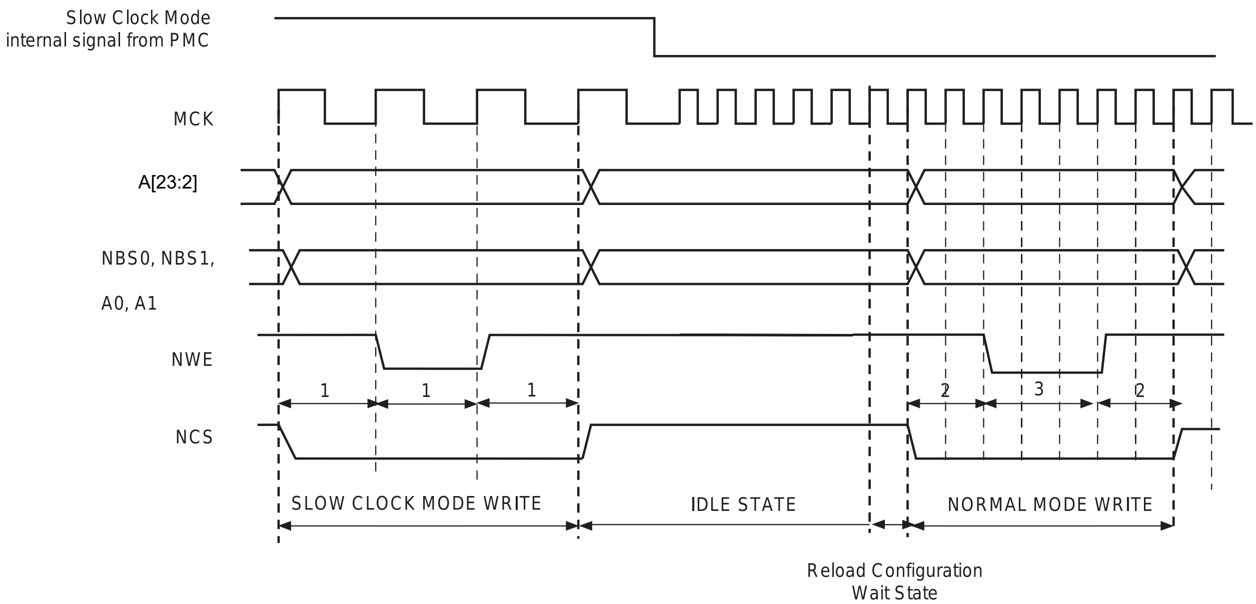
When switching from Slow clock mode to Normal mode, the current Slow clock mode transfer is completed at a high clock rate, with the set of Slow clock mode parameters. The external device may not be fast enough to support such timings.

Figure 40-31. Clock Rate Transition Occurs while the SMC is Performing a Write Operation



The following figure illustrates the recommended procedure to switch from one mode to the other.

Figure 40-32. Recommended Procedure to Switch from Slow Clock Mode to Normal Mode or from Normal Mode to Slow Clock Mode



40.7.15. Asynchronous Page Mode

The EBI supports asynchronous burst reads in Page mode, provided that the Page mode is enabled (SMC MODE.PMEN =1). The page size must be configured in the SMC MODE register (PS field) to 4, 8, 16 or 32 bytes.

The page defines a set of consecutive bytes into memory. A 4-byte page (resp. 8-, 16-, 32-byte page) is always aligned to 4-byte boundaries (resp. 8-, 16-, 32-byte boundaries) of memory. The MSB of data address defines the address of the page in memory, the LSB of address define the address of the data in the page as detailed in the following table.

With Page mode memory devices, the first access to one page (t_{pa}) takes longer than the subsequent accesses to the page (t_{sa}) as shown in *Page Mode Read Protocol*. When in Page mode, the EBI enables the user to define different read timings for the first access within one page, and next accesses within the page.

Table 40-5. Page Address and Data Address within a Page

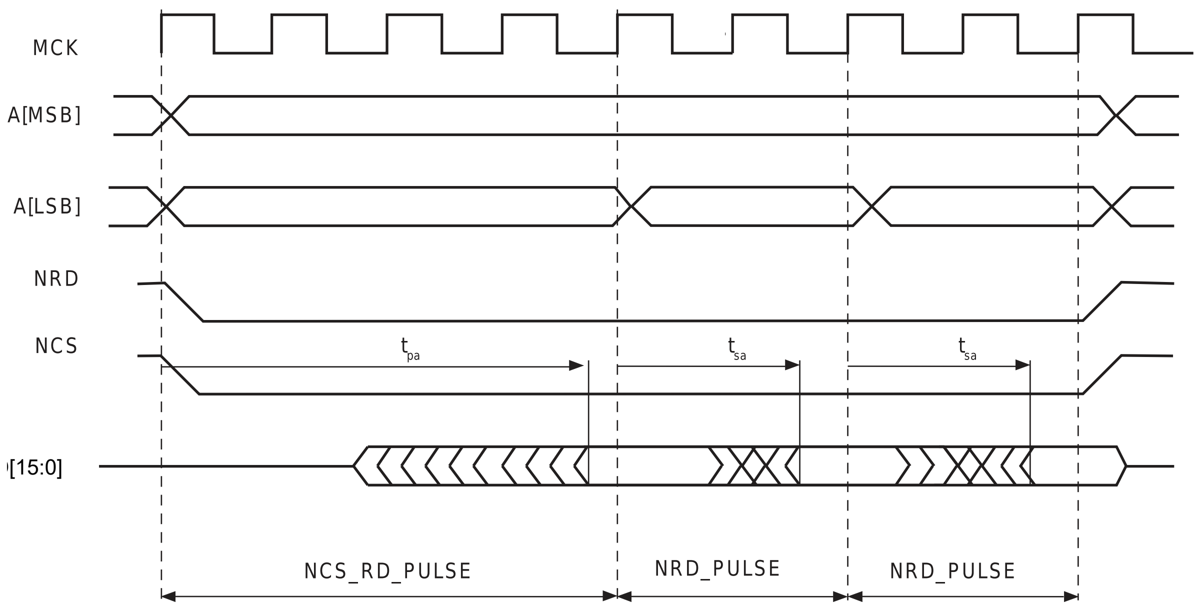
Page Size	Page Address (see Note)	Data Address in the Page
4 bytes	A[23:2]	A[1:0]
8 bytes	A[23:3]	A[2:0]
16 bytes	A[23:4]	A[3:0]
32 bytes	A[23:5]	A[4:0]

Note: "A" denotes the address bus of the memory device.

40.7.15.1. Protocol and Timings in Page Mode

The following figure shows the NRD and NCS timings in Page mode access.

Figure 40-33. Page Mode Read Protocol



The NRD and NCS signals are held low during all read transfers, whatever the programmed values of the setup and hold timings in the User Interface may be. Moreover, the NRD and NCS timings are identical. The pulse length of the first access to the page is defined with the NCS RD PULSE field of the

SMC PULSE register. The pulse length of subsequent accesses within the page are defined using the NRD PULSE parameter.

In Page mode, the programming of the read timings is described in the following table:

Table 40-6. Programming of Read Timings in Page Mode

Parameter	Value	Definition
READ MODE	'x'	No impact.
NCS RD SETUP	'x'	No impact.
NCS RD PULSE	t_{pa}	Access time of first access to the page.
NRD SETUP	'x'	No impact.
NRD PULSE	t_{sa}	Access time of subsequent accesses in the page.
NRD CYCLE	'x'	No impact.

The SMC does not check the coherency of timings. It will always apply the NCS RD PULSE timings as page access timing (t_{pa}) and the NRD PULSE for accesses to the page (t_{sa}), even if the programmed value for t_{pa} is shorter than the programmed value for t_{sa}.

40.7.15.2. Byte Access Type in Page Mode

The byte access type (BAT) configuration remains active in page mode. For 16-bit or 32-bit page mode devices that require byte selection signals, write a 0 to the BAT bit in the SMC Mode Register (SMC_MODE) to select the byte select access type.

40.7.15.3. Page Mode Restriction

The page mode is not compatible with the use of the NWAIT signal. Using the page mode and the NWAIT signal may lead to unpredictable behavior.

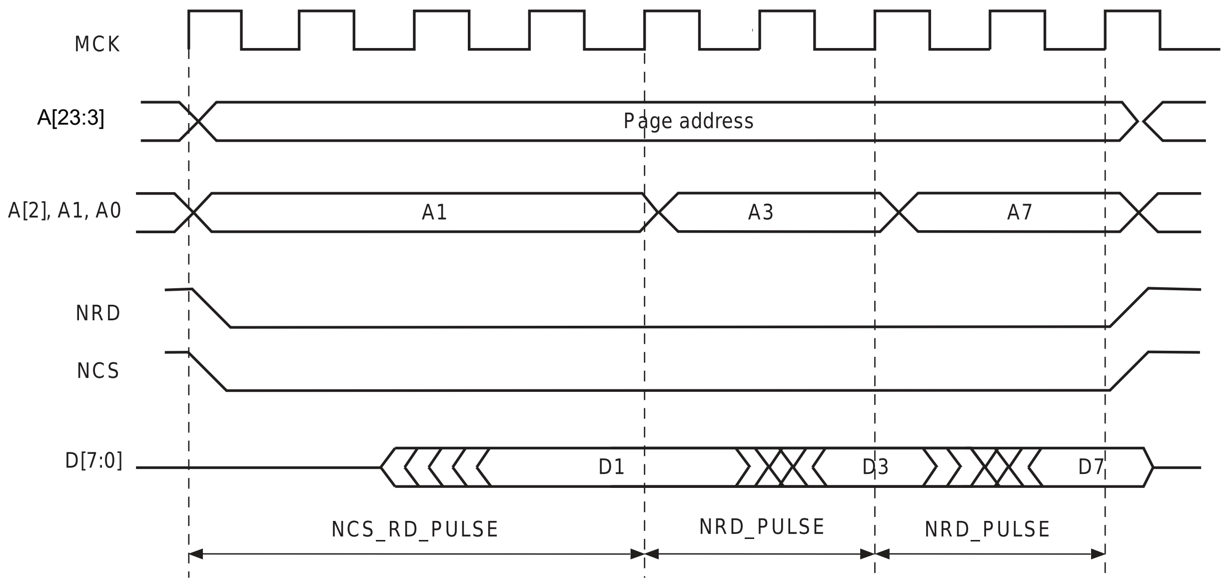
40.7.15.4. Sequential and Non-Sequential Accesses

If the chip select and the MSB of addresses are identical, then the current access lies in the same page as the previous one, and no page break occurs.

Using this information, all data within the same page, sequential or not sequential, are accessed with a minimum access time (t_{sa}). The following figure illustrates access to an 8-bit memory device in Page mode, with 8-byte pages. Access to D1 causes a page access with a long access time (t_{pa}). Accesses to D3 and D7, though they are not sequential accesses, only require a short access time (t_{sa}).

If the MSB of addresses are different, the EBI performs the access of a new page. In the same way, if the chip select is different from the previous access, a page break occurs. If two sequential accesses are made to the Page mode memory, but separated by an other internal or external peripheral access, a page break occurs on the second access because the chip select of the device was deasserted between both accesses.

Figure 40-34. Access to Non-Sequential Data within the Same Page



40.7.16. Register Write Protection

To prevent any single software error from corrupting EBI behavior, certain registers in the address space can be write-protected by setting the WPEN bit in the SMC Write Protection Mode Register (SMC_WPMR).

If a write access to a write-protected register is detected, the WPVS flag in the SMC Write Protection Status Register (SMC_WPSR) is set and the field WPVSRC indicates the register in which the write access has been attempted.

The WPVS bit is automatically cleared after reading the SMC_WPSR. The following registers can be write-protected:

- Setup Register
- Pulse Register
- Cycle Register
- Mode Register

40.8. Register Summary

For descriptions and definitions of both Register and bitfield properties, refer to Register Properties.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	SETUP0	31:24			NCS RD SETUP[5:0]					
		23:16			NRD SETUP[5:0]					
		15:8			NCS WR SETUP[5:0]					
		7:0			NWE SETUP[5:0]					
0x04	SETUP1	31:24			NCS RD SETUP[5:0]					
		23:16			NRD SETUP[5:0]					
		15:8			NCS WR SETUP[5:0]					
		7:0			NWE SETUP[5:0]					
0x04	PULSE0	31:24		NCS RD PULSE[6:0]						
		23:16		NRD PULSE[6:0]						
		15:8		NCS WR PULSE[6:0]						
		7:0		NWE PULSE[6:0]						
0x08	SETUP2	31:24			NCS RD SETUP[5:0]					
		23:16			NRD SETUP[5:0]					
		15:8			NCS WR SETUP[5:0]					
		7:0			NWE SETUP[5:0]					
0x08	PULSE1	31:24		NCS RD PULSE[6:0]						
		23:16		NRD PULSE[6:0]						
		15:8		NCS WR PULSE[6:0]						
		7:0		NWE PULSE[6:0]						
0x08	CYCLE0	31:24								NRD CYCLE[8]
		23:16	NRD CYCLE[7:0]							
		15:8								NWE CYCLE[8]
		7:0	NWE CYCLE[7:0]							
0x0C	SETUP3	31:24			NCS RD SETUP[5:0]					
		23:16			NRD SETUP[5:0]					
		15:8			NCS WR SETUP[5:0]					
		7:0			NWE SETUP[5:0]					
0x0C	PULSE2	31:24		NCS RD PULSE[6:0]						
		23:16		NRD PULSE[6:0]						
		15:8		NCS WR PULSE[6:0]						
		7:0		NWE PULSE[6:0]						
0x0C	CYCLE1	31:24								NRD CYCLE[8]
		23:16	NRD CYCLE[7:0]							
		15:8								NWE CYCLE[8]
		7:0	NWE CYCLE[7:0]							
0x0C	MODE0	31:24			PS[1:0]					PMEN
		23:16				TDF MODE	TDF CYCLES[3:0]			
		15:8				DBW				BAT
		7:0			EXNW MODE[1:0]				WRITE MODE	READ MODE
0x10	PULSE3	31:24		NCS RD PULSE[6:0]						
		23:16		NRD PULSE[6:0]						
		15:8		NCS WR PULSE[6:0]						
		7:0		NWE PULSE[6:0]						
0x10	CYCLE2	31:24								NRD CYCLE[8]
		23:16	NRD CYCLE[7:0]							
		15:8								NWE CYCLE[8]
		7:0	NWE CYCLE[7:0]							
0x10	MODE1	31:24			PS[1:0]					PMEN
		23:16				TDF MODE	TDF CYCLES[3:0]			
		15:8				DBW				BAT
		7:0			EXNW MODE[1:0]				WRITE MODE	READ MODE

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x14	CYCLE3	31:24								NRD CYCLE[8]
		23:16	NRD CYCLE[7:0]							
		15:8								NWE CYCLE[8]
		7:0	NWE CYCLE[7:0]							
0x14	MODE2	31:24			PS[1:0]					PMEN
		23:16				TDF MODE	TDF CYCLES[3:0]			
		15:8				DBW				BAT
		7:0			EXNW MODE[1:0]				WRITE MODE	READ MODE
0x18	MODE3	31:24			PS[1:0]					PMEN
		23:16				TDF MODE	TDF CYCLES[3:0]			
		15:8				DBW				BAT
		7:0			EXNW MODE[1:0]				WRITE MODE	READ MODE
0x1C ... 0xE3	Reserved									
0xE4	WPMR	31:24	WPKEY [23:16]							
		23:16	WPKEY [15:8]							
		15:8	WPKEY [7:0]							
		7:0								WPEN
0xE8	WPSR	31:24								
		23:16	WPVSRC [15:8]							
		15:8	WPVSRC [7:0]							
		7:0								WPVS

40.8.1. SMC Setup Register

Name: SETUP
Offset: 0x00 + n*0x04 [n=0..3]
Reset: 0x01010101
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the SMC Write Protection Mode Register.

Table 40-7. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
			NCS RD SETUP[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			1	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
			NRD SETUP[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			1	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
			NCS WR SETUP[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			1	0	0	0	0	0

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

Bits 29:24 – NCS RD SETUP[5:0] NCS Setup Length in READ Access
In read access, the NCS signal setup length is defined as:
NCS setup length = (128* NCS RD SETUP[5] + NCS RD SETUP[4:0]) clock cycles

Bits 21:16 – NRD SETUP[5:0] NRD Setup Length
The NRD signal setup length is defined in clock cycles as:
NRD setup length = (128* NRD SETUP[5] + NRD SETUP[4:0]) clock cycles

Bits 13:8 – NCS WR SETUP[5:0] NCS Setup Length in WRITE Address
In write access, the NCS signal setup length is defined as:
NCS setup length = (128* NCS WR SETUP[5] + NCS WR SETUP[4:0]) clock cycles

Bits 5:0 – NWE SETUP[5:0] NWE Setup Length
The NWE signal setup length is defined as:
NWE setup length = (128* NWE SETUP[5] + NWE SETUP[4:0]) clock cycles

40.8.2. SMC Pulse Register

Name: PULSE
Offset: 0x04 + n*0x04 [n=0..3]
Reset: 0x01010101
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the SMC Write Protection Mode Register.

Table 40-8. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
		NCS RD PULSE[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		1	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
		NRD PULSE[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		1	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
		NCS WR PULSE[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		1	0	0	0	0	0	0

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

Bits 30:24 – NCS RD PULSE[6:0] NCS Pulse Length in READ Access
In standard read access, the NCS signal pulse length is defined as:
NCS pulse length = (256* NCS RD PULSE[6] + NCS RD PULSE[5:0]) clock cycles The NCS pulse length must be at least 1 clock cycle.
In Page mode read access, the NCS RD PULSE parameter defines the duration of the first access to one page.

Bits 22:16 – NRD PULSE[6:0] NRD Pulse Length
In standard read access, the NRD signal pulse length is defined in clock cycles as: NRD pulse length = (256* NRD PULSE[6] + NRD PULSE[5:0]) clock cycles
The NRD pulse length must be at least 1 clock cycle.
In Page mode read access, the NRD PULSE parameter defines the duration of the subsequent accesses in the page.

Bits 14:8 – NCS WR PULSE[6:0] NCS Pulse Length in WRITE Address
In write access, the NCS signal pulse length is defined as:
NCS pulse length = (256* NCS WR PULSE[6] + NCS WR PULSE[5:0]) clock cycles The NCS pulse length must be at least 1 clock cycle.

Bits 6:0 – NWE PULSE[6:0] NWE Pulse Length
The NWE signal pulse length is defined as:
NWE pulse length = (256* NWE PULSE[6] + NWE PULSE[5:0]) clock cycles
The NWE pulse length must be at least 1 clock cycle.

40.8.3. SMC Cycle Register

Name: CYCLE
Offset: 0x08 + n*0x04 [n=0..3]
Reset: 0x03030303
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the SMC Write Protection Mode Register.

Table 40-9. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
								NRD CYCLE[8]
Access								R/W
Reset								0

Bit	23	22	21	20	19	18	17	16
	NRD CYCLE[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
								NWE CYCLE[8]
Access								R/W
Reset								0

Bit	7	6	5	4	3	2	1	0
	NWE CYCLE[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 24:16 – NRD CYCLE[8:0] Total Read Cycle Length
The total read cycle length is the total duration in clock cycles of the read cycle. It is equal to the sum of the setup, pulse and hold steps of the NRD and NCS signals. It is defined as:
Read cycle length = (NRD CYCLE[8:7]*256 + NRD CYCLE[6:0]) clock cycles

Bits 8:0 – NWE CYCLE[8:0] Total Write Cycle Length
The total write cycle length is the total duration in clock cycles of the write cycle. It is equal to the sum of the setup, pulse and hold steps of the NWE and NCS signals. It is defined as:
Write cycle length = (NWE CYCLE[8:7]*256 + NWE CYCLE[6:0]) clock cycles

40.8.4. SMC Mode Register

Name: MODE
Offset: 0x0C + n*0x04 [n=0..3]
Reset: 0x00
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the SMC Write Protection Mode Register. The user must confirm the SMC configuration by writing any one of the SMC MODE registers.

Table 40-10. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
			PS[1:0]					PMEN
Access			R/W	R/W				R/W
Reset			0	0				0

Bit	23	22	21	20	19	18	17	16
				TDF MODE	TDF CYCLES[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
				DBW				BAT
Access				R/W				R/W
Reset				0				0

Bit	7	6	5	4	3	2	1	0
			EXNW MODE[1:0]				WRITE MODE	READ MODE
Access			R/W	R/W			R/W	R/W
Reset			0	0			0	0

Bits 29:28 – PS[1:0] Page Size

If page mode is enabled, this field indicates the size of the page in bytes.

Value	Name	Description
0	4 BYTE	4-byte page
1	8 BYTE	8-byte page
2	16 BYTE	16-byte page
3	32 BYTE	32-byte page

Bit 24 – PMEN Page Mode Enabled

Value	Description
0	Standard read is applied.
1	Asynchronous burst read in page mode is applied on the corresponding chip select.

Bit 20 – TDF MODE TDF, Data Float Time, Optimization

Value	Description
0	TDF optimization disabled-the number of TDF wait states is inserted before the next access begins.
1	TDF optimization enabled-the number of TDF wait states is optimized using the setup period of the next read/write access.

Bits 19:16 – TDF CYCLES[3:0] Data Float Time

This field gives the integer number of clock cycles required by the external device to release the data after the rising edge of the read controlling signal. The SMC always provide one full cycle of bus turnaround after the TDF CYCLES period. The external bus cannot be used by another chip select during TDF CYCLES + 1 cycles. From 0 up to 15 TDF CYCLES can be set.

Bit 12 – DBW Data Bus Width

Value	Name	Description
0	8 BIT	8-bit Data Bus
1	16 BIT	16-bit Data Bus
2	-	Reserved
3	-	Reserved

Bit 8 – BAT Byte Access Type

This field is used only if DBW defines a 16-bit data bus.

Value	Name	Description
0	BYTE SELECT	Byte select access type: • Write operation is controlled using NCS, NWE, NBS0, NBS1, NBS2 and NBS3 • Read operation is controlled using NCS, NRD, NBS0, NBS1, NBS2 and NBS3
1	BYTE WRITE	Byte write access type: • Write operation is controlled using NCS, NWR0, NWR1, NWR2,NWR3 • Read operation is controlled using NCS and NRD

Bits 5:4 – EXNW MODE[1:0] NWAIT Mode

The NWAIT signal is used to extend the current read or write signal. It is only taken into account during the pulse phase of the read and write controlling signal. When the use of NWAIT is enabled, at least one cycle hold duration must be programmed for the read and write controlling signal.

Value	Name	Description
0	DISABLED	Disabled-The NWAIT input signal is ignored on the corresponding chip select.
1	Reserved	
2	FROZEN	Frozen Mode-If asserted, the NWAIT signal freezes the current read or write cycle. After deassertion, the read/write cycle is resumed from the point where it was stopped.
3	READY	Ready Mode-The NWAIT signal indicates the availability of the external device at the end of the pulse of the controlling read or write signal, to complete the access. If high, the access normally completes. If low, the access is extended until NWAIT returns high.

Bit 1 – WRITE MODE Write Mode

Value	Name	Description
0	NCS_CTRL	Write operation controlled by NCS signal—If TDF optimization is enabled (TDF_MODE = 1), TDF wait states will be inserted after the setup of NCS.
1	NWE_CTRL	Write operation controlled by NWE signal—If TDF optimization is enabled (TDF_MODE = 1), TDF wait states will be inserted after the setup of NWE.

Bit 0 – READ MODE Read Mode

Value	Name	Description
0	NCS_CTRL	Read operation controlled by NCS signal • If TDF cycles are programmed, the external bus is marked busy after the rising edge of NCS. • If TDF optimization is enabled (TDF_MODE = 1), TDF wait states are inserted after the setup of NCS.
1	NRD_CTRL	Read operation controlled by NRD signal • If TDF cycles are programmed, the external bus is marked busy after the rising edge of NRD. • If TDF optimization is enabled (TDF_MODE = 1), TDF wait states are inserted after the setup of NRD.

40.8.5. SMC Write Protection Mode Register

Name: WPMR
Offset: 0xE4
Reset: 0x00000000
Property: Read/Write

Table 40-11. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	WPKEY [23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	WPKEY [15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WPKEY [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
								WPEN
Access								R/W
Reset								0

Bits 31:8 – WPKEY [23:0] Write Protection Key

Value	Name	Description
0x534D43	PASSWD	Writing any other value in this field aborts the write operation of the WPEN bit. Always reads as 0.

Bit 0 – WPEN Write Protect Enable

See Register Write Protection for the list of registers that can be write-protected.

Value	Description
0	Disables the write protection if WPKEY corresponds to 0x534D43 ("SMC" in ASCII).
1	Enables the write protection if WPKEY corresponds to 0x534D43 ("SMC" in ASCII).

40.8.6. SMC Write Protection Status Register

Name: WPSR
Offset: 0xE8
Reset: 0x00000000
Property: Read-Only

Table 40-12. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	WPVSRC [15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WPVSRC [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
								WPVS
Access								R/W
Reset								0

Bits 23:8 – WPVSRC [15:0] Write Protection Violation Source
When WPVS = 1, WPVSRC indicates the register address offset at which a write access has been attempted.

Bit 0 – WPVS Write Protect Violation Status

Value	Description
0	No write protection violation has occurred since the last read of the SMC WPSR register.
1	A write protection violation has occurred since the last read of the SMC WPSR register. If this violation is an unauthorized attempt to write a protected register, the associated violation is reported into field WPVSRC.

41. SD/MMC Host Controller (SDHC)

41.1. Overview

The SD/MMC Host Controller (SDHC) supports the embedded MultiMedia Card (e.MMC) Specification, the SD Memory Card Specification, and the SDIO Specification. It is compliant with the SD Host Controller Standard specifications.

The SDHC includes the register set defined in the "SD Host Controller Simplified Specification V3.00" and additional registers to manage e.MMC devices and enhanced features.

The SDHC is clocked by up to three clocks (bus clock, SDHC core clock, and a slow clock for certain functions). Both the MCLK and GCLK must be configured before the SDHC can be used.

41.1.1. Reference Documents

Name	Link
SD Host Controller Simplified Specification V3.00	https://www.sdcard.org
SDIO Simplified Specification V3.00	
Physical Layer Simplified Specification V3.01	
Embedded MultiMedia Card (e.MMC) Electrical Standard 4.51	http://www.jedec.org

41.2. Features

The following are key features of the SDHC module:

- Compatibility:
  - SD Host Controller Standard Specification
  - MultiMedia Card Specification
  - SD Memory Card Specification
  - SDIO Specification Version
- Support for 1-bit/ 4-bit SD/SDIO Devices
- Support for 1-bit/4-bit e.MMC Devices
- Support for SD/SDIO Default Speed (Maximum SDCLK Frequency = 25MHz)
- Support for SD/SDIO High Speed (Maximum SDCLK Frequency = 50MHz)
- Support for e.MMC Default Speed (Maximum SDCLK Frequency = 26MHz)
- e.MMC Boot Operation Mode Support
- Support for Block Size from 1 to 512bytes
- Support for Stream, Block and Multi-block Data Read and Write -Advanced DMA and SDMA Capability
- Internal 2 x 512, (1024) -byte Dual Port RAM
- Support for both synchronous and asynchronous abort
- Supports for SDIO Card Interrupt

41.3. Block Diagram

Figure 41-1. SDHC Block Diagram

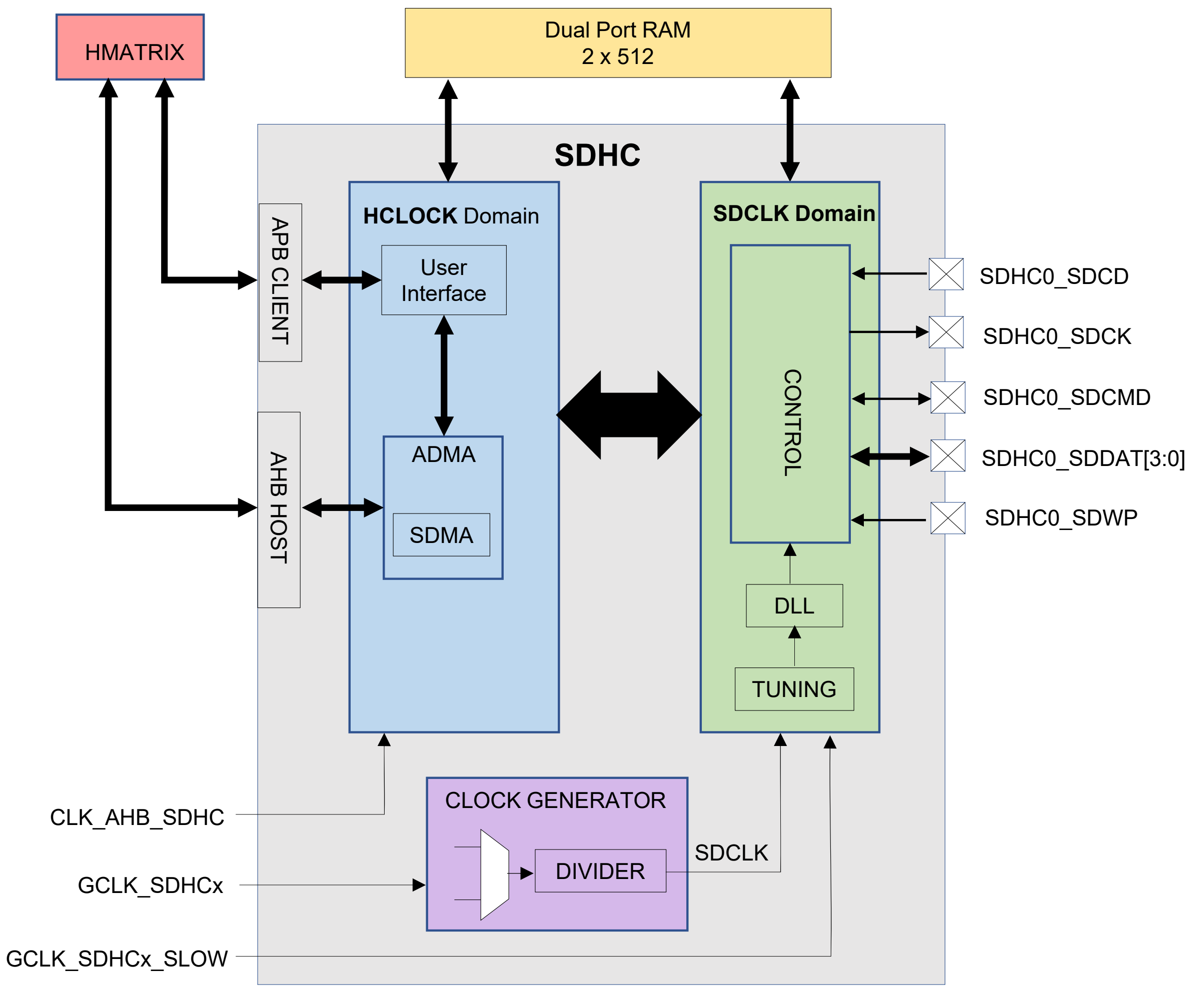
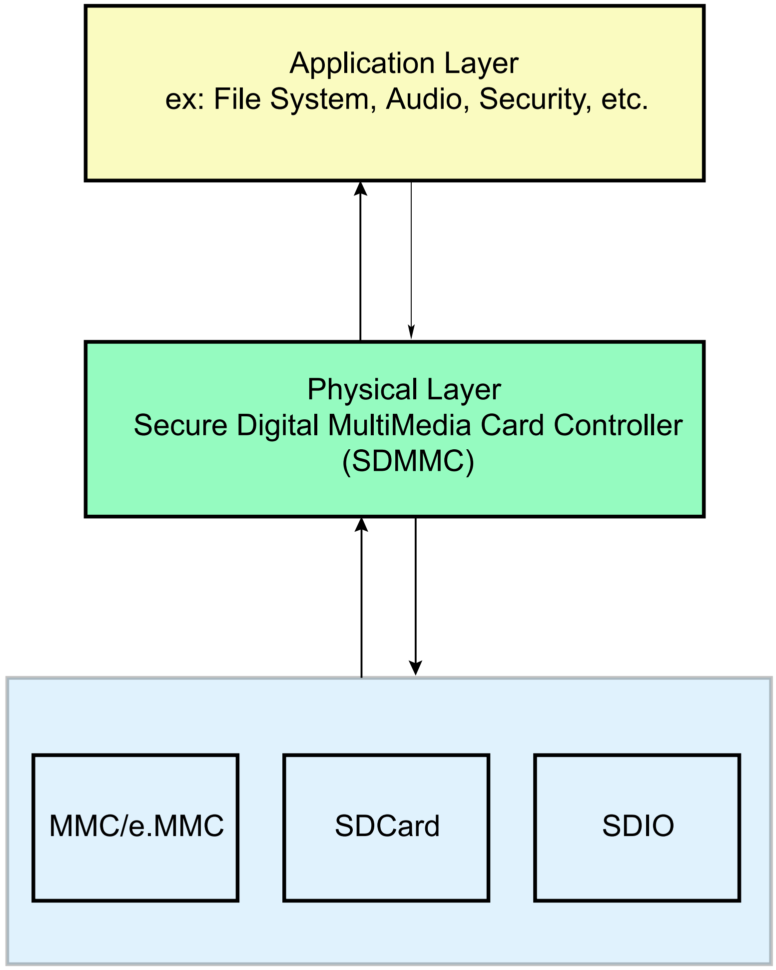
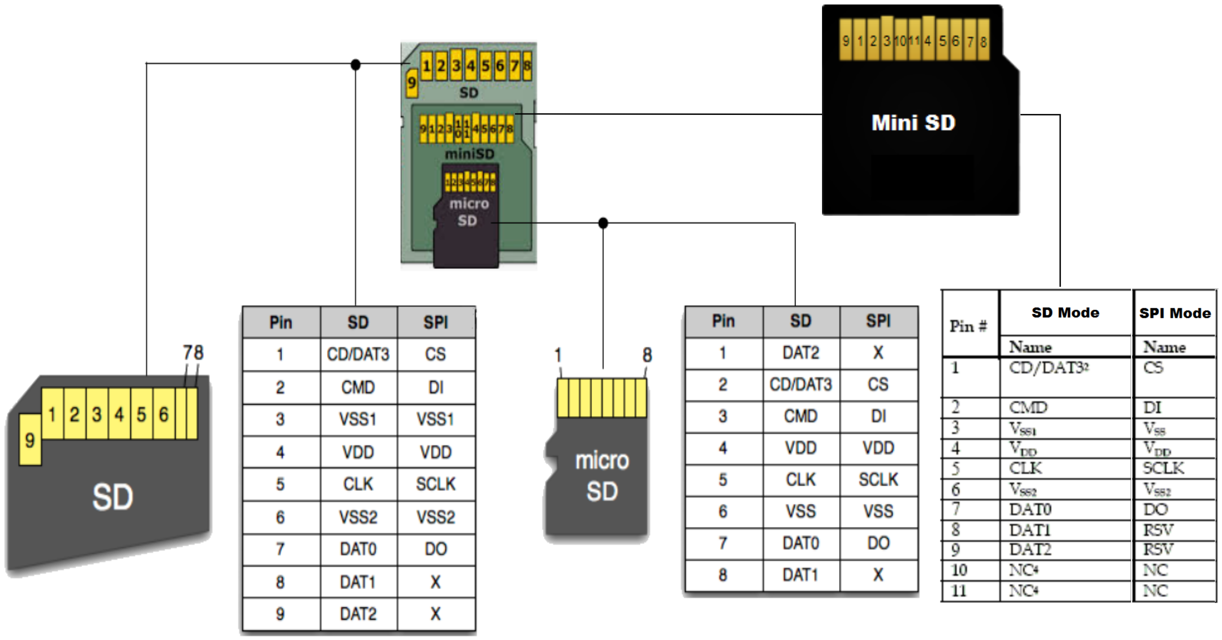


Figure 41-2. Application Block Diagram



41.4. Connection Diagram

Figure 41-3. RS SDHC Connection Diagram



41.5. Signal Description

SignalName	Type	Description
SDHCx_SDCD	Digital Input	SD Card / SDIO /e.MMC Card Detect (Requires 10k to 100k external pull-up)
SDHCx_SDCMD	Digital I/O	SD Card / SDIO /e.MMC Command/Response Line (Requires 10k to 100k external pull-up)
SDHCx_SDWP	Digital Input	SD Card Connector Write Protect Signal (Requires 10k to 100k external pull-up)
SDMMC CK	Digital Output	SD Card / SDIO /e.MMC Clock Signal
SDHCx_SDDAT[3:0]	Digital I/O	SD Card / SDIO /e.MMC data lines (Requires 10k to 100k external pull-up)

Note:
This product supports up to two, (i.e. x=0,1), SDHC controllers. Refer to the Pinout for additional information on the pin mapping for this peripheral.

41.6. Peripheral Dependencies

Peripheral Name	Base Address	NVIC IRQ Index:Source	MCLK AHBx/APBx Clock Enable Mask Bit	GCLK Peripheral Channel Clock Name : Register	PAC Peripheral ID (PAC.WRCTRL.PERIDx)	Power Domain
SDMMC0	0x4502_6000	148 : LINE, TIMER	MCLK.CLKMSK0[17]	GCLK_SDMMC0_SLOW : GCLK.PCHCTRL[18] GCLK_SDMMC0_CORE : GCLK.PCHCTRL[44]	55	VDDREG
SDMMC1	0x4502_8000	149 : LINE, TIMER	MCLK.CLKMSK0[18]	GCLK_SDMMC1_SLOW : GCLK.PCHCTRL[18] GCLK_SDMMC1_CORE : GCLK.PCHCTRL[45]	56	VDDREG

41.6.1. I/O Lines

In order to use the I/O lines, the I/O pins must be configured using the IO Pin Controller (PORT).

41.6.2. Clocks

The peripheral is using two generic clocks and one bus clock.

The clock for the SDHC bus interface (CLK AHB SDHC) is enabled and disabled by the Main Clock Controller. The default state of CLK AHB SDHC can be found in the Peripheral Clock Masking section.

The two generic clocks are:

- The core clock GCLK SDHCx is required to clock the SDHC core.
- The slow clock GCLK SDHCx SLOW is only required for certain functions. When this clock is required, GCLK SDHCx must be enabled.

These clocks must be configured and enabled in the Generic Clock Controller (GCLK) before using the SDHC. The generic clocks are asynchronous to the user interface clock (CLK SDHCx AHB). Due to this asynchronicity, writing to certain registers will require synchronization between the clock domains.

41.6.3. DMA

The SDHC has a built-in Direct Memory Access (DMA) and will read/write data to/from the system RAM when a SDHC transaction takes place. No CPU or DMA Controller (DMAC) resources are required.

41.6.4. Interrupts

The interrupt request line is connected to the Interrupt Controller. In order to use interrupt requests of this peripheral, the Interrupt Controller (NVIC) must be configured first.

41.7. Functional Description

41.7.1. SD/SDIO Operating Mode

This peripheral is fully compliant with the "SD Host Controller Simplified Specification V3.00" for SD/SDIO devices. Refer to this specification for configuration.

41.7.2. e.MMC Operating Mode

This peripheral supports e.MMC devices management. As the "SD Host Controller Simplified Specification V3.00" does not apply to e.MMC devices, some registers have been added to those described in this specification in order to manage e.MMC devices. Most of the registers described in the "SD Host Controller Simplified Specification V3.00" must be used for e.MMC management, but e.MMC- specific features are managed using SDHC_MC1R and SDHC_MC2R.

41.8. Register Summary

For descriptions and definitions of both Register and bitfield properties, refer to Register Properties.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	SSAR	31:24	ARG2[31:24]							
		23:16	ARG2[23:16]							
		15:8	ARG2[15:8]							
		7:0	ARG2[7:0]							
0x04	BSR	15:8		BOUNDARY[2:0]					BLKSIZE[9:8]	
		7:0	BLKSIZE[7:0]							
0x06	BCR	15:8	BLKCNT[15:8]							
		7:0	BLKCNT[7:0]							
0x08	ARG1R	31:24	ARG1[31:24]							
		23:16	ARG1[23:16]							
		15:8	ARG1[15:8]							
		7:0	ARG1[7:0]							
0x0C	TMR	15:8								
		7:0			MSBSEL	DTDSEL	ACMDEN[1:0]		BCEN	DMAEN
0x0E	CR	15:8			CMDIDX[5:0]					
		7:0	CMDTYP[1:0]		DPSEL	CMDICEN	CMDCCEN		RESPTYP[1:0]	
0x10	RR0	31:24	CMDRESP[31:24]							
		23:16	CMDRESP[23:16]							
		15:8	CMDRESP[15:8]							
		7:0	CMDRESP[7:0]							
0x14	RR1	31:24	CMDRESP[31:24]							
		23:16	CMDRESP[23:16]							
		15:8	CMDRESP[15:8]							
		7:0	CMDRESP[7:0]							
0x18	RR2	31:24	CMDRESP[31:24]							
		23:16	CMDRESP[23:16]							
		15:8	CMDRESP[15:8]							
		7:0	CMDRESP[7:0]							
0x1C	RR3	31:24	CMDRESP[31:24]							
		23:16	CMDRESP[23:16]							
		15:8	CMDRESP[15:8]							
		7:0	CMDRESP[7:0]							
0x20	BDPR	31:24	BUFDATA[31:24]							
		23:16	BUFDATA[23:16]							
		15:8	BUFDATA[15:8]							
		7:0	BUFDATA[7:0]							
0x24	PSR	31:24								CMDLL
		23:16	DATLL[3:0]				WRPPL	CARDDPL	CARDSS	CARDINS
		15:8					BUFRDEN	BUFWREN	RTACT	WTACT
		7:0						DLACT	CMDINHD	CMDINHC
0x28	HC1R	7:0	CARDDSEL	CARDDTL	EXTDW	DMASEL[1:0]		HSEN	DW	LEDCTRL
0x29	PCR	7:0								SDBPWR
0x2A	BGCR	7:0					INTBG	RWCTRL	CONTR	STPBGR
0x2B	WCR	7:0						WKENCREM	WKENCINS	WKENCINT
0x2C	CCR	15:8	SDCLKFSEL[7:0]							
		7:0	USDCLKFSEL[1:0]		CLKGSEL			SDCLKEN	INTCLKS	INTCLKEN
0x2E	TCR	7:0					DTCVAL[3:0]			
0x2F	SRR	7:0						SWRSTDAT	SWRSTCMD	SWRSTALL
0x30	NISTR	15:8	ERRINT	BOOTAR						CINT
		7:0	CREM	CINS	BRDRDY	BWRRDY	DMAINT	BLKGE	TRFC	CMDC
0x32	EISTR	15:8				BOOTAE		TUNING	ADMA	ACMD
		7:0	CURLIM	DATEND	DATCRC	DATTEO	CMDIDX	CMDEND	CMDCRC	CMDTEO
0x34	NISTER	15:8		BOOTAR						
		7:0	CREM	CINS	BRDRDY	BWRRDY	DMAINT	BLKGE	TRFC	CMDC
0x36	EISTER	15:8				BOOTAE		TUNING	ADMA	ACMD
		7:0	CURLIM	DATEND	DATCRC	DATTEO	CMDIDX	CMDEND	CMDCRC	CMDTEO

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x38	NISIER	15:8		BOOTAR						CINT
		7:0	CREM	CINS	BRDRDY	BWRRDY	DMAINT	BLKGE	TRFC	CMDC
0x3A	EISIER	15:8				BOOTAE		TUNING	ADMA	ACMD
		7:0	CURLIM	DATEND	DATCRC	DATTEO	CMDIDX	CMDEND	CMDCRC	CMDTEO
0x3C	ACESR	15:8								
		7:0	CMDNI			ACMDIDX	ACMDEND	ACMDCRC	ACMDTEO	ACMD12NE
0x3E	HC2R	15:8	PVALEN	ASINTEN						
		7:0								
0x40	CA0R	31:24	SLTYPE[1:0]		ASINTSUP	SB64SUP			V30VSUP	V33VSUP
		23:16	SRSUP	SDMASUP	HSSUP		ADMA2SUP	ED8SUP		
		15:8	BASECLKF[7:0]							
		7:0	TEOCLKU		TEOCLKF[5:0]					
0x44	CA1R	31:24								
		23:16	CLKMULT[7:0]							
		15:8								
		7:0		DRVDSUP	DRVCSUP	DRVASUP		DDR50SUP	SDR104SUP	SDR50SUP
0x48	MCCAR	31:24								
		23:16								
		15:8	MAXCUR30V[7:0]							
		7:0	MAXCUR33V[7:0]							
0x4C ... 0x4F	Reserved									
0x50	FERACES	15:8								
		7:0	CMDNI			ACMDIDX	ACMDEND	ACMDCRC	ACMDTEO	ACMD12NE
0x52	FEREIS	15:8				BOOTAE			ADMA	ACMD
		7:0	CURLIM	DATEND	DATCRC	DATTEO	CMDIDX	CMDEND	CMDCRC	CMDTEO
0x54	AESR	7:0						LMIS	ERRST[1:0]	
0x55 ... 0x57	Reserved									
0x58	ASARx	31:24	ADMASA[31:24]							
		23:16	ADMASA[23:16]							
		15:8	ADMASA[15:8]							
		7:0	ADMASA[7:0]							
0x5C ... 0x5F	Reserved									
0x60	PVR0	15:8						CLKGSEL	SDCLKFSEL[9:8]	
		7:0	SDCLKFSEL[7:0]							
0x62	PVR1	15:8						CLKGSEL	SDCLKFSEL[9:8]	
		7:0	SDCLKFSEL[7:0]							
0x64	PVR2	15:8						CLKGSEL	SDCLKFSEL[9:8]	
		7:0	SDCLKFSEL[7:0]							
0x66	PVR3	15:8						CLKGSEL	SDCLKFSEL[9:8]	
		7:0	SDCLKFSEL[7:0]							
0x68	PVR4	15:8						CLKGSEL	SDCLKFSEL[9:8]	
		7:0	SDCLKFSEL[7:0]							
0x6A	PVR5	15:8						CLKGSEL	SDCLKFSEL[9:8]	
		7:0	SDCLKFSEL[7:0]							
0x6C	PVR6	15:8						CLKGSEL	SDCLKFSEL[9:8]	
		7:0	SDCLKFSEL[7:0]							
0x6E	PVR7	15:8						CLKGSEL	SDCLKFSEL[9:8]	
		7:0	SDCLKFSEL[7:0]							
0x70 ... 0xFB	Reserved									
0xFC	SISR	15:8								
		7:0	INTSSL[7:0]							
0xFE	HCVR	15:8	VVER[7:0]							
		7:0	SVER[7:0]							

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0100 ... 0x01FF	Reserved									
0x0200	APSR	31:24								
		23:16								
		15:8								
		7:0					HDATLL[3:0]			
0x0204	MC1R	7:0	FCD		BOOTA				CMDTYP[1:0]	
0x0205	MC2R	7:0							ABOOT	SRESP
0x0206 ... 0x0207	Reserved									
0x0208	ACR	31:24								
		23:16								
		15:8								
		7:0			B1KBDIS	HNBRDIS			BMAX[1:0]	
0x020C	CC2R	31:24								
		23:16								
		15:8								
		7:0								FSDCLKD
0x0210 ... 0x022F	Reserved									
0x0230	CACR	31:24								
		23:16								
		15:8	KEY[7:0]							
		7:0								CAPWREN
0x0234	DBGR	15:8								
		7:0								NIDBG

41.8.1. SDHC SDMA System Address / Argument 2 Register

Name: SSAR
Offset: 0x00
Reset: 0x00000000
Property: -

This register contains the physical system memory address used for SDMA transfers or the second argument for Auto CMD23.

Table 41-1. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	ARG2[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
	ARG2[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
	ARG2[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
	ARG2[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 – ARG2[31:0] SDMA System Address/Argument 2
The function of this bit field depends on the operational mode:
For a SDMA transfer, this field is the system memory address. When the peripheral stops an SDMA transfer, this field points to the system address of the next contiguous data position. This field can be accessed only if no transaction is executing (i.e., after a transaction has stopped). Read operations during transfers may return an invalid value. An interrupt can be generated to instruct the software to update this field. Writing the next system address of the next data position restarts the SDMA transfer.
When executing Auto CMD23, this field is used with Auto CMD23 to set a 32-bit block count value to the CMD23 argument. If Auto CMD23 is used with ADMA, the full 32-bit block count value can be used. If Auto CMD23 is used without ADMA, the available block count value is limited by SDHC_BCR. In this case, 65535 blocks is the maximum value.

41.8.2. Block Size Register

Name: BSR
Offset: 0x04
Reset: 0x0000
Property: -

Table 41-2. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	15	14	13	12	11	10	9	8
		BOUNDARY[2:0]					BLKSIZE[9:8]	
Access		-	-	-			R/W	R/W
Reset		0	0	0			0	0

Bit	7	6	5	4	3	2	1	0
	BLKSIZE[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 14:12 – BOUNDARY[2:0] SDMA Buffer Boundary

This field specifies the size of the contiguous buffer in the system memory. The SDMA transfer waits at every boundary specified by this field and the peripheral generates the DMA Interrupt to instruct the software to update SDHC_SSAR. If this field is set to 0 (buffer size = 4 Kbytes), the lowest 12 bits of SDHC_SSAR.ADDRESS point to data in the contiguous buffer, and the upper 20 bits point to the location of the buffer in the system memory. This function is active when the DMA Enable bit in the Transfer Mode Register (SDHC_TMR.DMAEN) is '1'.

Value	Name	Description
0	4K	4-Kbyte boundary
1	8K	8-Kbyte boundary
2	16K	16-Kbyte boundary
3	32K	32-Kbyte boundary
4	64K	64-Kbyte boundary
5	128K	128-Kbyte boundary
6	256k	256-Kbyte boundary
7	512K	512-Kbyte boundary

Bits 9:0 – BLKSIZE[9:0] Transfer Block Size

This field specifies the block size of data transfers for CMD17, CMD18, CMD24, CMD25 and CMD53. Values ranging from 1 to SDMMC_MAX_BLOCK_SIZE can be set. It can be accessed only if no transaction is executing (i.e., after a transaction has stopped). Read operations during transfers may return an invalid value, and write operations are ignored.

41.8.3. SDHC Block Count Register

Name: BCR
Offset: 0x06
Reset: 0x0000
Property: -

Table 41-3. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	15	14	13	12	11	10	9	8
	BLKCNT[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
	BLKCNT[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 – BLKCNT[15:0] Block Count for Current Transfer

This field is used only if SDHC_TMR.BCEN (Block Count Enable) is set to 1 and is valid only for multiple block transfers. BLKCNT is the number of blocks to be transferred and it must be set to a value between 1 and the maximum block count. The peripheral decrements the block count after each block transfer and stops when the count reaches 0. When this field is set to 0, no data block is transferred.

This register should be accessed only when no transaction is executing (i.e., after transactions are stopped). During data transfer, read operations on this register may return an invalid value and write operations are ignored.

When a suspend command is completed, the number of blocks yet to be transferred can be determined by reading this register. Before issuing a resume command, the previously saved block count is restored.

41.8.4. SDHC Argument 1 Register

Name: ARG1R
Offset: 0x08
Reset: 0x00000000
Property: Read/Write

Table 41-4. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	ARG1[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
	ARG1[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
	ARG1[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
	ARG1[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 – ARG1[31:0] Argument 1

This register contains the SD command argument which is specified as the bit 39-8 of Command-Format in the “Physical Layer Simplified Specification V3.01” or “Embedded MultiMedia Card (e.MMC) Electrical Standard 4.51”.

41.8.5. SDHC_Transfer Mode Register

Name: TMR
Offset: 0x0C
Reset: 0x0000
Property: -

This register is used to control data transfers. The user shall set this register before issuing a command which transfers data (refer to bit DPSEL in SDHC_CR), or before issuing a Resume command. The user must save the value of this register when the data transfer is suspended (as a result of a Suspend command) and restore it before issuing a Resume command. To prevent data loss, this register cannot be written while data transactions are in progress. Writes to this register are ignored when bit SDHC_PSR.CMDINHD is '1'.

Table 41-5. Determining the Transfer Type

MSBSEL	BCEN	SDHC_BCR.BLKCNT	Function
0	Don't care	Don't care	Single Transfer
1	0	Don't care	Infinite Transfer
1	1	Not Zero	Multiple Transfer
1	1	Zero	Stop Multiple Transfer

Table 41-6. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			MSBSEL	DTDSEL	ACMDEN[1:0]		BCEN	DMAEN
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit 5 – MSBSEL Multi/Single Block Selection
Write this bit to '1' when issuing multiple-block transfer commands using DAT line(s). For any other commands, write this bit to 0. If this bit is 0, it is not necessary to write SDHC_BCR to '1'.

Bit 4 – DTDSEL Data Transfer Direction Selection
This bit defines the direction of the DAT lines data transfers. Write this bit to '1' to transfer data from the device (SD Card/SDIO/e.MMC) to the peripheral. Write this bit to '0' for all other commands.

Value	Name	Description
0	WRITE	Writes data from the peripheral to the device.
1	READ	Reads data from the device to the peripheral.

Bits 3:2 – ACMDEN[1:0] Auto Command Enable
Two methods can be used to stop Multiple-block read and write operation:

1. Auto CMD12: when the ACMDEN field is set to 1, the peripheral issues CMD12 automatically when the last block transfer is completed. An Auto CMD12 error is indicated to SDHC_ACESR. Auto CMD12 is not enabled if the command does not require CMD12.
2. Auto CMD23: when the ACMDEN field is set to 2, the peripheral issues a CMD23 automatically before issuing a command specified in SDHC_CR.

The following conditions are required to use Auto CMD23:

- A memory card that supports CMD23 (SCR[33] = 1)
- If DMA is used, it must be ADMA (SDMA not supported).
- Only CMD18 or CMD25 is issued.

Note: The peripheral does not check the command index.

Auto CMD23 can be used with or without ADMA. By writing SDHC_CR, the peripheral issues a CMD23 first and then issues a command specified by the SDHC_CR.CMDIDX field. If CMD23 response errors are detected, the second command is not issued. A CMD23 error is indicated in SDHC_ACESR. The CMD23 argument (32-bit block count value) is defined in SDHC_SSAR.
This field determines the use of auto command functions.

Value	Name	Description
0	DISABLED	Auto Command Disabled
1	CMD12	Auto CMD12 Enabled
2	CMD23	Auto CMD23 Enabled
3	Reserved	Reserved

Bit 1 – BCEN Block Count Enable

This bit is used to enable SDHC_BCR, which is only relevant for multiple block transfers. When this bit is 0, SDHC_BCR is disabled, which is useful when executing an infinite transfer . If an ADMA2 transfer is more than 65535 blocks, this bit is set to 0 and the data transfer length is designated by the Descriptor Table.

Value	Name	Description
0	DISABLED	Block count is disabled
1	ENABLED	Block count is enabled

Bit 0 – DMAEN DMA Enable

This bit enables the DMA functionality described in section “Supporting DMA” in “SD Host Controller Simplified Specification V3.00” . DMA can be enabled only if it is supported as indicated by the bit SDHC_CA0R.ADMA2SUP. One of the DMA modes can be selected using the field SDHC_HC1R.DMASEL. If DMA is not supported, this bit is meaningless and then always reads 0. When this bit is set to 1, a DMA operation begins when the user writes to the upper byte of SDHC_CR.

Value	Name	Description
0	DISABLED	DMA functionality is disabled
1	ENABLED	DMA functionality is enabled

41.8.6. SDHC_Command Register

Name: CR
Offset: 0x0E
Reset: 0x0000
Property: -

Table 41-7. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	15	14	13	12	11	10	9	8
			CMDIDX[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
	CMDTYP[1:0]		DPSEL	CMDICEN	CMDCCEN		RESPTYP[1:0]	
Access	R/W	R/W	R/W	-	R/W		R/W	R/W
Reset	0	0	0	0	0		0	0

Bits 13:8 – CMDIDX[5:0] Command Index

This bit shall be set to the command number (CMD0-63, ACMD0-63) that is specified in bits 45-40 of the Command-Format in the “Physical Layer Simplified Specification V3.01”, “SDIO Simplified Specification V3.00”, and “Embedded MultiMedia Card (e.MMC) Electrical Standard 4.51”.

Bits 7:6 – CMDTYP[1:0] Command Type

Value	Name	Description
0	NORMAL	Other commands
1	SUSPEND	CMD52 to write “Bus Suspend” in the Card Common Control Registers (CCCR) (for SDIO only)
2	RESUME	CMD52 to write “Function Select” in the Card Common Control Registers (CCCR) (for SDIO only)
3	ABORT	CMD12, CMD52 to write “I/O Abort” in the Card Common Control Registers (CCCR) (for SDIO only)

Bit 5 – DPSEL Data Present Select

This bit is set to 1 to indicate that data is present and shall be transferred using the DAT lines. It is set to 0 for the following:

1. Commands using only CMD line (Ex. CMD52)
2. Commands with no data transfer but using Busy signal on DAT[0] line (Ex. CMD38)
3. Resume command

Value	Description
0	No data present
1	Data present

Bit 4 – CMDICEN Command Index Check Enable

If this bit is set to 1, the peripheral checks the Index field in the response to see if it has the same value as the command index. If it has not, it is reported as a Command Index Error (CMDIDX) in SDHC_EISTR. If this bit is set to 0, the Index field of the response is not checked.

Value	Name	Description
0	DISABLED	The Command Index Check is disabled.

Value	Name	Description
1	ENABLED	The Command Index Check is enabled.

Bit 3 – CMDCCEN Command CRC Check Enable

If this bit is set to 1, the peripheral checks the CRC field in the response. If an error is detected, it is reported as a Command CRC Error (CMDCRC) in SDHC_EISTR. If this bit is set to 0, the CRC field is not checked. The position of the CRC field is determined according to the length of the response.

Value	Name	Description
0	DISABLED	The Command CRC Check is disabled.
1	ENABLED	The Command CRC Check is enabled.

Bits 1:0 – RESPTYP[1:0] Response Type

This field is set according to the response type expected for the command index (CMDIDX).

Value	Name	Description
0	NORESP	No Response
1	RL136	Response Length 136
2	RL48	Response Length 48
3	RL48BUSY	Response Length 48 with Busy

41.8.7. SDHC_Response Register x

Name: RRx
Offset: 0x10 + x*0x04 [x=0..3]
Reset: 0x000000000
Property: -

Table 41-8. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	CMDRESP[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
	CMDRESP[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
	CMDRESP[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
	CMDRESP[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – CMDRESP[31:0] Command Response
The table below describes the mapping of command responses from the SD/SDIO/e.MMC bus to these registers for each responses type. In this table, R[] refers to a bit range of the response data as transmitted on the SD/SDIO/e.MMC bus.

Type of response	Meaning of response	Response field	Response register
R1, R1b (normal response)	Card Status	R[39:8]	SDHC_RR0[31:0]
R1b (Auto CMD12 response)	Card Status for Auto CMD12	R[39:8]	SDHC_RR3[31:0]
R1 (Auto CMD23 response)	Card Status for Auto CMD23	R[39:8]	SDHC_RR3[31:0]
R2 (CID, CSD register)	CID or CSD register	R[127:8]	SDHC_RR0[31:0] SDHC_RR1[31:0] SDHC_RR2[31:0] SDHC_RR3[23:0]
R3 (OCR register)	OCR register for memory	R[39:8]	SDHC_RR0[31:0]
R4 (OCR register)	OCR register for I/O	R[39:8]	SDHC_RR0[31:0]
R5, R5b	SDIO response	R[39:8]	SDHC_RR0[31:0]
R6 (Published RCA response)	New published RCA[31:16] and Card status bits	R[39:8]	SDHC_RR0[31:0]

41.8.8. SDHC Buffer Data Port Register

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

Table 41-9. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	BUFDATA[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
	BUFDATA[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
	BUFDATA[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
	BUFDATA[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 – BUFDATA[31:0] Buffer Data
The peripheral's data buffer can be accessed through this 32-bit Data Port register.

41.8.9. SDHC Present State Register

Name: PSR
Offset: 0x24
Reset: 0x00F80000
Property: -

Table 41-10. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
								CMDLL
Access								R
Reset								0

Bit	23	22	21	20	19	18	17	16
	DATLL[3:0]				WRPPL	CARDDPL	CARDSS	CARDINS
Access	R	R	R	R	R	R	R	R
Reset	1	1	1	1	1	0	0	0

Bit	15	14	13	12	11	10	9	8
					BUFRDEN	BUFWREN	RTACT	WTACT
Access					R	R	R	R
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
						DLACT	CMDINHD	CMDINHC
Access						R	R	R
Reset						0	0	0

Bit 24 – CMDLL CMD Line Level
This status is used to check the CMD line level to recover from errors, and for debugging.

Bits 23:20 – DATLL[3:0] DAT[3:0] Line Level
This status is used to check the DAT line level to recover from errors, and for debugging. This is especially useful in detecting the Busy signal level from DAT[0].

Bit 19 – WRPPL Write Protect Pin Level
The Write Protect Switch is supported for memory and combo cards. This bit reflects the SDHC_WP pin.

Value	Description
0	Write protected (SDHC_WP = 0)
1	Write enabled (SDHC_WP = 1)

Bit 18 – CARDDPL Card Detect Pin Level
This bit reflects the inverse value of the SDHC_CD pin. Debouncing is not performed on this bit. This bit may be valid when CARDSS is set to 1, but it is not guaranteed because of the propagation delay. Use of this bit is limited to testing since it must be debounced by software.

Value	Description
0	No card present (SDHC_CD = 1)
1	Card present (SDHC_CD = 0)

Bit 17 – CARDSS Card State Stable

This bit is used for testing. If it is 0, the CARDDPL is not stable. If this bit is set to 1, it means that the CARDDPL is stable. No Card state can be detected if this bit is set to 1 and CARDINS is set to 0.
The Software Reset For All (SWRSTALL) in SDHC_SRR does not affect this bit.

Value	Description
0	Reset or debouncing
1	No card or card inserted

Bit 16 – CARDINS Card Inserted

This bit indicates whether a card has been inserted. The peripheral debounces this signal so that the user does not need to wait for it to stabilize.
A change from 0 to 1 rises the Card Insertion (CINS) status flag in SDHC_NISTR if SDHC_NISTER.CINS is set to 1. An interrupt is generated if SDHC_NISIER.CINS is set to 1.
A change from 1 to 0 rises the Card Removal (CREM) status flag in SDHC_NISTR if SDHC_NISTER.CREM is set to 1. An interrupt is generated if SDHC_NISIER.CREM is set to 1.
The Software Reset For All (SWRSTALL) in SDHC_SRR does not affect this bit.

Bit 11 – BUFRDEN Buffer Read Enable

This bit is used for non-DMA read transfers. This flag indicates that valid data exists in the peripheral data buffer. If this bit is 1, readable data exists in the buffer.
A change from 1 to 0 occurs when all the block data is read from the buffer.
A change from 0 to 1 occurs when block data is ready in the buffer. This rises the Buffer Read Ready (BRDRDY) status flag in SDHC_NISTR if SDHC_NISTER.BRDRDY is set to 1. An interrupt is generated if SDHC_NISIER.BRDRDY is set to 1.

Bit 10 – BUFWREN Buffer Write Enable

This bit is used for non-DMA write transfers. This flag indicates if space is available for write data. If this bit is 1, data can be written to the buffer.
A change from 1 to 0 occurs when all the block data are written to the buffer.
A change from 0 to 1 occurs when top of block data can be written to the buffer. This rises the Buffer Write Ready (BRWRDY) status flag in SDHC_NISTR if SDHC_NISTER.BRWRDY is set to 1. An interrupt is generated if SDHC_NISIER.BRWRDY is set to 1.

Bit 9 – RTACT Read Transfer Active

This bit is used to detect completion of a read transfer. Refer to section “Read Transaction Wait / Continue Timing” in the “SD Host Controller Simplified Specification V3.00” for more details on the sequence of events.
This bit is set to 1 in either of the following conditions:

- After the end bit of the read command.
- When a read operation is restarted by writing a 1 to SDHC_BGCR.CONTR (Continue Request).

This bit is cleared to 0 in either of the following conditions:

- When the last data block as specified by Transfer Block Size (BLKSIZE) is transferred to the system.
- In case of ADMA2, end of read is designated by the descriptor table.
- When all valid data blocks in the peripheral have been transferred to the system and no current block transfers are being sent as a result of the Stop At Block Gap Request (STPBGR) of SDHC_BGCR being set to 1.

A change from 1 to 0 rises the Transfer Complete (TRFC) status flag in SDHC_NISTR if SDHC_NISTER.TRFC is set to 1. An interrupt is generated if SDHC_NISIER.TRFC is set to 1.

Bit 8 – WTACT Write Transfer Active

This bit indicates a write transfer is active. If this bit is 0, it means no valid write data exists in the peripheral. Refer to section “Write Transaction Wait / Continue Timing” in the “SD Host Controller Simplified Specification V3.00” for more details on the sequence of events.
This bit is set to 1 in either of the following conditions:

- After the end bit of the write command.
- When a write operation is restarted by writing a 1 to SDHC_BGCR.CONTR (Continue Request).

This bit is cleared to 0 in either of the following conditions:

- After getting the CRC status of the last data block as specified by the transfer count (single and multiple). In case of ADMA2, transfer count is designated by the descriptor table.
- After getting the CRC status of any block where a data transmission is about to be stopped by a Stop At Block Gap Request (STPBGR) of SDHC_BGCR.

During a write transaction and as the result of the Stop At Block Gap Request (STPBGR) being set, a change from 1 to 0 rises the Block Gap Event (BLKGE) status flag in SDHC_NISTR if SDHC_NISTER.BLKGE is set to 1. An interrupt is generated if BLKGE is set to 1 in SDHC_NISIER.
This status is useful to determine whether non-DAT line commands can be issued during Write Busy.

Bit 2 – DLACT DAT Line Active

This bit indicates whether one of the DAT lines on the bus is in use.
In the case of read transactions:
This status indicates whether a read transfer is executing on the bus. A change from 1 to 0 resulting from setting the Stop At Block Gap Request (STPBGR) rises the Block Gap Event (BLKGE) status flag in SDHC_NISTR if SDHC_NISTER.BLKGE is set to 1. An interrupt is generated if SDHC_NISIER.BLKGE is set to 1. Refer to section “Read Transaction Wait / Continue Timing” in the “SD Host Controller Simplified Specification V3.00” for details on timing.
This bit is set in either of the following cases:

- After the end bit of the read command.
- When writing 1 to SDHC_BGCR.CONTR (Continue Request) to restart a read transfer.

This bit is peripheral cleared in either of the following cases:

- When the end bit of the last data block is sent from the bus to the peripheral. In case of ADMA2, the last block is designated by the last transfer of the Descriptor Table.
- When a read transfer is stopped at the block gap initiated by a Stop At Block Gap Request (STPBGR).

The peripheral stops a read operation at the start of the interrupt cycle by driving the Read Wait (DAT[2] line) or by stopping the SD Clock. If the Read Wait signal is already driven (due to the fact that the data buffer cannot receive data), the peripheral can continue to stop the read operation by driving the Read Wait signal. It is necessary to support the Read Wait in order to use the Suspend/Resume operation.
In the case of write transactions:
This status indicates that a write transfer is executing on the bus. A change from 1 to 0 rises the Transfer Complete (TRFC) status flag in SDHC_NISTR if SDHC_NISTER.TRFC is set to 1. An interrupt is generated if SDHC_NISIER.TRFC is set to 1. Refer to section “Write Transaction Wait / Continue Timing” in the “SD Host Controller Simplified Specification V3.00” for details on timing.
This bit is set in either of the following cases:

- After the end bit of the write command.
- When writing 1 to SDHC_BGCR.CONTR (Continue Request) to continue a write transfer.

This bit is cleared in either of the following cases:

- When the card releases Write Busy of the last data block. If the card does not drive a Busy signal for 8 SDCLK, the peripheral considers the card drive “Not Busy”. In the case of ADMA2, the last block is designated by the last transfer of the Descriptor Table.
- When the card releases Write Busy prior to wait for write transfer as a result of a Stop At Block Gap Request (STPBGR).

Command with Busy:
This status indicates whether a command that indicates Busy (ex. erase command for memory) is executing on the bus. This bit is set to 1 after the end bit of the command with Busy and cleared when Busy is de-asserted. A change from 1 to 0 rises the Transfer Complete (TRFC) status flag in SDHC_NISTR if SDHC_NISTER.TRFC is set to 1. An interrupt is generated if SDHC_NISIER.TRFC is set to 1. Refer to Figures 2.11 to 2.13 in the “SD Host Controller Simplified Specification V3.00”.

Value	Description
0	DAT Line Inactive
1	DAT Line Active

Bit 1 – CMDINHD Command Inhibit (DAT)

This status bit is 1 if either the DAT Line Active (DLACT) or the Read Transfer Active (RTACT) is set to 1. If this bit is 0, it indicates that the peripheral can issue the next command. Commands with a Busy signal belong to Command Inhibit (DAT) (ex. R1b, R5b type). A change from 1 to 0 rises the Transfer Complete (TRFC) status flag in SDHC_NISTR if SDHC_NISTER.TRFC is set to 1. An interrupt is generated if SDHC_NISIER.TRFC is set to 1.
Note: The software can save registers in the 000-00Dh range for a suspend transaction after this bit has changed from 1 to 0.

Value	Description
0	Can issue a command which uses the DAT line(s).
1	Cannot issue a command which uses the DAT line(s).

Bit 0 – CMDINHC Command Inhibit (CMD)

If this bit is 0, it indicates the CMD line is not in use and the peripheral can issue a command using the CMD line. This bit is set to 1 immediately after SDHC_CR is written. This bit is cleared when the command response is received. Auto CMD12 and Auto CMD23 consist of two responses. In this case, this bit is not cleared by the CMD12 or CMD23 response, but by the Read/Write command response.
Status issuing Auto CMD12 is not read from this bit. So, if a command is issued during Auto CMD12 operation, the peripheral manages to issue both commands: CMD12 and a command set by SDHC_CR.
Even if the Command Inhibit (DAT) is set to 1, commands using only the CMD line can be issued if this bit is 0.
A change from 1 to 0 rises the Command Complete (CMDC) status flag in SDHC_NISTR if SDHC_NISTER.CMDC is set to 1. An interrupt is generated if SDHC_NISIER.CMDC is set to 1.
If the peripheral cannot issue the command because of a command conflict error (refer to CMDCRC in SDHC_EISTR) or because of a ‘Command Not Issued By Auto CMD12’ error, this bit remains 1 and Command Complete is not set.

Value	Description
0	Can issue a command using only CMD line.
1	Cannot issue a command.

41.8.10. Host Control 1 Register

Name: HC1R
Offset: 0x28
Reset: 0x00
Property: -

Table 41-11. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
	CARDDSEL	CARDDTL	EXTDW	DMASEL[1:0]		HSEN	DW	LEDCTRL
Access	R/W	R/W	R/W	-	-	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 7 – CARDDSEL Card Detect Signal Selection

Note: This register entry is specific to the SD/SDIO operation mode.

This bit selects the source for the card detection.

Value	Description
0	The SDHC_CD pin is selected.
1	The Card Detect Test Level (CARDDTL) is selected (for test purpose).

Bit 6 – CARDDTL Card Detect Test Level

Note: This register entry is specific to the SD/SDIO operation mode.

This bit is enabled while the Card Detect Signal Selection (CARDDSEL) is set to 1 and it indicates whether the card is inserted or not.

Value	Description
0	No card.
1	Card inserted.

Bit 5 – EXTDW Extended Data Width

Note: This register entry is specific to the e.MMC operation mode.

This bit controls the 8-bit Bus Width mode for embedded devices. Support of this function is indicated in 8-bit Support for Embedded Device in SDHC_CA0R. If a device supports the 8-bit mode, this may be set to 1. If this bit is 0, the bus width is controlled by Data Width (DW).

Bits 4:3 – DMASEL[1:0] DMA Select

One of the supported DAM modes can be selected. The user must check support of DMA modes by referring the SDHC_CA0R. Use of selected DMA is determined by DMA Enable (DMAEN) in SDHC_TMR.

Value	Name	Description
0	SDMA	SDMA is selected
1	Reserved	Reserved
2	ADMA32	32-bit Address ADMA2 is selected
3	Reserved	Reserved

Bit 2 – HSEN High Speed Enable

Before setting this bit, the user must check the High Speed Support (HSSUP) in SDHC_CA0R.
If this bit is set to 0 (default), the peripheral outputs CMD line and DAT lines at the falling edge of the SD clock (up to 25 MHz). If this bit is set to 1, the SDMMC outputs the CMD line and the DAT lines at the rising edge of the SD clock (up to 50 MHz).
If Preset Value Enable (PVALEN) in SDHC_HC2R is set to 1, the user needs to reset SD Clock Enable (SDCLKEN) before changing this bit to avoid generating clock glitches. After setting this bit to 1, the user sets SDCLEN to 1 again.

Value	Description
0	Normal Speed mode.
1	High Speed mode. Note: 1. This bit is effective only if SDHC_MC1R.DDR is set to 0. 2. The clock divider (DIV) in SDHC_CCR must be set to a value different from 0 when HSEN is 1.

Bit 1 – DW Data Width

This bit selects the data width of the peripheral. It must be set to match the data width of the card.
Note: If the Extended Data Transfer Width is 1, this bit has no effect and the data width is 8-bit mode.

Value	Name	Description
0	1_BIT	1-bit mode
1	4_BIT	4-bit mode

Bit 0 – LEDCTRL LED Control

Note: This register entry is specific to the SD/SDIO operation mode.

This bit is used to caution the user not to remove the card while it is being accessed. If the software is going to issue multiple commands, this bit is set to 1 during all transactions.

Value	Name	Description
0	OFF	LED off
1	ON	LED on

41.8.11. SDHC Power Control Register

Name: PCR
Offset: 0x29
Reset: 0x0E
Property: -

Table 41-12. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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

Bit 0 – SDBPWR SD Bus Power
This bit is automatically cleared by the peripheral if the card is removed. If this bit is cleared, the peripheral stops driving SDHC_CMD and SDHC_DAT[7:0] (tri-state) and drives SDHC_CK to low level.

41.8.12. SDHC Block Gap Control Register

Name: BGCR
Offset: 0x2A
Reset: 0x00
Property: -

Table 41-13. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
					INTBG	RWCTRL	CONTR	STPBGR
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit 3 – INTBG Interrupt at Block Gap

Note: This register entry is specific to the SD/SDIO operation mode.

This bit is valid only in 4-bit mode of the SDIO card and selects a sample point in the interrupt cycle. Setting to 1 enables interrupt detection at the block gap for a multiple block transfer. If the SDIO card cannot signal an interrupt during a multiple block transfer, this bit should be set to 0. When the software detects an SDIO card insertion, it sets this bit according to the CCCR of the SDIO card.

Value	Name	Description
0	DISABLED	Interrupt detection disabled
1	ENABLED	Interrupt detection enabled

Bit 2 – RWCTRL Read Wait Control

Note: This register entry is specific to the SD/SDIO operation mode.

The Read Wait control is optional for SDIO cards. If the card supports Read Wait, set this bit to enable use of the Read Wait protocol to stop read data using the SDHC_DAT[2] line. Otherwise, the peripheral stops the SDCLK to hold read data, which restricts command generation. When the software detects an SD card insertion, this bit must be set according to the CCCR of the SDIO card. If the card does not support Read Wait, this bit shall never be set to 1, otherwise an SDHC_DAT line conflict may occur. If this bit is set to 0, Suspend/Resume cannot be supported.

Value	Description
0	Disables Read Wait control.
1	Enables Read Wait control.

Bit 1 – CONTR Continue Request

This bit is used to restart a transaction which was stopped using a Stop At Block Gap Request (STPBGR). To cancel stop at the block gap, set STPBGR to 0 and set this bit to 1 to restart the transfer.
The peripheral automatically clears this bit in either of the following cases:

- In the case of a read transaction, the DAT Line Active (DLACT) changes from 0 to 1 as a read transaction restarts.
- In the case of a write transaction, the Write Transfer Active (WTACT) changes from 0 to 1 as the write transaction restarts.

Therefore, it is not necessary to set this bit to 0. If STPBGR is set to 1, any write to this bit is ignored.

Refer to the “Abort Transaction” and “Suspend/Resume” sections in the “SD Host Controller Simplified Specification V3.00” for more details.

Value	Description
0	No affect
1	Restart

Bit 0 – STPBGR Stop At Block Gap Request

This bit is used to stop executing read and write transactions at the next block gap for non-DMA, SDMA, and ADMA transfers. The user must leave this bit set to 1 until Transfer Complete (TRFC) in SDHC_NISTR. Clearing both Stop At Block Gap Request and Continue Request does not cause the transaction to restart. This bit can be set whether the card supports the Read Wait signal or not.
During read transfers, the peripheral stops the transaction by using the Read Wait signal (SDHC_DAT[2]) if supported, or by stopping the SD clock otherwise.
In case of write transfers in which the user writes data to SDHC_BDPR, this bit must be set to 1 after all the block of data is written. If this bit is set to 1, the user does not write data to SDHC_BDPR.
This bit affects Read Transfer Active (RTACT), Write Transfer Active (WTACT), DAT Line Active (DLACT) and Command Inhibit (DAT) (CMDINHD) in SDHC_PSR.
Refer to the “Abort Transaction” and “Suspend/Resume” sections in the “SD Host Controller Simplified Specification V3.00” for more details.

Value	Description
0	Transfer
1	Stop

41.8.13. SDHC Wakeup Control Register: SD/SDIO

Name: WCR
Offset: 0x2B
Reset: 0x00
Property: -

Table 41-14. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
						WKENCREM	WKENCINS	WKENCINT
Access						R/W	R/W	R/W
Reset						0	0	0

Bit 2 – WKENCREM Wake-up Event Enable on Card Removal
This bit enables a wake-up event through Card Removal (CREM) in SDHC_NISTR. FN_WUS (Wake-Up Support) in the CIS (Card Information Structure) does not affect this bit.

Value	Name	Description
0	DISABLED	Wake-Up Event disabled
1	ENABLED	Wake-Up Event enabled

Bit 1 – WKENCINS Wake-Up Event Enable on Card Insertion
This bit enables a wake-up event through Card Insertion (CINS) in SDHC_NISTR. FN_WUS (Wake-Up Support) in the CIS (Card Information Structure) does not affect this bit.

Value	Name	Description
0	DISABLED	Wake-Up Event disabled
1	ENABLED	Wake-Up Event enabled

Bit 0 – WKENCINT Wake-Up Event Enable on Card Interrupt
This bit enables a wake-up event through Card Interrupt (CINT) in SDHC_NISTR. This bit can be set to 1 if FN_WUS (Wake-Up Support) in the CIS (Card Information Structure) is set to 1 in the SDIO card.

Value	Name	Description
0	DISABLED	Wake-Up Event disabled
1	ENABLED	Wake-Up Event enabled

41.8.14. SDHC Clock Control Register

Name: CCR
Offset: 0x2C
Reset: 0x0000
Property: -

Table 41-15. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	15	14	13	12	11	10	9	8
	SDCLKFSEL[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
	USDCLKFSEL[1:0]		CLKGSEL			SDCLKEN	INTCLKS	INTCLKEN
Access	R/W	R/W	R/W			R/W	R/W	R/W
Reset	0	0	0			0	0	0

Bits 15:8 – SDCLKFSEL[7:0] SDCLK Frequency Select

This register is used to select the frequency of the SDCLK pin. There are two SDCLK Frequency modes according to Clock Generator Select (CLKGSEL).
The length of the clock divider (DIV) is extended to 10 bits (DIV[9:8] = USDCLKFSEL, DIV[7:0] = SDCLKFSEL)
– 10-bit Divided Clock Mode (CLKGSEL = 0):

$$F_{\mathrm{SDCLK}} = F_{\mathrm{BASECLK}}/(2 \times \mathrm{DIV})$$

. If DIV = 0 then

$$F_{\mathrm{SDCLK}} = F_{\mathrm{BASECLK}}$$

– Programmable Clock Mode (CLKGSEL = 1):

$$F_{\mathrm{SDCLK}} = F_{\mathrm{MULTCLK}}/(\mathrm{DIV}{+}1)$$

This field depends on the setting of Preset Value Enable (PVALEN) in SDHC_HC2R.
If SDHC_HC2R.PVALEN = 0, this field is set by the user.
If SDHC_HC2R.PVALEN = 1, this field is automatically set to a value specified in one of the SDHC_PVR.

Bits 7:6 – USDCLKFSEL[1:0] Upper Bits of SDCLK Frequency Select

These bits expand the SDCLK Frequency Select (SDCLKFSEL) to 10 bits. These two bits are assigned to bit 09-08 of the clock divider as described in SDCLKFSEL.

Bit 5 – CLKGSEL Clock Generator Select

This bit is used to select the clock generator mode in the SDCLK Frequency Select field. If the Programmable mode is not supported (SDHC_CA1R.CLKMULT (Clock Multiplier) set to 0), then this bit cannot be written and is always read at 0.
This bit depends on the setting of Preset Value Enable (PVALEN) in SDHC_HC2R.
If SDHC_HC2R.PVALEN = 0, this bit is set by the user.
If SDHC_HC2R.PVALEN = 1, this bit is automatically set to a value specified in one of the SDHC_PVRx.

Value	Description
0	Divided Clock mode (BASECLK is used to generate SDCLK).

Value	Description
1	Programmable Clock mode (MULTCLK is used to generate SDCLK).

Bit 2 – SDCLKEN SD Clock Enable

The peripheral stops the SD Clock when writing this bit to 0. SDCLK Frequency Select (SDCLKFSEL) can be changed when this bit is 0. Then, the peripheral maintains the same clock frequency until SDCLK is stopped (Stop at SDCLK=0). If Card Inserted (CARDINS) in SDHC_PSR is cleared, this bit is also cleared.

Value	Description
0	SD Clock disabled
1	SD Clock enabled

Bit 1 – INTCLKS Internal Clock Stable

This bit is set to 1 when the SD clock is stable after setting SDHC_CCR.INTCLKEN (Internal Clock Enable) to 1. The user must wait to set SD Clock Enable (SDCLKEN) until this bit is set to 1.

Value	Description
0	Internal clock not ready
1	Internal clock ready

Bit 0 – INTCLKEN Internal Clock Enable

This bit is set to 0 when the peripheral is not used or is awaiting a wakeup interrupt. In this case, its internal clock is stopped to reach a very low power state. Registers are still able to be read and written. The clock starts to oscillate when this bit is set to 1. Once the clock oscillation is stable, the peripheral sets Internal Clock Stable (INTCLKS) in this register to 1.
This bit does not affect card detection.

Value	Description
0	The internal clock stops.
1	The internal clock oscillates.

41.8.15. SDHC Timeout Control Register

Name: TCR
Offset: 0x2E
Reset: 0x00
Property: -

Table 41-16. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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

Bits 3:0 – DTCVAL[3:0] Data Timeout Counter Value

This value determines the interval at which DAT line timeouts are detected. For more information about timeout generation, refer to Data Timeout Error (DATTEO) in SDHC_EISTR. When setting this register, the user can prevent inadvertent timeout events by clearing the Data Timeout Error Status Enable (in SDHC_EISTER).

$$\mathrm{TIMEOUT}_{(\mu\mathrm{S})} = \frac{2^{13 + \mathrm{DTCVAL}}}{F_{\mathrm{BASECLK(MHz)}}}$$

Note: DTCVAL = $F_{(Hexa)}$ is reserved.

41.8.16. SDHC Software Reset Register

Name: SRR
Offset: 0x2F
Reset: 0x00
Property: -

Table 41-17. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
						SWRSTDAT	SWRSTCMD	SWRSTALL
Access						R/W	R/W	R/W
Reset						0	0	0

Bit 2 – SWRSTDAT Software reset for DAT line
Only part of a data circuit is reset. The DMA circuit is also reset.
The following registers and bits are cleared by this bit:

- Buffer Data Port Register SDHC_BDPR: BUFDATA is cleared and initialized.
- Present State Register SDHC_PSR:
  - Buffer Read Enable (BUFRDEN)
  - Buffer Write Enable (BUFWREN)
  - Read Transfer Active (RTACT)
  - Write Transfer Active (WTACT)
  - DAT Line Active (DATLL)
  - Command Inhibit - DAT (CMDINHD)
- Block Gap Control Register SDHC_BGCR:
  - Continue Request (CONTR)
  - Stop At Block Gap Request (STPBGR)
- Normal Interrupt Status Register SDHC_NISTR:
  - Buffer Read Ready (BRDRDY)
  - Buffer Write Ready (BWRRDY)
  - DMA Interrupt (DMAINT)
  - Block Gap Event (BLKGE)
  - Transfer Complete (TRFC)

Value	Description
0	Work
1	Reset

Bit 1 – SWRSTCMD Software reset for CMD line
Only part of a command circuit is reset.
The following registers and bits are cleared by this bit:

- Present State Register :
  - Command Inhibit (CMD) (CMDINHC)

- Normal Interrupt Status Register :
  – Command Complete (CMDC)

Value	Description
0	Work
1	Reset

Bit 0 – SWRSTALL Software reset for All

This reset affects the entire peripheral except the card detection circuit. During initialization, the peripheral must be reset by setting this bit to 1. This bit is automatically cleared to 0 when SDHC_CA0R and SDHC_CA1R are valid and the user can read them. If this bit is set to 1, the user should issue a reset command and reinitialize the card.

List of registers cleared to 0:

- SDMA System Address / Argument 2 Register
- Block Size Register
- Block Count Register
- Argument 1 Register
- Transfer Mode Register
- Command Register
- Response Register n
- Buffer Data Port Register
- Present State Register (except CMDLL, DATLL, WRPPL, CARDDDPL, CARDSS, CARDINS)
- Host Control 1 Register
- Power Control Register
- Block Gap Control Register
- Wakeup Control Register
- Clock Control Register
- Timeout Control Register
- Normal Interrupt Status Register
- Error Interrupt Status Register
- Normal Interrupt Status Enable Register
- Error Interrupt Status Enable Register
- Normal Interrupt Signal Enable Register
- Error Interrupt Signal Enable Register
- Auto CMD Error Status Register
- Host Control 2 Register
- ADMA Error Status Register
- ADMA System Address Registers
- Slot Interrupt Status Register
- e.MMC Control 1 Register
- e.MMC Control 2 Register
- AHB Control Register
- Clock Control 2 Register
- Capabilities Control Register (except KEY)

Value	Description
0	Work
1	Reset

41.8.17. SDHC Normal Interrupt Status Register

Name: NISTR
Offset: 0x30
Reset: 0x0000
Property: -

Note: Interrupt flags must be cleared and then read back to confirm the clear before exiting the ISR to avoid double interrupts.

Table 41-18. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	15	14	13	12	11	10	9	8
	ERRINT	BOOTAR						CINT
Access	R/W	R/W						R/W
Reset	0	0						0

Bit	7	6	5	4	3	2	1	0
	CREM	CINS	BRDRDY	BWRRDY	DMAINT	BLKGE	TRFC	CMDC
Access	R/W	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 – ERRINT Error Interrupt

If any of the bits in SDHC_EISTR are set, then this bit is set. Therefore, the user can efficiently test for an error by checking this bit first. This bit is read-only.

Value	Description
0	No error
1	Error

Bit 14 – BOOTAR Boot Acknowledge Received

Note: This register entry is specific to the e.MMC operation mode.

This bit is set to 1 when the peripheral received a Boot Acknowledge pattern from the e.MMC.
This bit can only be set to 1 if SDHC_NISTER.BOOTAR is set to 1. An interrupt can only be generated if SDHC_NISIER.BOOTAR is set to 1.
Writing this bit to 1 clears this bit.

Value	Description
0	Boot Acknowledge pattern not received.
1	Boot Acknowledge pattern received.

Bit 8 – CINT Card Interrupt

Note: This register entry is specific to the SD/SDIO operation mode.

Writing this bit to 1 does not clear this bit. It is cleared by resetting the SD card interrupt factor. In 1-bit mode, the peripheral detects the Card Interrupt without SDCLK to support wake-up. In 4-bit mode, the Card Interrupt signal is sampled during the interrupt cycle, so there are some sample delays between the interrupt signal from the SD card and the interrupt to the system.
When this bit has been set to 1 and the user needs to start this interrupt service, Card Interrupt Status Enable (CINT) in SDHC_NISTER may be set to 0 in order to clear the card interrupt statuses latched in the peripheral and to stop driving the interrupt signal to the system. After completion of

the card interrupt service (it should reset interrupt factors in the SD card and the interrupt signal may not be asserted), set SDHC_NISTER.CINT to 1 and start sampling the interrupt signal again. Interrupt detected by DAT[1] is supported when there is one card per slot. In case of a shared bus, interrupt pins are used to detect interrupts. If 0 is set to Interrupt Pin Select (INTPSEL) in SDHC_SBCR, this status is effective. If a non-zero value is set to INTPSEL, INT_A, INT_B or INT_C is used as device interrupts.
This bit can only be set to 1 if SDHC_NISTER.CREM is set to 1. An interrupt can only be generated if SDHC_NISIER.CREM is set to 1.

Value	Description
0	No card interrupt
1	Card interrupt

Bit 7 – CREM Card Removal

Note: This register entry is specific to the SD/SDIO operation mode.

This status is set to 1 if Card Inserted (CARDINS) in SDHC_PSR changes from 1 to 0. When the user writes this bit to 1 to clear this status, the status of SDHC_PSR.CARDINS must be confirmed because the card detect state may possibly be changed when the user clears this bit and no interrupt event can be generated.
This bit can only be set to 1 if SDHC_NISTER.CREM is set to 1. An interrupt can only be generated if SDHC_NISIER.CREM is set to 1.
Writing this bit to 1 clears this bit.

Value	Description
0	Card state unstable or card inserted
1	Card removed

Bit 6 – CINS Card Insertion

Note: This register entry is specific to the SD/SDIO operation mode.

This status is set if Card Inserted (CARDINS) in SDHC_PSR changes from 0 to 1. When the user writes this bit to 1 to clear this status, the status of SDHC_PSR.CARDINS must be confirmed because the card detect state may possibly be changed when the user clears this bit and no interrupt event can be generated.
This bit can only be set to 1 if SDHC_NISTER.CINS is set to 1. An interrupt can only be generated if SDHC_NISIER.CINS is set to 1.
Writing this bit to 1 clears this bit.

Value	Description
0	Card state unstable or card removed
1	Card inserted

Bit 5 – BRDRDY Buffer Read Ready

This status is set to 1 if the Buffer Read Enable (BUFRDEN) changes from 0 to 1. Refer to BUFRDEN in SDHC_PSR.
This bit can only be set to 1 if SDHC_NISTER.BRDRDY is set to 1. An interrupt can only be generated if SDHC_NISIER.BRDRDY is set to 1.
Writing this bit to 1 clears this bit.

Value	Description
0	Not ready to read buffer
1	Ready to read buffer

Bit 4 – BWRRDY Buffer Write Ready

This status is set to 1 if the Buffer Write Enable (BUFWREN) changes from 0 to 1. Refer to BUFWREN in SDHC_PSR.

This bit can only be set to 1 if SDHC_NISTER.BWRRDY is set to 1. An interrupt can only be generated if SDHC_NISIER.BWRRDY is set to 1.
Writing this bit to 1 clears this bit.

Value	Description
0	Not ready to write buffer
1	Ready to write buffer

Bit 3 – DMAINT DMA Interrupt

This status is set if the peripheral detects the Host SDMA Buffer boundary during transfer. Refer to SDMA Buffer Boundary (BOUNDARY) in SDHC_BSR.
In case of ADMA, by setting the "int" field in the descriptor table, the peripheral rises this status flag when the descriptor line is completed. This status flag does not rise after Transfer Complete (TRFC).
This bit can only be set to 1 if SDHC_NISTER.DMAINT is set to 1. An interrupt can only be generated if SDHC_NISIER.DMAINT is set to 1.
Writing this bit to 1 clears this bit.

Value	Description
0	No DMA Interrupt
1	DMA Interrupt

Bit 2 – BLKGE Block Gap Event

If the Stop At Block Gap Request (STPBGR) in SDHC_BGCR is set to 1, this bit is set when either a read or a write transaction is stopped at a block gap. If STPBGR is not set to 1, this bit is not set to 1.
In the case of a Read transaction:
This bit is set at the falling edge of the DAT Line Active (DLACT) status (when the transaction is stopped at SD bus timing). The Read Wait must be supported in order to use this function. Refer to section "Read Transaction Wait / Continue Timing" in the "SD Host Controller Simplified Specification V3.00" about the detailed timing.
In the case of a Write transaction:
This bit is set at the falling edge of the Write Transfer Active (WTACT) status (after getting the CRC status at SD bus timing). Refer to section "Write Transaction Wait / Continue Timing" in the "SD Host Controller Simplified Specification V3.00" for more details on the sequence of events.
This bit can only be set to 1 if SDHC_NISTER.BLKGE is set to 1. An interrupt can only be generated if SDHC_NISIER.BLKGE is set to 1.
Writing this bit to 1 clears this bit.

Value	Description
0	No block gap event
1	Transaction stopped at block gap

Bit 1 – TRFC Transfer Complete

This bit is set when a read/write transfer and a command with Busy is completed.
In the case of a Read Transaction:
This bit is set at the falling edge of the Read Transfer Active Status. The interrupt is generated in two cases. The first is when a data transfer is completed as specified by the data length (after the last data has been read to the system). The second is when data has stopped at the block gap and completed the data transfer by setting the Stop At Block Gap Request (STPBGR) in SDHC_BGCR (after valid data has been read to the system). Refer to section "Read Transaction Wait / Continue Timing" in the "SD Host Controller Simplified Specification V3.00" for more details on the sequence of events.
In the case of a Write Transaction:
This bit is set at the falling edge of the DAT Line Active (DLACT) status. This interrupt is generated in two cases. The first is when the last data is written to the card as specified by the data length and the Busy signal is released. The second is when data transfers are stopped at the block gap by setting Stop At Block Gap Request (STPBGR) in SDHC_BGCR and data transfers are completed. (After valid data is written to the card and the Busy signal is released). Refer to section "Write Transaction

Wait / Continue Timing” in the “SD Host Controller Simplified Specification V3.00” for more details on the sequence of events.
In the case of command with Busy:
This bit is set when Busy is de-asserted. Refer to DAT Line Active (DLACT) and Command Inhibit (DAT) (CMDINHD) in SDHC_PSR.
This bit can only be set to 1 if SDHC_NISTER.TRFC is set to 1. An interrupt can only be generated if SDHC_NISIER.TRFC is set to 1.
Writing this bit to 1 clears this bit.
The table below shows that Transfer Complete (TRFC) has a higher priority than Data Timeout Error (DATTEO). If both bits are set to 1, execution of a command can be considered to be completed.

TRFC	DATTEO	Meaning of the status
0	0	Interrupted by another factor
0	1	Timeout occurred during transfer
1	Don’t Care	Command execution complete

Value	Description
0	Command execution is not complete.
1	Command execution is complete.

Bit 0 – CMDC Command Complete
This bit is set when getting the end bit of the command response. Auto CMD12 and Auto CMD23 consist of two responses. Command Complete is not generated by the response of CMD12 or CMD23, but it is generated by the response of a read/write command. Refer to Command Inhibit (CMD) in SDHC_PSR for details on how to control this bit.
This bit can only be set to 1 if SDHC_NISTER.CMDC is set to 1. An interrupt can only be generated if SDHC_NISIER.CMDC is set to 1.
Writing this bit to 1 clears this bit.
The table below shows that Command Timeout Error (CMDTEO) has a higher priority than Command Complete (CMDC). If both bits are set to 1, it can be considered that the response was not received correctly.

CMDC	CMDTEO	Meaning of the status
0	0	Interrupted by another factor
Don’t care	1	Response not received within 64 SDCLK cycles
1	0	Response received

Value	Description
0	No command complete
1	Command complete

41.8.18. SDHC Error Interrupt Status Register

Name: EISTR
Offset: 0x32
Reset: 0x0000
Property: -

Table 41-19. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	15	14	13	12	11	10	9	8
				BOOTAE		TUNING	ADMA	ACMD
Access				R/W		R/W	R/W	R/W
Reset				0		0	0	0

Bit	7	6	5	4	3	2	1	0
	CURLIM	DATEND	DATCRC	DATTEO	CMDIDX	CMDEND	CMDCRC	CMDTEO
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 12 – BOOTAE Boot Acknowledge Error

Note: This register entry is specific to the e.MMC operation mode.

This bit is set to 1 when detecting that the e.MMC Boot Acknowledge Status has a value other than "010".
This bit can only be set to 1 if SDHC_EISTER.BOOTAE is set to 1. An interrupt can only be generated if SDHC_EISIER.BOOTAE is set to 1.
Writing this bit to 1 clears this bit.

Value	Description
0	No error
1	Error

Bit 10 – TUNING Tuning Error

This bit is set to 1 when an unrecoverable error is detected in a tuning circuit, except during a tuning procedure (occurrence of an error during tuning procedure is indicated by Sampling Clock Select (SCLKSEL) in SDHC_HC2R).
When detecting a tuning error, the user needs to abort the execution of a command and to perform tuning. To reset the tuning circuit, SCLKSEL must be set to 0 before executing the tuning procedure (refer to Figure 2-29 in the "SD Host Controller Simplified Specification V3.00").
Tuning Error has a higher priority than the other error statuses generated during data transfer.
When detecting a tuning error, the user should discard any data transferred by a current read/write command and retry the transfer after the peripheral recovered from tuning circuit error.
This bit can only be set to 1 if SDHC_EISTER.TUNING is set to 1. An interrupt can only be generated if SDHC_EISIER. TUNING is set to 1.
Writing this bit to 1 clears this bit.

Value	Description
0	No error
1	Error

Bit 9 – ADMA ADMA Error

This bit is set to 1 when the peripheral detects errors during an ADMA-based data transfer. The state of the ADMA at an error occurrence is saved in SDHC_AESR.
In addition, the peripheral rises this status bit when it detects some invalid description data (Valid=0) at the ST_FDS state (refer to section "Advanced DMA" in the "SD Host Controller Simplified Specification V3.00". ADMA Error Status (ERRST) in SDHC_AESR indicates that an error occurs in ST_FDS state. The user may find that the Valid bit is not set at the error descriptor.
This bit can only be set to 1 if SDHC_EISTER.ADMA is set to 1. An interrupt can only be generated if SDHC_EISIER.ADMA is set to 1.
Writing this bit to 1 clears this bit.

Value	Description
0	No error
1	Error

Bit 8 – ACMD Auto CMD Error

Auto CMD12 and Auto CMD23 use this error status. This bit is set to 1 when detecting that one of the 0 to 4 bits in SDHC_AESR (SDHC_ACESR[4:0]) has changed from 0 to 1. In the case of Auto CMD12, this bit is set to 1, not only when errors occur in Auto CMD12, but also when Auto CMD12 is not executed due to the previous command error.
This bit can only be set to 1 if SDHC_EISTER.ACMD is set to 1. An interrupt can only be generated if SDHC_EISIER.ACMD is set to 1.
Writing this bit to 1 clears this bit.

Value	Description
0	No error
1	Error

Bit 7 – CURLIM Current Limit Error

By setting SD Bus Power (SDBPWR) in SDHC_PCR, the peripheral is requested to supply power for the SD Bus. The peripheral is protected from an illegal card by stopping power supply to the card, in which case this bit indicates a failure status. Reading 1 means the peripheral is not supplying power to the card due to some failure. Reading 0 means that the peripheral is supplying power and no error has occurred. The peripheral may require some sampling time to detect the current limit.
This bit can only be set to 1 if SDHC_EISTER.CURLIM is set to 1. An interrupt can only be generated if SDHC_EISIER.CURLIM is set to 1.
Writing this bit to 1 clears this bit.

Value	Description
0	No error
1	Error

Bit 6 – DATEND Data End Bit Error

This bit is set to 1 either when detecting 0 at the end bit position of read data which uses the DAT line or at the end bit position of the CRC Status.
This bit can only be set to 1 if SDHC_EISTER.DATEND is set to 1. An interrupt can only be generated if SDHC_EISIER.DATEND is set to 1.
Writing this bit to 1 clears this bit.

Value	Description
0	No error
1	Error

Bit 5 – DATCRC Data CRC Error

This bit is set to 1 when detecting a CRC error during a transfer of read data which uses the DAT line or when detecting that the Write CRC Status has a value other than '010'.

This bit can only be set to 1 if SDHC_EISTER. DATCRC is set to 1. An interrupt can only be generated if SDHC_EISIER. DATCRC is set to 1.
Writing this bit to 1 clears this bit.

Value	Description
0	No error
1	Error

Bit 4 – DATTEO Data Timeout error
This bit is set to 1 when detecting one of following timeout conditions:

- Busy timeout for R1b, R5b response type (see “Physical Layer Simplified Specification V3.01” and “SDIO Simplified Specification V3.00”).
- Busy timeout after Write CRC Status.
- Write CRC Status timeout.
- Read data timeout.

This bit can only be set to 1 if SDHC_EISTER.DATTEO is set to 1. An interrupt can only be generated if SDHC_EISIER.DATTEO is set to 1.
Writing this bit to 1 clears this bit.

Value	Description
0	No error
1	Error

Bit 3 – CMDIDX Command Index Error
This bit is set to 1 if a Command Index error occurs in the command response.
This bit can only be set to 1 if SDHC_EISTER.CMDIDX is set to 1. An interrupt can only be generated if SDHC_EISIER.CMDIDX is set to 1.
Writing this bit to 1 clears this bit.

Value	Description
0	No error
1	Error

Bit 2 – CMDEND Command End Bit Error
This bit is set to 1 when detecting that the end bit of a command response is 0.
This bit can only be set to 1 if SDHC_EISTER.CMDEND is set to 1. An interrupt can only be generated if SDHC_EISIER.CMDEND is set to 1.
Writing this bit to 1 clears this bit.

Value	Description
0	No error
1	Error

Bit 1 – CMDCRC Command CRC Error
The Command CRC Error is generated in two cases.
If a response is returned and Command Timeout Error (CMDTEO) is set to 0 (indicating no command timeout), this bit is set to 1 when detecting a CRC error in the command response.
The peripheral detects a CMD line conflict by monitoring the CMD line when a command is issued. If the peripheral drives the CMD line to 1 level, but detects 0 level on the CMD line at the next SDCLK edge, then the peripheral aborts the command (stops driving the CMD line) and sets this bit to 1. CMDTEO is also set to 1 to indicate a CMD line conflict.
This bit can only be set to 1 if SDHC_EISTER.CMDCRC is set to 1. An interrupt can only be generated if SDHC_EISIER.CMDCRC is set to 1.
Writing this bit to 1 clears this bit.

Bit 0 – CMDTEO Command Timeout Error

This bit is set to 1 only if no response is returned within 64 SDCLK cycles from the end bit of the command. If the peripheral detects a CMD line conflict, in which case Command CRC Error (CMDCRC) is also set to 1, this bit is set without waiting for 64 SDCLK cycles because the command is aborted by the peripheral.

This bit can only be set to 1 if SDHC_EISTER.CMDTEO is set to 1. An interrupt can only be generated if SDHC_EISIER.CMDTEO is set to 1.

Writing this bit to 1 clears this bit.

Table 41-20. CMD Error Types

CMDCRC	CMDTEO	Types of error
0	0	No error
0	1	Response timeout error
1	0	Response CRC error
1	1	CMD line conflict

41.8.19. SDHC Normal Interrupt Status Enable Register: e.MMC

Name: NISTER
Offset: 0x34
Reset: 0x0000
Property: -

Table 41-21. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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

Bit	7	6	5	4	3	2	1	0
	CREM	CINS	BRDRDY	BWRRDY	DMAINT	BLKGE	TRFC	CMDC
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 14 – BOOTAR Boot Acknowledge Received Status Enable

Note: This register entry is specific to the e.MMC operation mode.

Value	Name	Description
0	MASKED	The BOOTAR status flag in SDHC_NISTR is masked.
1	ENABLED	The BOOTAR status flag in SDHC_NISTR is enabled.

Bit 7 – CREM Card Removal Status Enable

Value	Name	Description
0	MASKED	The CREM status flag in SDHC_NISTR is masked.
1	ENABLED	The CREM status flag in SDHC_NISTR is enabled.

Bit 7 – CINT Card Interrupt Status Enable

If this bit is set to 0, the peripheral clears interrupt requests to the system. The Card Interrupt detection is stopped when this bit is cleared and restarted when this bit is set to 1. The user may clear this bit before servicing the Card Interrupt and may set this bit again after all interrupt requests from the card are cleared to prevent inadvertent interrupts.

Value	Name	Description
0	MASKED	The CINT status flag in SDHC_NISTR is masked.
1	ENABLED	The CINT status flag in SDHC_NISTR is enabled.

Bit 6 – CINS Card Insertion Status Enable

Value	Name	Description
0	MASKED	The CINS status flag in SDHC_NISTR is masked.
1	ENABLED	The CINS status flag in SDHC_NISTR is enabled.

Bit 5 – BRDRDY Buffer Read Ready Status Enable

Value	Name	Description
0	MASKED	The BRDRDY status flag in SDHC_NISTR is masked.

Value	Name	Description
1	ENABLED	The BRDRDY status flag in SDHC_NISTR is enabled.

Bit 4 – BWRRDY Buffer Write Ready Status Enable

Value	Name	Description
0	MASKED	The BWRRDY status flag in SDHC_NISTR is masked.
1	ENABLED	The BWRRDY status flag in SDHC_NISTR is enabled.

Bit 3 – DMAINT DMA Interrupt Status Enable

Value	Name	Description
0	MASKED	The DMAINT status flag in SDHC_NISTR is masked.
1	ENABLED	The DMAINT status flag in SDHC_NISTR is enabled.

Bit 2 – BLKGE Block Gap Event Status Enable

Value	Name	Description
0	MASKED	The BLKGE status flag in SDHC_NISTR is masked.
1	ENABLED	The BLKGE status flag in SDHC_NISTR is enabled.

Bit 1 – TRFC Transfer Complete Status Enable

Value	Name	Description
0	MASKED	The TRFC status flag in SDHC_NISTR is masked.
1	ENABLED	The TRFC status flag in SDHC_NISTR is enabled.

Bit 0 – CMDC Command Complete Status Enable

Value	Name	Description
0	MASKED	The CMDC status flag in SDHC_NISTR is masked.
1	ENABLED	The CMDC status flag in SDHC_NISTR is enabled.

41.8.20. SDHC Error Interrupt Status Enable Register

Name: EISTER
Offset: 0x36
Reset: 0x0000
Property: -

Table 41-22. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	15	14	13	12	11	10	9	8
				BOOTAE		TUNING	ADMA	ACMD
Access				R/W		R/W	R/W	R/W
Reset				0		0	0	0

Bit	7	6	5	4	3	2	1	0
	CURLIM	DATEND	DATCRC	DATTEO	CMDIDX	CMDEND	CMDCRC	CMDTEO
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 12 – BOOTAE Boot Acknowledge Error Status Enable
Note: This register entry is specific to the e.MMC operation mode.

Value	Name	Description
0	MASKED	The BOOTAE status flag in SDHC_EISTR is masked.
1	ENABLED	The BOOTAE status flag in SDHC_EISTR is enabled.

Bit 10 – TUNING Tuning Error Status Enable

Value	Name	Description
0	MASKED	The TUNING status flag in SDHC_EISTR is masked.
1	ENABLED	The TUNING status flag in SDHC_EISTR is enabled.

Bit 9 – ADMA ADMA Error Status Enable

Value	Name	Description
0	MASKED	The ADMA status flag in SDHC_EISTR is masked.
1	ENABLED	The ADMA status flag in SDHC_EISTR is enabled.

Bit 8 – ACMD Auto CMD Error Status Enable

Value	Name	Description
0	MASKED	The ACMD status flag in SDHC_EISTR is masked.
1	ENABLED	The ACMD status flag in SDHC_EISTR is enabled.

Bit 7 – CURLIM Current Limit Error Status Enable

Value	Name	Description
0	MASKED	The CURLIM status flag in SDHC_EISTR is masked.
1	ENABLED	The CURLIM status flag in SDHC_EISTR is enabled.

Bit 6 – DATEND Data End Bit Error Status Enable

Value	Name	Description
0	MASKED	The DATEND status flag in SDHC_EISTR is masked.
1	ENABLED	The DATEND status flag in SDHC_EISTR is enabled.

Bit 5 – DATCRC Data CRC Error Status Enable

Value	Name	Description
0	MASKED	The DATCRC status flag in SDHC_EISTR is masked.
1	ENABLED	The DATCRC status flag in SDHC_EISTR is enabled.

Bit 4 – DATTEO Data Timeout Error Status Enable

Value	Name	Description
0	MASKED	The DATTEO status flag in SDHC_EISTR is masked.
1	ENABLED	The DATTEO status flag in SDHC_EISTR is enabled.

Bit 3 – CMDIDX Command Index Error Status Enable

Value	Name	Description
0	MASKED	The CMDIDX status flag in SDHC_EISTR is masked.
1	ENABLED	The CMDIDX status flag in SDHC_EISTR is enabled.

Bit 2 – CMDEND Command End Bit Error Status Enable

Value	Name	Description
0	MASKED	The CMDEND status flag in SDHC_EISTR is masked.
1	ENABLED	The CMDEND status flag in SDHC_EISTR is enabled.

Bit 1 – CMDCRC Command CRC Error Status Enable

Value	Name	Description
0	MASKED	The CMDCRC status flag in SDHC_EISTR is masked.
1	ENABLED	The CMDCRC status flag in SDHC_EISTR is enabled.

Bit 0 – CMDTEO Command Timeout Error Status Enable

Value	Name	Description
0	MASKED	The CMDTEO status flag in SDHC_EISTR is masked.
1	ENABLED	The CMDTEO status flag in SDHC_EISTR is enabled.

41.8.21. SDHC Normal Interrupt Signal Enable Register

Name: NISIER
Offset: 0x38
Reset: 0x0000
Property: -

Table 41-23. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	15	14	13	12	11	10	9	8
		BOOTAR						CINT
Access		R/W						R/W
Reset		0						0

Bit	7	6	5	4	3	2	1	0
	CREM	CINS	BRDRDY	BWRRDY	DMAINT	BLKGE	TRFC	CMDC
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 14 – BOOTAR Boot Acknowledge Received Signal Enable
Note: This register entry is specific to the e.MMC operation mode.

Value	Name	Description
0	MASKED	No interrupt is generated when SDHC_NISTR.BOOTAR is set.
1	ENABLED	An interrupt is generated when SDHC_NISTR.BOOTAR is set.

Bit 8 – CINT Card Interrupt Signal Enable
Note: This register entry is specific to the SD/SDIO operation mode.

Value	Name	Description
0	MASKED	No interrupt is generated when SDHC_NISTR.CINT is set.
1	ENABLED	An interrupt is generated when SDHC_NISTR.CINT is set.

Bit 7 – CREM Card Removal Signal Enable
Note: This register entry is specific to the SD/SDIO operation mode.

Value	Name	Description
0	MASKED	No interrupt is generated when SDHC_NISTR.CREM is set.
1	ENABLED	An interrupt is generated when SDHC_NISTR.CREM is set.

Bit 6 – CINS Card Insertion Signal Enable
Note: This register entry is specific to the SD/SDIO operation mode.

Value	Name	Description
0	MASKED	No interrupt is generated when SDHC_NISTR.CINS is set.
1	ENABLED	An interrupt is generated when SDHC_NISTR.CINS is set.

Bit 5 – BRDRDY Buffer Read Ready Signal Enable

Value	Name	Description
0	MASKED	No interrupt is generated when SDHC_NISTR.BRDRDY is set.
1	ENABLED	An interrupt is generated when SDHC_NISTR.BRDRDY is set.

Bit 4 – BWRRDY Buffer Write Ready Signal Enable

Value	Name	Description
0	MASKED	No interrupt is generated when SDHC_NISTR.BWRRDY is set.
1	ENABLED	An interrupt is generated when SDHC_NISTR.BWRRDY is set.

Bit 3 – DMAINT DMA Interrupt Signal Enable

Value	Name	Description
0	MASKED	No interrupt is generated when SDHC_NISTR.DMAINT is set.
1	ENABLED	An interrupt is generated when SDHC_NISTR.DMAINT is set.

Bit 2 – BLKGE Block Gap Event Signal Enable

Value	Name	Description
0	MASKED	No interrupt is generated when SDHC_NISTR.BLKGE is set.
1	ENABLED	An interrupt is generated when SDHC_NISTR.BLKGE is set.

Bit 1 – TRFC Transfer Complete Signal Enable

Value	Name	Description
0	MASKED	No interrupt is generated when SDHC_NISTR.TRFC is set.
1	ENABLED	An interrupt is generated when SDHC_NISTR.TRFC is set.

Bit 0 – CMDC Command Complete Signal Enable

Value	Name	Description
0	MASKED	No interrupt is generated when SDHC_NISTR.CMDC is set.
1	ENABLED	An interrupt is generated when SDHC_NISTR.CMDC is set.

41.8.22. SDHC Error Interrupt Signal Enable Register

Name: EISIER
Offset: 0x3A
Reset: 0x0000
Property: -

Table 41-24. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	15	14	13	12	11	10	9	8
				BOOTAE		TUNING	ADMA	ACMD
Access				R/W		R/W	R/W	R/W
Reset				0		0	0	0

Bit	7	6	5	4	3	2	1	0
	CURLIM	DATEND	DATCRC	DATTEO	CMDIDX	CMDEND	CMDCRC	CMDTEO
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 12 – BOOTAE Boot Acknowledge Error Signal Enable
Note: This register entry is specific to the e.MMC operation mode.

Value	Name	Description
0	MASKED	No interrupt is generated when SDHC_EISTR.BOOTAE is set.
1	ENABLED	An interrupt is generated when SDHC_EISTR.BOOTAE is set.

Bit 10 – TUNING Tuning Error Signal Enable

Value	Name	Description
0	MASKED	No interrupt is generated when SDHC_EISTR.TUNING is set.
1	ENABLED	An interrupt is generated when SDHC_EISTR.TUNING is set.

Bit 9 – ADMA ADMA Error Signal Enable

Value	Name	Description
0	MASKED	No interrupt is generated when SDHC_EISTR.ADMA is set.
1	ENABLED	An interrupt is generated when SDHC_EISTR.ADMA is set.

Bit 8 – ACMD Auto CMD Error Signal Enable

Value	Name	Description
0	MASKED	No interrupt is generated when SDHC_EISTR.ACMD is set.
1	ENABLED	An interrupt is generated when SDHC_EISTR.ACMD is set.

Bit 7 – CURLIM Current Limit Error Signal Enable

Value	Name	Description
0	MASKED	No interrupt is generated when SDHC_EISTR.CURLIM is set.
1	ENABLED	An interrupt is generated when SDHC_EISTR.CURLIM is set.

Bit 6 – DATEND Data End Bit Error Signal Enable

Value	Name	Description
0	MASKED	No interrupt is generated when SDHC_EISTR.DATEND is set.
1	ENABLED	An interrupt is generated when SDHC_EISTR.DATEND is set.

Bit 5 – DATCRC Data CRC Error Signal Enable

Value	Name	Description
0	MASKED	No interrupt is generated when SDHC_EISTR.DATCRC is set.
1	ENABLED	An interrupt is generated when SDHC_EISTR.DATCRC is set.

Bit 4 – DATTEO Data Timeout Error Signal Enable

Value	Name	Description
0	MASKED	No interrupt is generated when SDHC_EISTR.DATTEO is set.
1	ENABLED	An interrupt is generated when SDHC_EISTR.DATTEO is set.

Bit 3 – CMDIDX Command Index Error Signal Enable

Value	Name	Description
0	MASKED	No interrupt is generated when SDHC_EISTR.CMDIDX is set.
1	ENABLED	An interrupt is generated when SDHC_EISTR.CMDIDX is set.

Bit 2 – CMDEND Command End Bit Error Signal Enable

Value	Name	Description
0	MASKED	No interrupt is generated when SDHC_EISTR.CMDEND is set.
1	ENABLED	An interrupt is generated when SDHC_EISTR.CMDEND is set.

Bit 1 – CMDCRC Command CRC Error Signal Enable

Value	Name	Description
0	MASKED	No interrupt is generated when SDHC_EISTR.CMDCRC is set.
1	ENABLED	An interrupt is generated when SDHC_EISTR.CMDCRC is set.

Bit 0 – CMDTEO Command Timeout Error Signal Enable

Value	Name	Description
0	MASKED	No interrupt is generated when SDHC_EISTR.CMDTEO is set.
1	ENABLED	An interrupt is generated when SDHC_EISTR.CMDTEO is set.

41.8.23. SDHC Auto CMD Error Status Register

Name: ACESR
Offset: 0x3C
Reset: 0x0000
Property: -

Table 41-25. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	CMDNI			ACMDIDX	ACMDEND	ACMDCRC	ACMDTEO	ACMD12NE
Access	R			R	R	R	R	R
Reset	0			0	0	0	0	0

Bit 7 – CMDNI Command Not Issued by Auto CMD12 Error

Setting this bit to 1 means CMD_wo_DAT is not executed due to an Auto CMD12 error (SDHC_ACESR[4:1]). This bit is set to 0 when Auto CMD Error is generated by Auto CMD23.

Value	Description
0	No error
1	Error

Bit 4 – ACMDIDX Auto CMD Index Error

This bit is set to 1 when the Command Index error occurs in response to a command.

Value	Description
0	No error
1	Error

Bit 3 – ACMDEND Auto CMD End Bit Error

This bit is set to 1 when detecting that the end bit of the command response is 0.

Value	Description
0	No error
1	Error

Bit 2 – ACMDCRC Auto CMD CRC Error

This bit is set to 1 when detecting a CRC error in the command response.

Bit 1 – ACMDTEO Auto CMD Timeout Error

This bit is set to 1 if no response is returned within 64 SDCLK cycles from the end bit of the command. If this bit is set to 1, the other error status bits (SDHC_ACESR[4:2]) are meaningless.

ACMDCRC	ACMDTEO	Types of error
0	0	No error
0	1	Response Timeout error

ACMDTEO (continued)		
ACMDCRC	ACMDTEO	Types of error
1	0	Response CRC error
1	1	CMD line conflict

Bit 0 – ACMD12NE Auto CMD12 Not Executed

If a memory multiple block data transfer is not started due to a command error, this bit is not set to 1 because it is not necessary to issue Auto CMD12. Setting this bit to 1 means the peripheral cannot issue Auto CMD12 to stop a memory multiple block data transfer due to some error. If this bit is set to 1, other error status bits (SDHC_ACESR[4:1]) are meaningless.

This bit is set to 0 when an Auto CMD error is generated by Auto CMD23.

Value	Description
0	No error
1	Error

41.8.24. SDHC Host Control 2 Register: SD/SDIO

Name: HC2R
Offset: 0x3E
Reset: 0x0000
Property: -

Note: The content of the SDHC_HC2R register is depending on the mode. This description is for SD/SDIO mode.

Table 41-26. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	15	14	13	12	11	10	9	8
	PVALEN	ASINTEN						
Access	R/W	R/W						
Reset	0	0						

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bit 15 – PVALEN Preset Value Enable

As the operating SDCLK frequency depend on the system implementation, it is difficult to determine these parameters in the standard host driver. When Preset Value Enable (PVALEN) is set to 1, automatic SDCLK frequency generation performed without considering system-specific conditions. This bit enables the functions defined in SDHC_PVR.

If this bit is written to 0, the Clock Generator Select bit (SDHC_CCR.CLKGSEL) and the SDCLK Frequency Select bit (SDHC_CCR.SDCLKFSEL) in the Clock Control Register (SDHC_CCR) are selected by the user.

If this bit is set to 1, SDHC_CCR.SDCLKFSEL and SDHC_CCR.CLKGSEL are set by the peripheral as specified in the Preset Value Register (SDHC_PVR).

Value	Description
0	SDHC_CCR.SDCLK, SDHC_CCR.SDCLKFSEL controlled by the user.
1	Automatic selection by Preset Value is enabled.

Bit 14 – ASINTEN Asynchronous Interrupt Enable

This bit can be set to 1 if a card support asynchronous interrupts and Asynchronous Interrupt Support (ASINTSUP) is set to 1 in SDHC_CA0R. Asynchronous interrupt is effective when DAT[1] interrupt is used in 4-bit SD mode. If this bit is set to 1, the user can stop the SDCLK during the asynchronous interrupt period to save power. During this period, the peripheral continues to deliver the Card Interrupt to the host when it is asserted by the card.

Value	Description
0	Disabled
1	Enabled

41.8.25. SDHC Capabilities 0 Register

Name: CA0R
Offset: 0x40
Reset: 0x27E80080
Property: -

Note: The Capabilities 0 Register is not supposed to be written by the user.

Table 41-27. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	SLTYPE[1:0]		ASINTSUP	SB64SUP			V30VSUP	V33VSUP
Access	R/W	R/W	R/W	R/W			R/W	R/W
Reset	0	0	1	0			1	1

Bit	23	22	21	20	19	18	17	16
	SRSUP	SDMASUP	HSSUP		ADMA2SUP	ED8SUP		
Access	R/W	R/W	R/W		R/W	R/W		
Reset	1	1	1		1	0		

Bit	15	14	13	12	11	10	9	8
	BASECLKF[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
	TEOCLKU		TEOCLKF[5:0]					
Access	R/W		R/W	R/W	R/W	R/W	R/W	R/W
Reset	1		0	0	0	0	0	0

Bits 31:30 – SLTYPE[1:0] Slot Type

This field indicates usage of a slot by a specific system. An peripheral control register set is defined per slot.
Embedded Slot for One Device means that only one non-removable device is connected to a bus slot.
The Standard Host Driver controls a removable card (SLTYPE = 0) or one embedded device (SLTYPE = 1) connected to an SD bus slot.

Value	Name
0	Removable Card Slot
1	Embedded Slot for One Device
2	Shared Bus Slot
2	Reserved
3	Reserved

Bit 29 – ASINTSUP Asynchronous Interrupt Support

Refer to section “Asynchronous Interrupt” in the “SDIO Simplified Specification V3.00”.

Value	Description
0	Asynchronous interrupt not supported

Value	Description
1	Asynchronous interrupt supported

Bit 28 – SB64SUP 64-Bit System Bus Support

This bit indicates if the peripheral supports the 64-bit Address Descriptor mode and is connected to the 64-bit address system bus.

Value	Description
0	64-bit address bus not supported
1	64-bit address bus supported

Bit 25 – V30VSUP Voltage Support 3.0V

Note: The signal and supply voltages of the peripheral are limited by the supply voltage of the device.

Value	Description
0	3.0V Voltage supply not supported
1	3.0V Voltage supply supported

Bit 24 – V33VSUP Voltage Support 3.3V

Note: The signal and supply voltages of the peripheral are limited by the supply voltage of the device.

Value	Description
0	3.3V Voltage supply not supported
1	3.3V Voltage supply supported

Bit 23 – SRSUP Suspend/Resume Support

This bit indicates whether the peripheral supports the Suspend/Resume functionality. If this bit is set to 0, the user does not issue either Suspend or Resume commands because the Suspend and Resume mechanism (refer to “Suspend and Resume Mechanism” in the “SD Host Controller Simplified Specification V3.00”) is not supported.

Value	Description
0	Suspend/Resume not supported
1	Suspend/Resume supported

Bit 22 – SDMASUP SDMA Support

This bit indicates whether the peripheral is capable of using SDMA to transfer data between system memory and the peripheral directly.

Value	Description
0	SDMA not supported
1	SDMA supported

Bit 21 – HSSUP High Speed Support

This bit indicates whether the peripheral and the system support High Speed mode and they can supply SDCLK frequency from 25MHz to 50MHz.

Value	Description
0	High Speed not supported
1	High Speed supported

Bit 19 – ADMA2SUP ADMA2 Support

This bit indicates whether the peripheral is capable of using ADMA2.

Value	Description
0	ADMA2 not supported

Value	Description
1	ADMA2 supported

Bit 18 – ED8SUP 8-Bit Support for Embedded Device

This bit indicates whether the peripheral is capable of using the 8-bit Bus Width mode.

Value	Description
0	8-bit bus width not supported
1	8-bit bus width supported

Bits 15:8 – BASECLKF[7:0] Base Clock Frequency

This value indicates the frequency of the base clock (BASECLK). The user uses this value to calculate the clock divider value (refer to SDCLK Frequency Select (SDCLKFSEL) in SDHC_CCR).
If this field is set to 0, the user must get the information via another method.

$F_{\mathrm{BASECLK}} = \mathrm{BASECLKF}_{\mathrm{MHz}}$

Bit 7 – TEOCLKU Timeout Clock Unit

This bit shows the unit of the base clock frequency used to detect Data Timeout Error.

Value	Description
0	kHz
1	MHz

Bits 5:0 – TEOCLKF[5:0] Timeout Clock Frequency

This bit shows the timeout clock frequency (TEOCLK) used to detect Data Timeout Error.
If this field is set to 0, the user must get the information via another method.
The Timeout Clock Unit (TEOCLKU) defines the unit of this field’s value.

– TEOCLKU = 0:

$F_{\mathrm{TEOCLK}} = \mathrm{TEOCLKF}_{\mathrm{KHz}}$

– TEOCLKU = 1:

$F_{\mathrm{TEOCLK}} = \mathrm{TEOCLKF}_{\mathrm{MHz}}$

41.8.26. SDHC Capabilities 1 Register

Name: CA1R
Offset: 0x44
Reset: 0x00000070
Property: -

Note: The Capabilities 1 Register is not supposed to be written by the user.

Table 41-28. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	CLKMULT[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
		DRVDSUP	DRVCSUP	DRVASUP		DDR50SUP	SDR104SUP	SDR50SUP
Access		R/W	R/W	R/W		R/W	R/W	R/W
Reset		0	0	0		0	0	0

Bits 23:16 – CLKMULT[7:0] Clock Multiplier

This field indicates the multiplier factor between the Base Clock (BASECLK) used for the Divided Clock Mode and the Multiplied Clock (MULTCLK) used for the Programmable Clock mode (refer to SDHC_CCR).

Reading this field to 0 means that the Programmable Clock mode is not supported.

$F_{\mathrm{MULTCLK}} = F_{\mathrm{BASECLK}} \times (\mathrm{CLKMULT}+1)$

Bit 6 – DRVDSUP Driver Type D Support

Value	Description
0	Driver type D is not supported.

Bit 5 – DRVCSUP Driver Type C Support

Value	Description
0	Driver type C is not supported.

Bit 4 – DRVASUP Driver Type A Support

Value	Description
0	Driver type A is not supported.

Bit 2 – DDR50SUP DDR50 Support

Value	Description
0	DDR50 mode is not supported.

Bit 1 – SDR104SUP SDR104 Support

Value	Description
0	SDR104 mode is not supported.
1	SDR104 mode is supported.

Bit 0 – SDR50SUP SDR50 Support

Value	Description
0	SDR50 mode is not supported.
1	SDR50 mode is supported.

41.8.27. SDHC Maximum Current Capabilities Register

Name: MCCAR
Offset: 0x48
Reset: 0x00000000
Property: -

Table 41-29. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
	MAXCUR30V[7:0]							
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
	MAXCUR33V[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 15:8 – MAXCUR30V[7:0] Maximum Current for 3.0V

This field indicates the maximum current capability for 3.0V voltage. This value is meaningful only if V30VSUP is set to 1 in SDHC_CA0R. Reading MAXCUR30V at 0 means that the user must get information via another method.

$\mathrm{Imax}_{\mathrm{mA}} = 4 \times \mathrm{MAXCURR30}V$

Bits 7:0 – MAXCUR33V[7:0] Maximum Current for 3.3V

This field indicates the maximum current capability for 3.3V voltage. This value is meaningful only if V33VSUP is set to 1 in SDHC_CA0R. Reading MAXCUR33V at 0 means that the user must get information via another method.

$\mathrm{Imax}_{\mathrm{mA}} = 4 \times \mathrm{MAXCURR33}V$

41.8.28. SDHC Force Event Register for Auto CMD Error Status

Name: FERACES
Offset: 0x50
Reset: 0x0000
Property: -

Table 41-30. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	CMDNI			ACMDIDX	ACMDEND	ACMDCRC	ACMDTEO	ACMD12NE
Access	W			W	W	W	W	W
Reset	0			0	0	0	0	0

Bit 7 – CMDNI Force Event for Command Not Issued by Auto CMD12 Error
For testing purposes, the user can write this bit to 1 to rise the CMDNI status flag in SDHC_ACESR.
Writing this bit to 0 has no effect.

Bit 4 – ACMDIDX Force Event for Auto CMD Index Error
For testing purposes, the user can write this bit to 1 to rise the ACMDIDX status flag in SDHC_ACESR.
Writing this bit to 0 has no effect.

Bit 3 – ACMDEND Force Event for Auto CMD End Bit Error
For testing purposes, the user can write this bit to 1 to rise the ACMDEND status flag in SDHC_ACESR.
Writing this bit to 0 has no effect.

Bit 2 – ACMDCRC Force Event for Auto CMD CRC Error
For testing purposes, the user can write this bit to 1 to rise the ACMDCRC status flag in SDHC_ACESR.
Writing this bit to 0 has no effect.

Bit 1 – ACMDTEO Force Event for Auto CMD Timeout Error
For testing purposes, the user can write this bit to 1 to rise the ACMDTEO status flag in SDHC_ACESR.
Writing this bit to 0 has no effect.

Bit 0 – ACMD12NE Force Event for Auto CMD12 Not Executed
For testing purposes, the user can write this bit to 1 to rise the ACMD12NE status flag in SDHC_ACESR.
Writing this bit to 0 has no effect.

41.8.29. SDHC Force Event Register for Error Interrupt Status

Name: FEREIS
Offset: 0x52
Reset: 0x0000
Property: -

Table 41-31. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	15	14	13	12	11	10	9	8
				BOOTAE			ADMA	ACMD
Access				W			W	W
Reset				0			0	0

Bit	7	6	5	4	3	2	1	0
	CURLIM	DATEND	DATCRC	DATTEO	CMDIDX	CMDEND	CMDCRC	CMDTEO
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit 12 – BOOTAE Force Event for Boot Acknowledge Error
For testing purposes, the user can write this bit to 1 to rise the BOOTAE status flag in SDHC_EISTR. Writing this bit to 0 has no effect.

Bit 9 – ADMA Force Event for ADMA Error
For testing purposes, the user can write this bit to 1 to rise the ADMA status flag in SDHC_EISTR. Writing this bit to 0 has no effect.

Bit 8 – ACMD Force Event for Auto CMD Error
For testing purposes, the user can write this bit to 1 to rise the ACMD status flag in SDHC_EISTR. Writing this bit to 0 has no effect.

Bit 7 – CURLIM Force Event for Current Limit Error
For testing purposes, the user can write this bit to 1 to rise the CURLIM status flag in SDHC_EISTR. Writing this bit to 0 has no effect.

Bit 6 – DATEND Force Event for Data End Bit Error
For testing purposes, the user can write this bit to 1 to rise the DATEND status flag in SDHC_EISTR. Writing this bit to 0 has no effect.

Bit 5 – DATCRC Force Event for Data CRC error
For testing purposes, the user can write this bit to 1 to rise the DATCRC status flag in SDHC_EISTR. Writing this bit to 0 has no effect.

Bit 4 – DATTEO Force Event for Data Timeout error
For testing purposes, the user can write this bit to 1 to rise the DATTEO status flag in SDHC_EISTR. Writing this bit to 0 has no effect.

Bit 3 – CMDIDX Force Event for Command Index Error
For testing purposes, the user can write this bit to 1 to rise the CMDIDX status flag in SDHC_EISTR.

Writing this bit to 0 has no effect.

Bit 2 – CMDEND Force Event for Command End Bit Error
For testing purposes, the user can write this bit to 1 to rise the CDMEND status flag in SDHC_EISTR.
Writing this bit to 0 has no effect.

Bit 1 – CMDCRC Force Event for Command CRC Error
For testing purposes, the user can write this bit to 1 to rise the CMDCRC status flag in SDHC_EISTR.
Writing this bit to 0 has no effect.

Bit 0 – CMDTEO Force Event for Command Timeout Error
For testing purposes, the user can write this bit to 1 to rise the CMDTEO status flag in SDHC_EISTR.
Writing this bit to 0 has no effect.

41.8.30. SDHC ADMA Error Status Register

Name: AESR
Offset: 0x54
Reset: 0x00
Property: -

Table 41-32. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
						LMIS	ERRST[1:0]	
Access						R	R	R
Reset						0	0	0

Bit 2 – LMIS ADMA Length Mismatch Error

This error occurs in the following two cases:

- While Block Count Enable (BCEN) is being set, the total data length specified by the Descriptor table is different from that specified by the Block Count (BLKCNT) and Transfer Block Size (BLKSIZE).
- The total data length cannot be divided by the Transfer Block Size (BLKSIZE).

Value	Description
0	No error
1	Error

Bits 1:0 – ERRST[1:0] ADMA Error State

This field indicates the state of ADMA when an error has occurred during an ADMA data transfer. This field never indicates 2 because ADMA never stops in this state.

Value	Name	Description
0x0	ST_STOP (Stop DMA)	Points to the descriptor following the error descriptor
0x1	ST_FDS (Fetch Descriptor)	Points to the error descriptor
0x2	-	Reserved
0x3	ST_TRF (Transfer Data)	Points to the descriptor following the error descriptor

41.8.31. SDHC ADMA System Address Register

Name: ASARx
Offset: 0x58
Reset: 0x00000000
Property: -

Table 41-33. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	ADMASA[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
	ADMASA[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
	ADMASA[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
	ADMASA[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 – ADMASA[31:0] ADMA System Address

This field holds the byte address of the executing command of the descriptor table. At the start of ADMA, the user must set the start address of the descriptor table. The ADMA increments this register address, which points to the next Descriptor line to be fetched.

When the ADMA Error (ADMA) status flag rises, this field holds a valid descriptor address depending on the ADMA Error State (ERRST). The user must program Descriptor Table on 32-bit boundary and set 32-bit boundary address to this register. ADMA2 ignores the lower 2 bits of this register and assumes it to be 0.

41.8.32. SDHC Preset Value Register

Name: PVRx
Offset: 0x60 + x*0x02 [x=0..7]
Reset: 0x0000
Property: R/W

One of the Preset Value Registers is effective based on the selected bus speed mode. The table below defines the conditions to select one of the SDHC_PVRs.

Table 41-34. Preset Value Register Select Condition

Selected Bus Speed Mode	VS18EN (SDHC_HC2R)	HSEN (SDHC_HC1R)	UHSMS (SDHC_HC2R)
Default Speed	0	0	don't care
High Speed	0	Response Timeout Error	don't care
Reserved	1	don't care	Other values

The following table shows the effective Preset Value Register according to the Selected Bus Speed mode.

Table 41-35. Preset Value Registers

SDHC_PVRx	Selected Bus Speed Mode	Signal Voltage
SDHC_PVR0	Initialization	3.3V or 1.8V
SDHC_PVR1	Default Speed	3.3V
SDHC_PVR2	High Speed	3.3V

When Preset Value Enable (PVALEN) in SDHC_HC2R is set to 1, SDCLK Frequency Select (SDLCKFSEL) and Clock Generator Select (CLKGSEL) in SDHC_CCR are automatically set based on the Selected Bus Speed mode. This means that the user does not need to set these fields when preset is enabled.
A Preset Value Register for Initialization (SDHC_PVR0) is not selected by Bus Speed mode. Before starting the initialization sequence, the user needs to set a clock preset value to SDCLKFSEL in SDHC_CCR. PVALEN can be set to 1 after the initialization is completed.

Note: Preset Values in SDHC_PVRx registers are not supposed to be written by the user. However, the user can modify preset values only if Capabilities Write Enable (CAPWREN) is written to 1 in SDHC_CACR.

Table 41-36. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	15	14	13	12	11	10	9	8
						CLKGSEL	SDCLKFSEL[9:8]	
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	7	6	5	4	3	2	1	0
	SDCLKFSEL[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 10 – CLKGSEL Clock Generator Select
Refer to CGGSEL in SDHC_CCR.

Bits 9:0 – SDCLKFSEL[9:0] SDCLK Frequency Select
Refer to SDCLKFSEL in SDHC_CCR.

41.8.33. SDHC Slot Interrupt Status Register

Name: SISR
Offset: 0xFC
Reset: 0x0000
Property: -

Table 41-37. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

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

Bits 7:0 – INTSSL[7:0] Interrupt Signal for Each Slot
These status bits indicate the logical OR of Interrupt Signals and WakeUp Signal for each peripheral instance in the device. INTSSL[x] corresponds to instance SDHCx. There are 2 instances in this device.

41.8.34. SDHC Host Controller Version Register

Name: HCVR
Offset: 0xFE
Reset: 0x1802
Property: -

Table 41-38. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	15	14	13	12	11	10	9	8
	VVER[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	1	1	0	0	0

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

Bits 15:8 – VVER[7:0] Vendor Version Number
Reserved. Value subject to change. No functionality associated.

Bits 7:0 – SVER[7:0] Specification Version Number
This status indicates the SD Host Controller Specification Version.

Value	Name
0	SD Host Specification Version 1.00
1	SD Host Specification Version 2.00, including the feature of the ADMA and Test Register
2	SD Host Specification Version 3.00

41.8.35. SDHC Additional Present State Register

Name: APSR
Offset: 0x200
Reset: 0x0000000F
Property: -

Table 41-39. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

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

Bits 3:0 – HDATLL[3:0] High Line Level
This status is used to check the DAT[7:4] line level to recover from errors, and for debugging.

41.8.36. SDHC e.MMC Control 1 Register

Name: MC1R
Offset: 0x204
Reset: 0x00
Property: R/W

Note: This register is reserved to manage e.MMC specific features only.

Table 41-40. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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

Bit 7 – FCD e.MMC Force Card Detect

When using e.MMC, the user can set this bit to 1 to bypass the card detection procedure using the SDHC_CD signal.

Value	Name	Description
0	DISABLED	e.MMC Forced Card Detect is disabled. The SDHC_CD signal is used and debounce timing is applied.
1	ENABLED	e.MMC Forced Card Detect is enabled.

Bit 5 – BOOTA e.MMC Boot Acknowledge Enable

This bit must be set according to the value of BOOT_ACK in the Extended CSD Register (refer to “Embedded MultiMedia Card (e.MMC) Electrical Standard 4.51”).
When this bit is set to 1, the peripheral waits for boot acknowledge pattern from the e.MMC before receiving boot data.
If the boot acknowledge pattern is wrong, the BOOTAE status flag rises in SDHC_EISTR if BOOTAE is set in SDHC_EISTER. An interrupt is generated if BOOTAE is set in SDHC_EISIER.
If the no boot acknowledge pattern is received, the DATTEO status flag rises in SDHC_EISTR if DATTEO is set in SDHC_EISTER. An interrupt is generated if DATTEO is set in SDHC_EISIER.

Bits 1:0 – CMDTYP[1:0] e.MMC Command Type

Value	Name	Description
0	NORMAL	The command is not an e.MMC specific command.
1	WAITIRQ	This bit must be set to 1 when the e.MMC is in Interrupt mode (CMD40). Refer to “Interrupt Mode” in the “Embedded MultiMedia Card (e.MMC) Electrical Standard 4.51” .
2	STREAM	This bit must be set to 1 in the case of Stream Read(CMD11) or Stream Write (CMD20). Only effective for e.MMC up to revision 4.41.
3	BOOT	Starts a Boot Operation mode at the next write to SDHC_CR. Boot data are read directly from e.MMC device.

41.8.37. SDHC e.MMC Control 2 Register

Name: MC2R
Offset: 0x205
Reset: 0x00
Property: -

Note: This register is reserved to manage e.MMC specific features only.

Table 41-41. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
							ABOOT	SRESP
Access							W	W
Reset							0	0

Bit 1 – ABOOT e.MMC Abort Boot

This bit is used to exit from Boot mode. Writing this bit to 1 exits the Boot Operation mode. Writing 0 is ignored.

Bit 0 – SRESP e.MMC Abort Wait IRQ

This bit is used to exit from the Interrupt mode. When this bit is written to 1, the peripheral sends the CMD40 response automatically. This brings the e.MMC from Interrupt mode to the standard Data Transfer mode. Writing this bit to 0 is ignored.
Note: This bit is only effective when CMD_TYP in SDHC_MC1R is set to WAITIRQ.

41.8.38. SDHC AHB Control Register

Name: ACR
Offset: 0x208
Reset: 0x00000000
Property: -

Table 41-42. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			B1KBDIS	HNBRDIS			BMAX[1:0]	
Access			R/W	R/W			R/W	R/W
Reset			0	0			0	0

Bit 5 – B1KBDIS 1kB Boundary Disable
Only significant when the xKBBoundary is not supported by the HMATRIX. Used for debug.

Bit 4 – HNBRDIS HNBREQ Disable
Used for debug to modulate the peripheral host interface bandwidth. Set to 1 to reduce the peripheral bandwidth.

Bits 1:0 – BMAX[1:0] AHB Maximum Burst
This field selects the maximum burst size in case of DMA transfer.

Value	Name	Description
0	INCR16	The maximum burst size is INCR16.
1	INCR8	The maximum burst size is INCR8.
2	INCR4	The maximum burst size is INCR4.
3	SINGLE	Only SINGLE transfers are performed.

41.8.39. SDHC Clock Control 2 Register

Name: CC2R
Offset: 0x20C
Reset: 0x00000000
Property: -

Table 41-43. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

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

Bit 0 – FSDCLKD Force SDCLK Disabled

The user can choose to maintain the SDCLK during 8 SDCLK cycles after the end bit of the last data block in case of a read transaction, or after the end bit of the CRC status in case of a write transaction.

Value	Description
0	The SDCLK is forced and it cannot be stopped immediately after the transaction.
1	The SDCLK is not forced and it can be stopped immediately after the transaction.

41.8.40. SDHC Capabilities Control Register

Name: CACR
Offset: 0x230
Reset: 0x00000000
Property: -

Table 41-44. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
	KEY[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
								CAPWREN
Access								R/W
Reset								0

Bits 15:8 – KEY[7:0] Key

Value	Name	Description
46h	KEY	Writing any other value in this field aborts the write operation of the CAPWREN bit. Always reads as 0.

Bit 0 – CAPWREN Capabilities Write Enable
This bit can only be written if KEY correspond to 46h.

Value	Description
0	Capabilities registers (SDHC_CA0R and SDHC_CA1R) cannot be written.
1	Capabilities registers (SDHC_CA0R and SDHC_CA1R) can be written.

41.8.41. SDHC Debug Register

Name: DBGR
Offset: 0x234
Reset: 0x00
Property: -

Table 41-45. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

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

Bit 0 – NIDBG Non-Intrusive Debug

Value	Name	Description
0	DISABLED	Reading the SDHC_BDPR via debugger increments the dual port RAM read pointer.
1	ENABLED	Reading the SDHC_BDPR via debugger does not increment the dual port RAM read pointer.

42. True Random Number Generator (TRNG)

42.1. Overview

The True Random Number Generator (TRNG) module generates unpredictable random numbers that are not generated by an algorithm.

The TRNG is used to generate a random number for cryptographic applications. For applications that require an entropy source for seeding a NIST approved DRNG (Deterministic RNG) as required by FIPS PUB 140-2 and 140-3, it is recommended that an external TRNG source be used that is compliant with the NIST Special Publication 800-22 Test Suite.

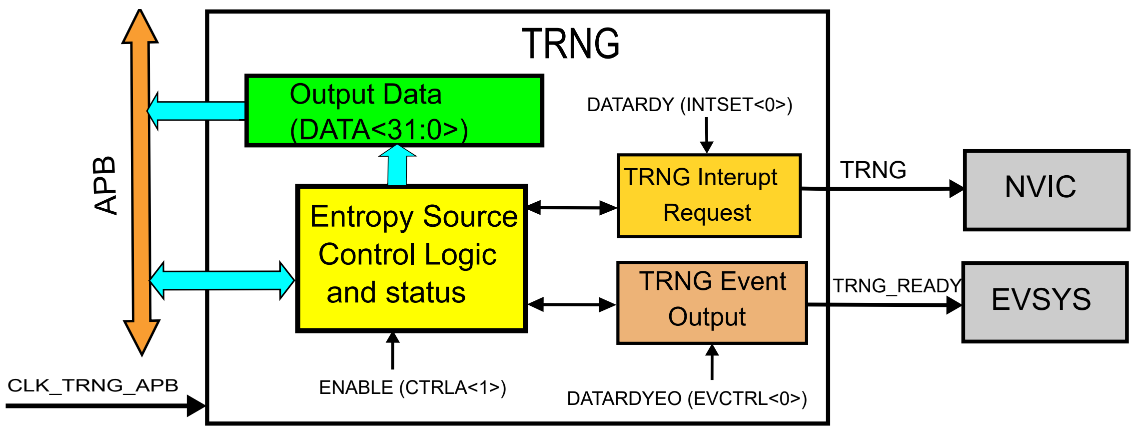
42.2. Features

The following is key feature of the module:

- Provides a 32-bit random number every 84 clock cycles

42.3. Block Diagram

Figure 42-1. TRNG Block Diagram.



42.4. Peripheral Dependencies

Peripheral Name	Base Address	NVIC IRQ Index: Source	MCLK AHBx/APBx Clock Enable Mask Bit	PAC Peripheral Identifier (PAC.WRCTRL.PERIDx)	EVSYS Generator (EVSYS.CHANNELn.EVGENx)	Power Domain
TRNG	0x4502_4000	147 : ISO	112 : CLK_TRNG_APB	54	125 : READY	VDDCORE_SW

42.5. Clocks

The TRNG bus clock (CLK_TRNG_APB) can be enabled and disabled in the Main Clock module, and the default state of CLK_TRNG_APB can be found in Peripheral Clock Masking.

References:

Peripheral Clock Masking.

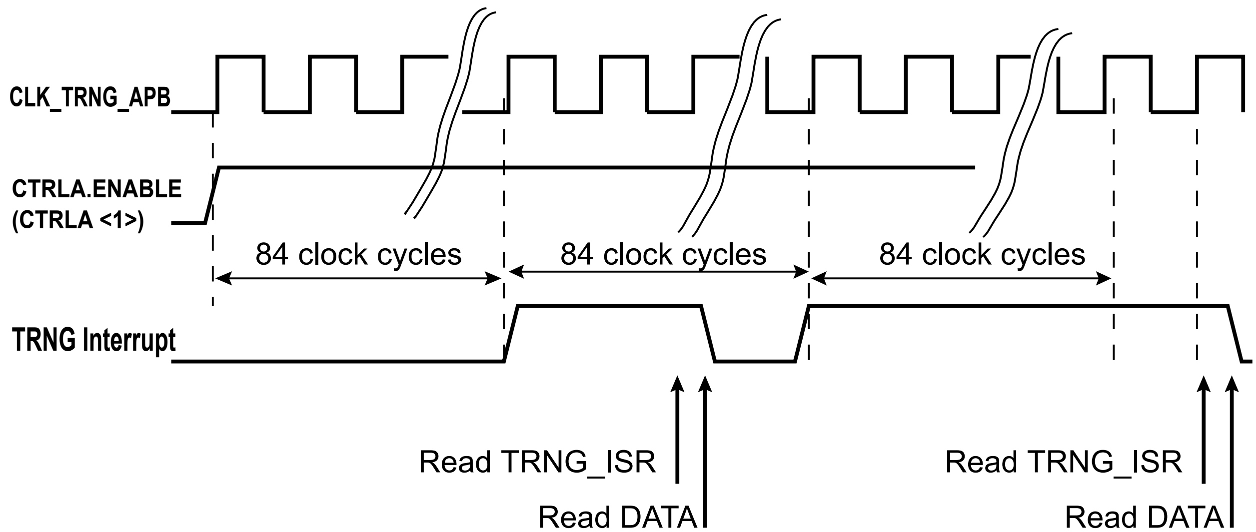
42.6. Functional Description

42.6.1. Principle of Operation

When the TRNG module is enabled, the peripheral starts providing new 32-bit random numbers every 84 CLK_TRNG_APB clock cycles.

The TRNG module can be configured to generate an interrupt or event when a new random number is available.

Figure 42-2. TRNG Data Generation Sequence



42.6.2. Basic Operation

42.6.2.1. Initialization

To operate the TRNG, do the following:

- 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
- Enable the TRNG operation by writing a ‘1’ to CTRLA.ENABLE bit (CTRLA.ENABLE <1>)
- When the INTFLAG.DATARDY bit (INTFLAG <0>) is set, read the Output Data register (DATA<31:0>) to get the newly generated random number

Note: INTFLAG.DATARDY bit (INTFLAG <0>) is cleared automatically when the DATA register is read.

42.6.2.2. Enabling and Disabling

The TRNG module is enabled by writing '1' to the Enable bit in the Control A register (CTRLA.ENABLE bit (CTRLA <1>)). The TRNG is disabled by writing a ‘0’ to CTRLA.ENABLE bit (CTRLA <1>).

42.6.3. Interrupts

The TRNG module has the TRNG Data Ready (TRNG_READY) interrupt source, which indicates that a new random number is available in the DATA register and ready to be read.

The interrupt source has an interrupt flag associated with it. The interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.DATARDY (INTFLAG <0>)) is set to ‘1’ when the interrupt condition occurs. The interrupt can be enabled by writing a '1' to the INTENSET.DATARDY bit (INTENSET <0>) in the Interrupt Enable Set register and disabled by writing a '1' to the INTENCLR.DATARDY bit (INTENCLR <0>) in the Interrupt Enable Clear (INTENCLR) register.

The interrupt request line is connected to the Nested Vector Interrupt Controller (NVIC). Using the TRNG interrupt requires the interrupt controller to be configured first. Refer to Nested Vector

Interrupt Controller for details. This interrupt is a synchronous wake-up source. See Sleep Mode Controller for details.

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 INTFLAG Status and Clear register for details on how to clear interrupt flags.

The following steps are used to configure the TRNG in interrupt mode:

- Configure the NVIC by setting group priority, sub priority, and by enabling the TRNG IRQ (Vector No 210)
- Enable TRNG interrupt by setting the INTENSET.DATARDY bit (INTENSET<0>)
- Generated 32-bit random number is read from the Output data register (DATA<31:0>) in the TRNG Interrupt Service Routine (TRNG_ISR)

References:

Sleep Mode Controller

Nested Vector Interrupt Controller

42.6.4. Events

The TRNG can generate the following output event:

- TRNG Data Ready (TRNG_READY): Generated when a new 32-bit random number is available in the Output Data register (DATA <31:0>).

Writing '1' to the TRNG Data Ready Event Output bit in the Event Control Register (EVCTRL.DATARDYEO bit (EVCTRL <0>)) enables the TRNG_READY event. Writing a '0' to this bit disables the corresponding output event. The TRNG cannot use any events from other peripherals, as it is not an Event User. Refer to *EVSYS – Event System* for details on configuring the Event System.

References:

EVSYS - Event System

42.6.5. Sleep Mode Operation

The Run in Standby bit in Control A register (CTRLA.RUNSTDBY bit (CTRLA <6>)) controls the behavior of the TRNG during standby sleep mode:

When this bit is '0', the TRNG is halted during sleep, but maintains its current configuration.

When this bit is '1', the TRNG continues to operate during sleep and the TRNG 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.

References:

Sleep Mode Controller

42.6.6. Debug Mode Operation

When the CPU is halted in debug mode the TRNG continues normal operation.

During debug, if the TRNG interrupt is enabled and the generated data is not read periodically in the Interrupt Service Routine (TRNG_ISR) data loss may result.

42.7. Register Summary

For descriptions and definitions of both Register and bitfield properties, refer to Register Properties.

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]							

42.7.1. Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00
Property: PAC Write-Protection

Table 42-1. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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 TRNG behaves during standby sleep mode:

Value	Description
0	The TRNG is halted during standby sleep mode.
1	The TRNG continues to operate in standby sleep mode.

Bit 1 – ENABLE Enable

Value	Description
0	The TRNG is disabled.
1	The TRNG is enabled.

42.7.2. Event Control

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

Table 42-2. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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

Bit 0 – DATARDYEO TRNG Data Ready Event (TRNG_RDY) Output
This bit indicates whether the TRNG Data Ready event output is enabled and whether an output event will be generated when a new random value is ready.

Value	Description
0	TRNG Data Ready event output is disabled and an event will not be generated.
1	TRNG Data Ready event output is enabled and an event will be generated.

42.7.3. Interrupt Enable Clear

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

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set (INTENSET) register.

Table 42-3. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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

Bit 0 – DATARDY Data Ready Interrupt (TRNG) Enable
Writing a '1' to this bit will clear the Data Ready Interrupt (TRNG) Enable bit, which disables the corresponding interrupt request.
Writing a '0' to this bit has no effect. Reading this bit provides the following information.

Value	Description
0	The TRNG interrupt is disabled.
1	The TRNG interrupt is enabled.

42.7.4. Interrupt Enable Set

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

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear (INTENCLR) register.

Table 42-4. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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

Bit 0 – DATARDY Data Ready Interrupt (TRNG) Enable
Writing a '1' to this bit will set the Data Ready Interrupt (TRNG) Enable bit, which enables the corresponding interrupt request.
Writing a '0' to this bit has no effect. Reading this bit provides the following information.

Value	Description
0	The TRNG interrupt is disabled.
1	The TRNG interrupt is enabled.

42.7.5. Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x0A
Reset: 0x00
Property: -

Note: Interrupt flags must be cleared and then read back to confirm the clear before exiting the ISR to avoid double interrupts.

Table 42-5. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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

Bit 0 – DATARDY Data Ready Interrupt Flag

This flag is set when a new random value is generated, and the TRNG interrupt will be generated if this interrupt is enabled (INTENSET.DATARDY bit (INTENSET <0>) =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.

42.7.6. Output Data

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

Table 42-6. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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] Output Data
These bits hold the 32-bit randomly generated output data.

43. Configurable Custom Logic (CCL)

43.1. 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 (LUT0/LUT1, LUT2/LUT3 etc.) outputs, enabling complex waveform generation.

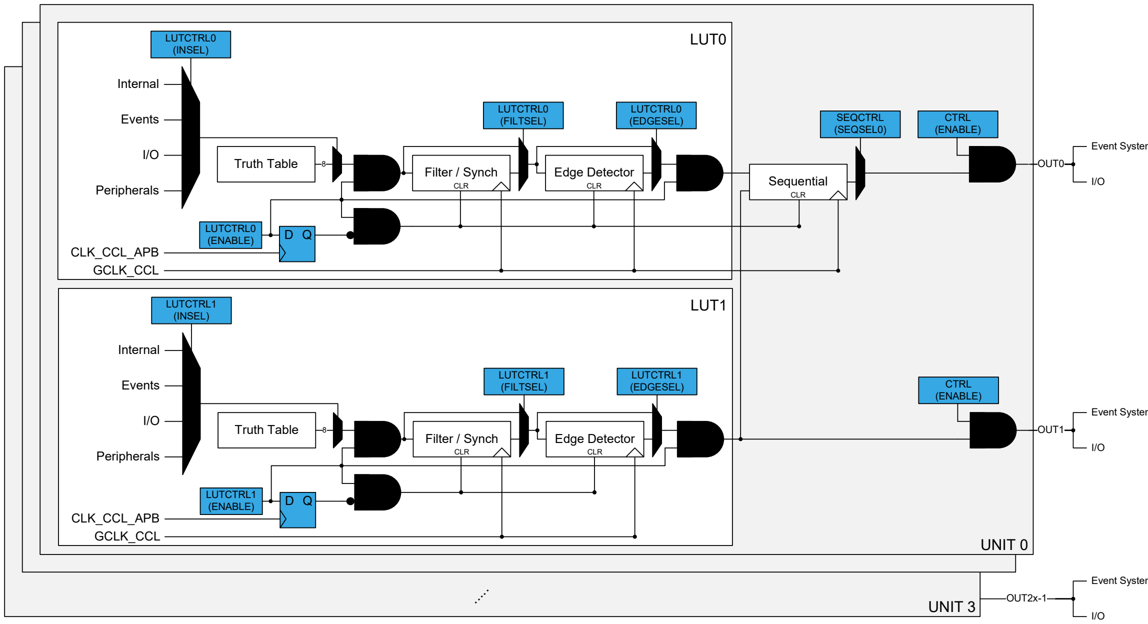
43.2. Features

The following are key features of the CCL module:

- 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

43.3. Block Diagram

Figure 43-1. Configurable Custom Logic



43.4. Signal Description

Pin Name	Type	Description
OUT[3:0]	Digital output	Output from lookup table
IN[11:0]	Digital input	Input to lookup table

For additional information on the pin mapping for this peripheral, refer to “I/O Multiplexing and Considerations”. One signal can be mapped on several pins.

43.5. Peripheral Dependencies

Base Address	MCLK AHBx/APBx Clock Enable Mask Bit	GCLK Peripheral Channel Clock Name:Register	PAC Peripheral Identifier (PAC.WRCTRL.PERIDx)	EVSYS Users (EVSYS.USERm)	EVSYS Generator (EVSYS.CHANNELn.EVGENx)	Power Domain
0x4501_A000	MCLK.CLKMSK3[13]	37 : GCLK_CCL	49	93-96 : LUTIN_x	114-117 : LUTOUT_x	VDDREG

43.5.1. I/O Lines

The CCL can take inputs and generate output through I/O pins. For this to function properly, the I/O pins must be configured to be used by a Look Up Table (LUT).

43.5.2. Power Management

This peripheral can continue to operate in any Sleep mode where its source clock is running. Events connected to the event system can trigger other operations in the system without exiting Sleep modes.

43.5.3. Clocks

The CCL bus clock (CLK_CCL_APB) can be enabled and disabled in the Main Clock module, MCLK (refer to the Main Clock (MCLK)), and the default state of CLK_CCL_APB can be found in Peripheral Clock Masking.

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 the Generic Clock Controller (GCLK) for additional information.

This generic clock is asynchronous to the user interface clock (CLK_CCL_APB).

43.5.4. Events

The CCL can use events from other peripherals and generate events that can be used by other peripherals. For this feature to function, the events have to be configured properly. Refer to the Chapter Event System (EVSYS) for additional information about the event users and event generators.

43.5.5. 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.

43.5.6. Register Access Protection

All registers with write access can be write-protected optionally by the Peripheral Access Controller (PAC). Refer to the Chapter Peripheral Access Controller (PAC) for additional information.

Optional write protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write Protection" property in each individual register description.

PAC write protection does not apply to accesses through an external debugger.

43.6. Functional Description

43.6.1. 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.

43.6.2. Operation

43.6.2.1. Initialization

The following bits are enable-protected, meaning that they can only be written when the corresponding even LUT is disabled (LUTCTRLx.ENABLE=0):

- Sequential Selection bits in the Sequential Control x (SEQCTRLx.SEQSEL) register

The following registers are enable-protected, meaning that they can only be written when the corresponding LUT is disabled (LUTCTRLx.ENABLE=0):

- LUT Control x (LUTCTRLx) register, except the ENABLE bit

Enable-protected bits in the LUTCTRLx registers can be written at the same time as LUTCTRLx.ENABLE is written to '1', but not at the same time as LUTCTRLx.ENABLE is written to '0'.

Enable-protection is denoted by the Enable-Protected property in the register description.

43.6.2.2. Enabling, Disabling, and Resetting

The CCL is enabled by writing a '1' to the Enable bit in the Control register (CTRL.ENABLE). The CCL is disabled by writing a '0' to CTRL.ENABLE.

Each LUT is enabled by writing a '1' to the Enable bit in the LUT Control x register (LUTCTRLx.ENABLE). Each LUT is disabled by writing a '0' to LUTCTRLx.ENABLE.

The CCL is reset by writing a '1' to the Software Reset bit in the Control register (CTRL.SWRST). All registers in the CCL will be reset to their initial state, and the CCL will be disabled. Refer to CTRL for details.

43.6.2.3. 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 the following figure. One or more inputs can be masked. The truth table for the expression is defined by TRUTH bits in LUT Control x register (LUTCTRLx.TRUTH).

Figure 43-2. Truth Table Output Value Selection

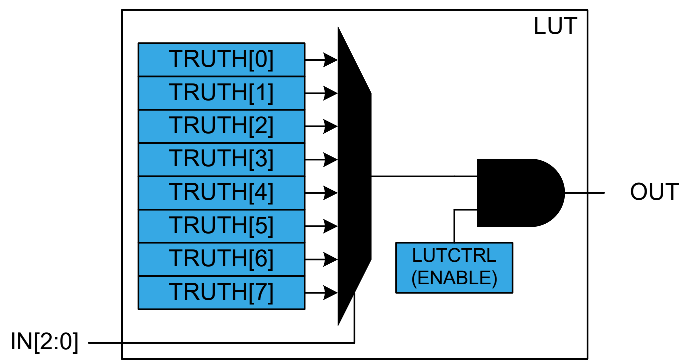


Table 43-1. Truth Table of LUT

IN[2]	IN[1]	IN[0]	OUT
0	0	0	TRUTH[0]
0	0	1	TRUTH[1]
0	1	0	TRUTH[2]
0	1	1	TRUTH[3]
1	0	0	TRUTH[4]
1	0	1	TRUTH[5]
1	1	0	TRUTH[6]
1	1	1	TRUTH[7]

43.6.2.4. Truth Table Inputs Selection

Input Overview

The inputs can be individually:

- Masked
- Driven by peripherals:
  - Analog comparator output (AC)
  - Timer/Counters waveform outputs (TC)
  - Serial Communication output transmit interface (SERCOM)
- Driven by internal events from Event System

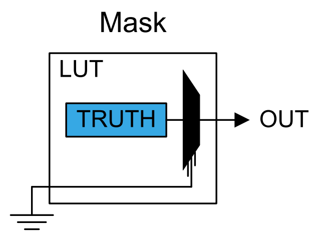
- Driven by other CCL sub-modules

The Input Selection for each input 'y' of LUT x is configured by writing the Input 'y' Source Selection bit in the LUT x Control register (LUTCTRLx.INSELy).

Masked Inputs (MASK)

When a LUT input is masked (LUTCTRLx.INSELy = MASK), the corresponding TRUTH input (IN) is internally tied to zero, as shown in this figure:

Figure 43-3. Masked Input Selection



Internal Feedback Inputs (FEEDBACK)

When selected (LUTCTRLx.INSELy = FEEDBACK), the Sequential (SEQ) output is used as input for the corresponding LUT.

The output from an internal sequential sub-module can be used as input source for the LUT, see figure below for an example for LUT0 and LUT1. The sequential selection for each LUT follows the formula:

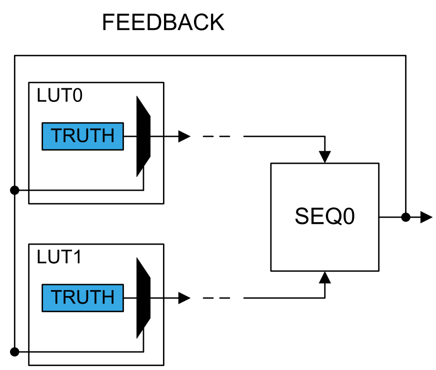
$IN[2N][i] = \mathrm{SEQ}[N]$

$IN[2N{+}1][i] = \mathrm{SEQ}[N]$

With *N* representing the sequencer number and *i*=0,1,2 representing the LUT input index.

For additional information, refer to Sequential Logic.

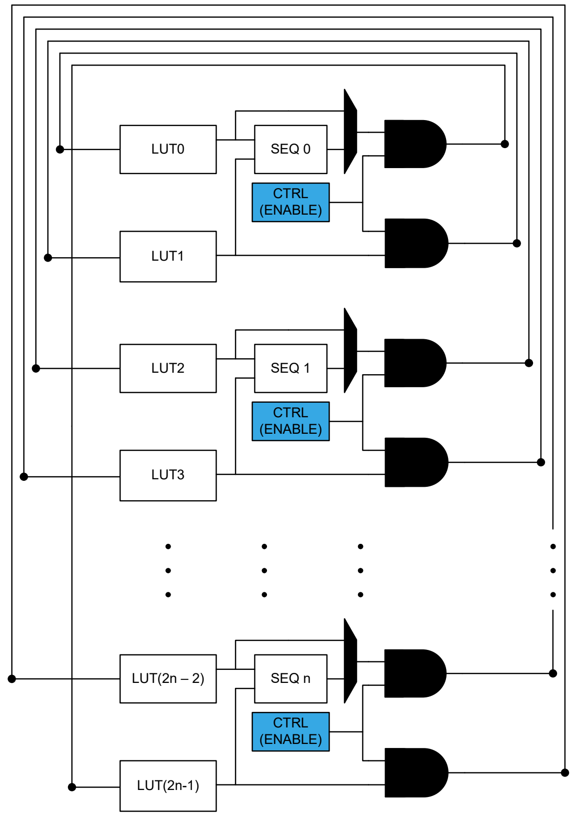
Figure 43-4. Feedback Input Selection



Linked LUT (LINK)

When selected (LUTCTRLx.INSELy=LINK), the subsequent LUT output is used as the LUT input (for example, LUT2 is the input for LUT1), as shown in the figure below:

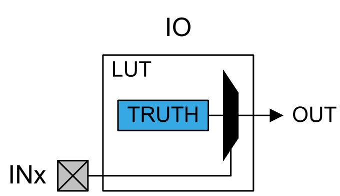
Figure 43-5. Linked LUT Input Selection



I/O Pin Inputs (IO)

When the I/O pin is selected as LUT input (LUTCTRLx.INSELy = IO), the corresponding LUT input will be connected to the pin, as shown in the figure below.

Figure 43-6. I/O Pin Input Selection



Analog Comparator Inputs (AC)

The AC outputs can be used as input source for the LUT (LUTCTRLx.INSELy=AC).

The analog comparator outputs are distributed following the formula:

`IN[N][i]=AC[N % ComparatorOutput_Number]`

With *N* representing the LUT number and *i*=[0,1,2] representing the LUT input index.

Before selecting the comparator output, the AC must be configured first.

The output of comparator 0 is available on even LUTs ("LUT(2x)": LUT0, LUT2) and the comparator 1 output is available on odd LUTs ("LUT(2x+1)": LUT1, LUT3), as shown in the figure below.

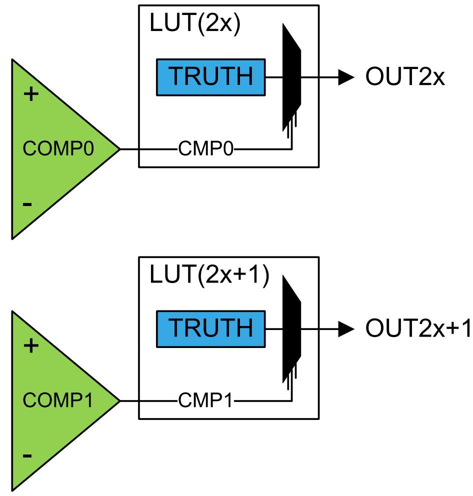


Figure 43-7. AC Input Selection

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.

Note:
The TCC selection for each LUT follows the formula:

$\mathrm{IN}[N][i] = TCC[N\,\%\,\mathrm{TCC\_Instance\_Number}].\mathrm{WO}[\mathrm{i}]$

Where *N* represents the LUT number and i represents the LUT input index (i=0,1,2).

Before selecting the waveform outputs, the TCC must be configured first.

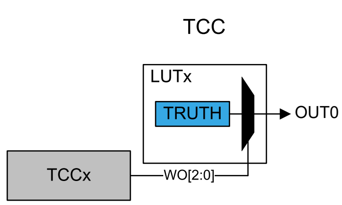


Figure 43-8. TCC Input Selection

Serial Communication Output Transmit Inputs (SERCOM)

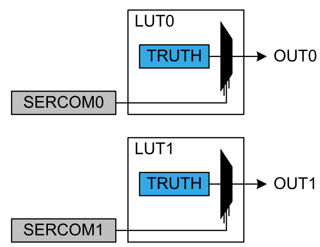
The serial engine transmitter output from Serial Communication Interface (SERCOM TX, TXd for USART, MOSI for SPI) can be used as input source for the LUT. The figure below shows an example for LUT0 and LUT1. The SERCOM selection for each LUT follows the formula:

$\mathrm{IN}[N][i] = SERCOM[N\,\%\,\mathrm{SERCOM\_Instance\_Number}]$

With *N* representing the LUT number and *i*=0,1,2 representing the LUT input index.

Before selecting the SERCOM as 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 43-9. SERCOM Input Selection

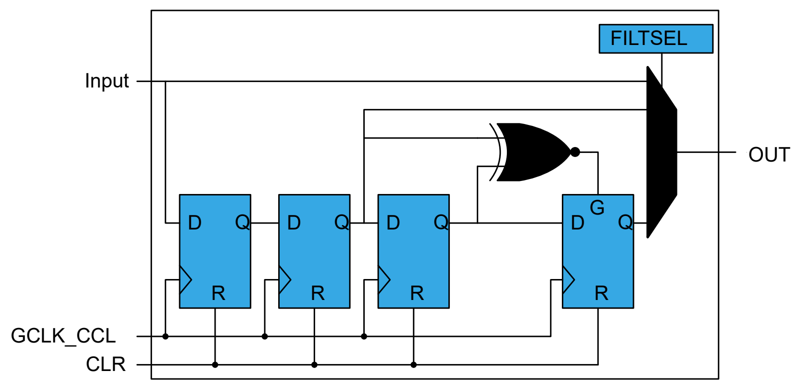


43.6.2.5. Filter

By default, the LUT output is a combinatorial function of the LUT inputs. This may cause some short glitches when the inputs change value. These glitches can be removed by clocking through filters, if demanded by application needs.

The Filter Selection bits in LUT Control register (LUTCTRLx.FILTSEL) define the synchronizer or digital filter options. When a filter is enabled, the OUT output will be delayed by two to five GCLK cycles. One APB clock after the corresponding LUT is disabled, all internal filter logic is cleared.
Note: Events used as LUT input will also be filtered, if the filter is enabled.

Figure 43-10. Filter



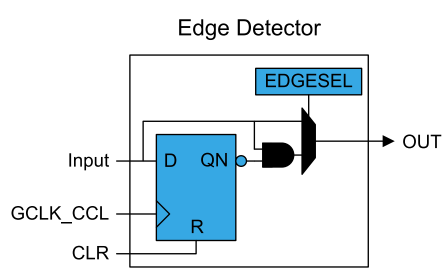
43.6.2.6. Edge Detector

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 (LUTCTRLx.EDGESEL). In order to avoid unpredictable behavior, either the filter or synchronizer must be enabled.

Edge detection is disabled by writing a '0' to LUTCTRLx.EDGESEL. After disabling a LUT, the corresponding internal Edge Detector logic is cleared one APB clock cycle later.

Figure 43-11. Edge Detector



43.6.2.7. Sequential Logic

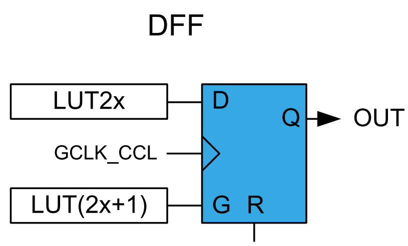
Each LUT pair can be connected to the internal sequential logic which can be configured to work as D flip flop, JK flip flop, gated D-latch or RS-latch by writing the Sequential Selection bits on the corresponding Sequential Control x register (SEQCTRLx.SEQSEL). Before using sequential logic, the GCLK_CCL clock and optionally each LUT filter or edge detector must be enabled.

Note: While configuring the sequential logic, the even LUT must be disabled. When configured the even LUT must be enabled.

Gated D Flip-Flop (DFF)

When the DFF is selected, the D-input is driven by the even LUT output (LUT0 and LUT2), and the G-input is driven by the odd LUT output (LUT1 and LUT3), as shown in the following figure.

Figure 43-12. D Flip Flop



When the even LUT is disabled (LUTCTRL0.ENABLE=0 / LUTCTRL2.ENABLE=0), the flip-flop is asynchronously cleared. The reset command (R) is kept enabled for one APB clock cycle. In all other cases, the flip-flop output (OUT) is refreshed on rising edge of the GCLK_CCL, as shown in the following table.

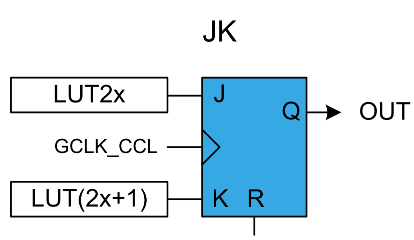
Table 43-2. DFF Characteristics

R	G	D	OUT
1	X	X	Clear
0	1	1	Set
		0	Clear
	0	X	Hold state (no change)

JK Flip-Flop (JK)

When this configuration is selected, the J-input is driven by the even LUT output (LUT0 and LUT2), and the K-input is driven by the odd LUT output (LUT1 and LUT3), as shown in the following figure.

Figure 43-13. JK Flip Flop



When the even LUT is disabled (LUTCTRL0.ENABLE=0 / LUTCTRL2.ENABLE=0), the flip-flop is asynchronously cleared. The reset command (R) is kept enabled for one APB clock cycle. In all other cases, the flip-flop output (OUT) is refreshed on rising edge of the GCLK_CCL, as shown in the following table.

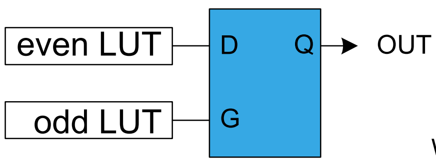
Table 43-3. JK Characteristics

R	J	K	OUT
1	X	X	Clear
0	0	0	Hold state (no change)
0	0	1	Clear
0	1	0	Set
0	1	1	Toggle

Gated D-Latch (DLATCH)

When the DLATCH is selected, the D-input is driven by the even LUT output (LUT0 and LUT2), and the G-input is driven by the odd LUT output (LUT1 and LUT3), as shown in the following figure.

Figure 43-14. D-Latch



When the even LUT is disabled (LUTCTRL0.ENABLE=0 / LUTCTRL2.ENABLE=0), the latch output will be cleared. The G-input is forced enabled for one more APB clock cycle, and the D-input to zero. In all other cases, the latch output (OUT) is refreshed as shown in the following table.

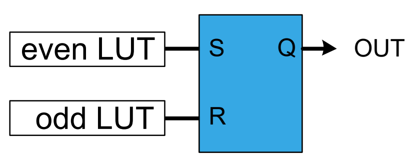
Table 43-4. D-Latch Characteristics

G	D	OUT
0	X	Hold state (no change)
1	0	Clear
1	1	Set

RS Latch (RS)

When this configuration is selected, the S-input is driven by the even LUT output (LUT0 and LUT2), and the R-input is driven by the odd LUT output (LUT1 and LUT3), as shown in the following figure.

Figure 43-15. RS-Latch



When the even LUT is disabled LUTCTRL0.ENABLE=0 / LUTCTRL2.ENABLE=0), the latch output will be cleared. The R-input is forced enabled for one more APB clock cycle and S-input to zero. In all other cases, the latch output (OUT) is refreshed as shown in the following table.

Table 43-5. RS-Latch Characteristics

S	R	OUT
0	0	Hold state (no change)
0	1	Clear
1	0	Set
1	1	Forbidden state

43.6.3. Events

The CCL can generate the following output events:

- LUTOUTn with n=0-3: Lookup Table Output Value

Writing a '1' to the LUT Control Event Output Enable bit (LUTCTRLx.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:

- INSELy: The event is used as input for the TRUTH table. For additional information, refer to Events.

Writing a '1' to the LUT Control Event Input Enable bit (LUTCTRLx.LUTEI) enables the corresponding action on input event. Writing a '0' to this bit disables the corresponding action on input event.

43.6.4. 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.

43.7. Register Summary

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	

43.7.1. Control

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

Note: CTRL register (except the bits ENABLE & SWRST) is Enable Protected when CCL.CTRL.ENABLE = 1.

Table 43-6. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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 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.

43.7.2. Sequential Control x

Name: SEQCTRL
Offset: 0x04 + n*0x01 [n=0..1]
Reset: 0x00
Property: PAC Write-Protection, Enable-Protected

Note: SEQCTRLx register is Enable-protected when LUTCTRLx.ENABLE = 1.

Table 43-7. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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

43.7.3. LUT Control n

Name: LUTCTRLn
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.

Table 43-8. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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 – INSELy y = [0,1,2]. LUT Input y Source Selection

These bits select the LUT input y source:

Note: TCC6 and TCC7 connections are unavailable for devices that do not have those TCC instances.

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
0x0B - 0x0F	Reserved	Reserved

Table 43-9. CCL Internal Connection

	0x5	0x6	0x7	0x8	0x9	0xA
LUT0.IN0	AC0 output	SERCOM0 padout[0]	TCC0 WO0	TCC0 WO4	TCC4 WO0	TCC5 WO0
LUT0.IN1	AC0 output	SERCOM0 padout[0]	TCC0 WO1	TCC0 WO5	TCC4 WO1	TCC5 WO1
LUT0.IN2	AC0 output	SERCOM0 padout[0]	TCC0 WO2	TCC0 WO6	TCC4 WO0	TCC5 WO0
LUT1.IN0	AC1 output	SERCOM1 padout[0]	TCC1 WO0	TCC1 WO4	TCC6 WO0	TCC7 WO0
LUT1.IN1	AC1 output	SERCOM1 padout[0]	TCC1 WO1	TCC1 WO5	TCC6 WO1	TCC7 WO1
LUT1.IN2	AC1 output	SERCOM1 padout[0]	TCC1 WO2	TCC1 WO6	TCC6 WO0	TCC7 WO0
LUT2.IN0	AC0 output	SERCOM2 padout[0]	TCC2 WO0	TCC2 WO4	TCC0 WO0	TCC1 WO0
LUT2.IN1	AC0 output	SERCOM2 padout[0]	TCC2 WO1	TCC2 WO5	TCC0 WO1	TCC1 WO1
LUT2.IN2	AC0 output	SERCOM2 padout[0]	TCC2 WO2	TCC2 WO6	TCC0 WO0	TCC1 WO0
LUT3.IN0	AC1 output	SERCOM3 padout[0]	TCC3 WO0	TCC3 WO4	TCC2 WO0	TCC3 WO0
LUT3.IN1	AC1 output	SERCOM3 padout[0]	TCC3 WO1	TCC3 WO5	TCC2 WO1	TCC3 WO1
LUT3.IN2	AC1 output	SERCOM3 padout[0]	TCC3 WO2	TCC3 WO6	TCC2 WO0	TCC3 WO0

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

Value	Name	Description
0x3	-	Reserved

Bit 1 – ENABLE LUT Enable

Value	Description
0	The LUT is disabled.
1	The LUT is enabled.

44. Analog-to-Digital Converter (ADC)

44.1. Overview

The Analog-to-Digital Convertor can convert one independent signal using the SAR (Successive Approximation Register) ADC Module. The ADC module supports up to 12 bit resolution. For sample timing and conversion rates, see the Analog-to-Digital Converter (ADC) Electrical Specifications. Single-ended and differential measurements are supported. Signed and unsigned sampling formats are supported.

The SAR ADC module can support up to 14 separate analog input channels.

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 AVDD and external VREFH reference voltages can be used (i.e., AVDD or VREFH respectively).

The ADC has a digital result compare function for accurate monitoring of user-defined thresholds, with minimum software intervention required. Hardware oversampling is also supported in the ADC's hardware.

Note: If AVDD < 2.5V, the user must enable analog charge pumps in SUPC.VREGCTRL.CPEN[1:0].

44.2. Features

The following are key features of the ADC module:

Up to 12-bit resolution of the numerical output, signed or unsigned (Higher resolutions possible with oversampling).

- Signed/Unsigned results
- Left or Right aligned result
- Fractional or integer results
- Throughput rates (refer to the *Electrical Specifications)*:
  Note:
  Assuming, GCLK_ADC = 50 MHz, TAD = 1/50 MHz = 20 ns.
  12-bit resolution: Up to 3.125 Msps
  10-bit resolution: Up to 3.571429 Msps
  8-bit resolution: Up to 4.166667 Msps
- Maximum of 12 unique external analog input channels
  – VINP[11:0] ADC Module 0
- Internal inputs:
  – Temperature Sensor internal analog channel AIN12
  – IVREF 1.2v, internal analog channel AIN13

  Note: The minimum sample rate for ADC is Ftpr >= 100 ksps.
- ADC 12 single ended external analog inputs or up to three differential inputs plus 2 internal channels
- Up to 16 trigger sources, off-chip hardware or on-chip hardware or software generated per analog input channel
- Edge or level active triggering modes, generating single conversions or bursts of conversions
- A scan trigger to start a scan cycle which can individually include or not include any of the analog inputs

- Any of the 16 trigger sources or the scan triggers can be assigned to individual analog input channels
- The scan trigger itself can select any of the 16 trigger sources as its own source
- Programmable sampling time, CORCTRL.SAMC
- Each analog input/channel output register can be read from a general dedicated output register (Write to the CORDYID and CHRDYID registers, then reads the CHRDYDATA register)
- 16 sample deep FIFO supporting all channels
- Digital Comparator for monitoring output values in relation to user-specified ADC result threshold
- Digital filter, providing averaging/oversampling for increased noise immunity and are assignable to any analog input

44.3. Block Diagram

The ADC Module has a dedicated Analog Input Multiplexer, which can select from seven up to 16 different analog inputs . The ADC Module has a Post Processing Block, consisting of a dedicated Digital Filter and a dedicated Digital Comparator.

The ADC's sample data is available in the ADC Channel Ready Data Register (CHRDYDAT) and a 16-sample-deep FIFO (PFFDATA).

The ADC module provides two interrupts to the NVIC: one Global Interrupt and one ADC Module-specific interrupts. Global interrupts are serviced by the CTLINTENCLR, CTLINTENSET, and CTLINTFLAG registers, while the ADC module's interrupts are serviced by INTENCLR, INTENSET, and INTFLAG registers .

Figure 44-1. ADC Block Diagram

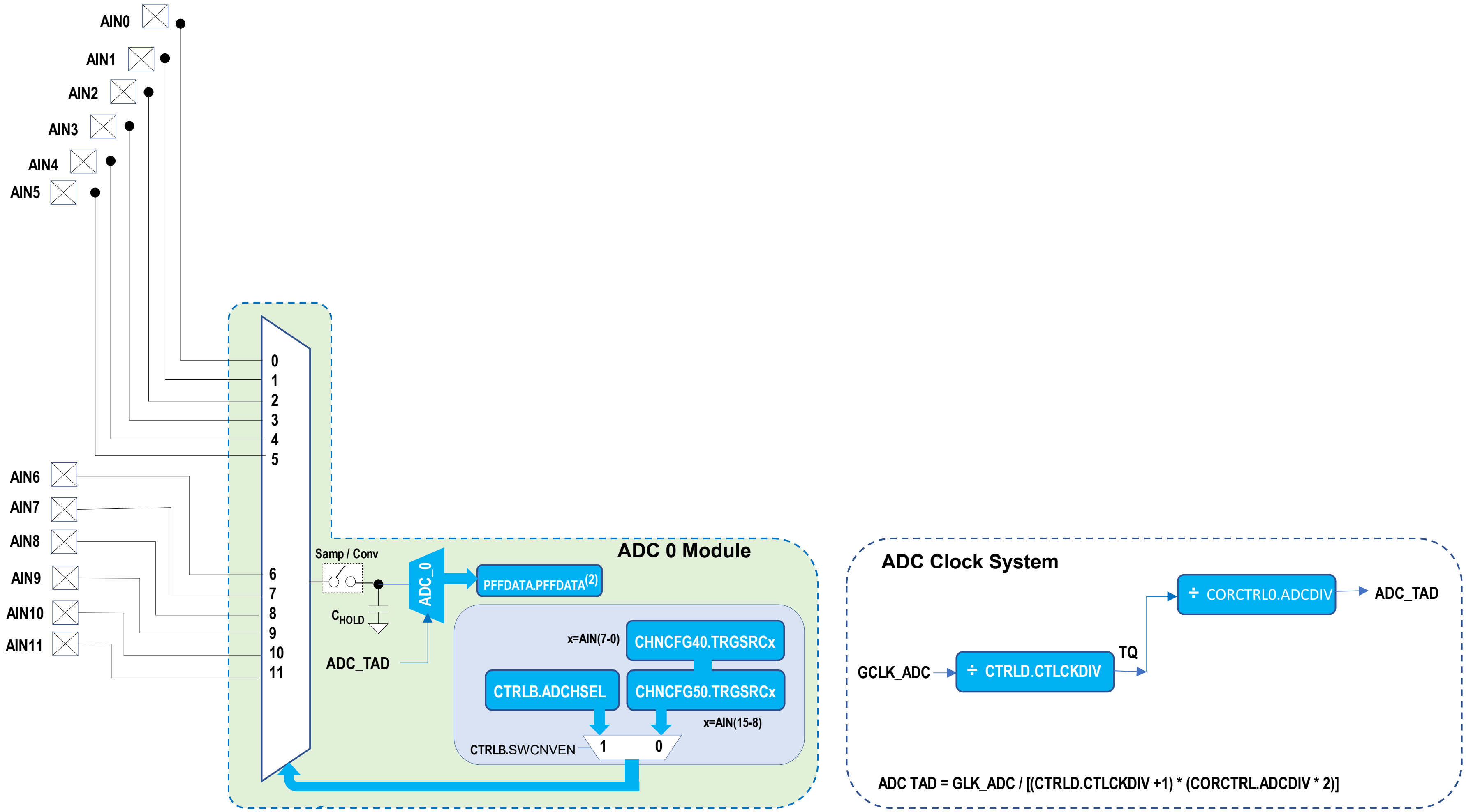
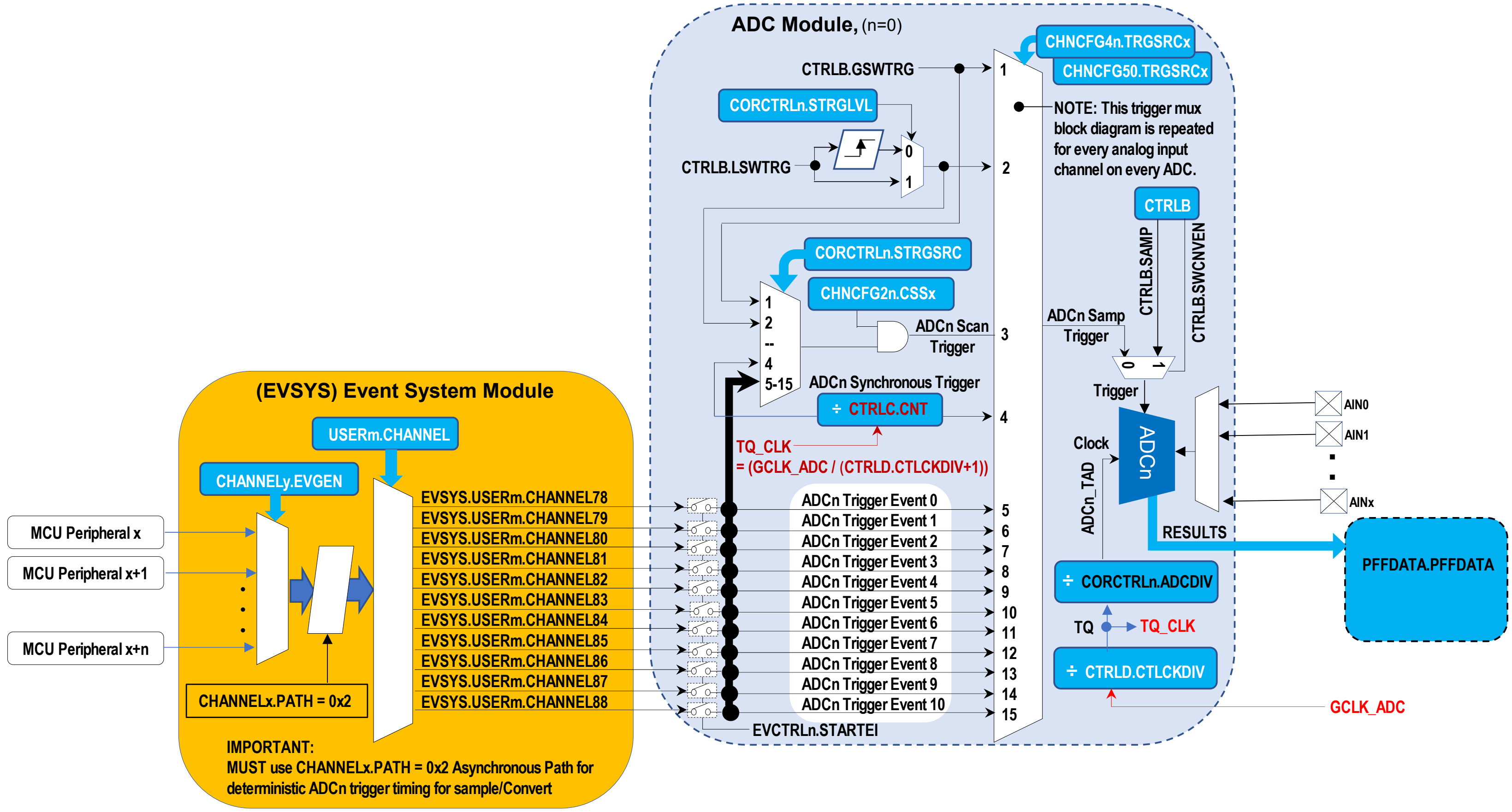


Figure 44-2. ADC Trigger Block Diagram



The ADC runs on a peripheral clock provided by the Generic Clock (GCLK) module, known as the GCLK_ADC, and identified as ADC Control Clock, or CTL_CLK in this chapter. The ADC Module has a clock derived from the ADC Control Clock, CORE_CLK. ADC Special Function Registers (SFRs) are identified as to which clock domain they belong: “APB_CLK” for the APB/Main Clock and “GCLK” for the clock derived from the GCLK, i.e. the ADC Control Clock (CTL_CLK).

The ADC Data Bus is 19 bits wide: { ChannelValid, CoreChannelID[5:0], ChannelData{11:0] }. The output results of each of the four ADC modules are stacked onto the bus in a 4:1 Time Division Multiplexing (TDM) scheme and the ChannelValid signal identifies whether the core’s time slot contains valid data. The input channel index is reported in the CoreChannelID bits. The captured signal is reported in the 12 bits of ChannelData.

The ADC Data Bus sends data to the ADC’s status and data registers as well as to the Digital Filter and Digital Comparator Post Processing Blocks. The bus also supports sending filtered results to the ADC’s data registers.

44.4. Signal Description

Table 44-1. Signal Description

Signal	Type	Description
AVDD	Power	Analog Power Supply
AVSS	Power	Analog Ground Supply
ADC_VREFP0	Reference	External voltage reference
ADC VINPx, VINMy, x=0-11, y=0, 2, 4, x =0,1,...,11	Analog	VINPx/VINMy Analog inputs for ADC Module 0

Notes:
1. Channel 12 is attached internally to the Temperature Sensor Monitor.
2. Channel 13 is attached internally to the IVREF, Band Gap 1.2 Volt Monitor.

Note: Please consult the section *Signal Description* and *ADC Pinout I/O Description* table to determine which pins provide which input signals, depending on the device package in use.

All the VINPx signals listed above can be used for single-ended input. Only 3 specific VINPx inputs can be combined with the VINMx inputs for differential measurements.

- (ADC_VINP0 (Diff +) , ADC_VINM0 (Diff -))
- (ADC_VINP2 (Diff +) , ADC_VINM2 (Diff -))
- (ADC_VINP4 (Diff +) , ADC_VINM4 (Diff -))

These pairs are configured in the CHNCFG3n (n=1,2,3) register by setting the DIFF bits corresponding to the signal pairs in use.

44.5. Peripheral Dependencies

Peripheral Name	Base Address	NVIC IRQ Index: Source	MCLK AHBx/APBx Clock Enable Mask Bit	GCLK Peripheral Channel Clock Name:Register	PAC Peripheral Identifier (PAC.WRCTRL.PERIDx)	EVSYS Users (EVSYS.USERm)	EVSYS Generator (EVSYS.CHANNELn.EVGENx)	DMA Trigger Index:Source (DMAC.CHCTRLBk.TRIGx)	Power Domain
ADC	0x4501_0000	132: REQ0 133: REQ1	MCLK.CLKMSK3[8]	GCLK_ADC: GCLK.PCHCTRL[33]	44	78-88 : TRIG_x	106 : CHRDYC 107 : CMP	61 : PFFRDY	VDDREG, AVDD

44.6. Functional Description

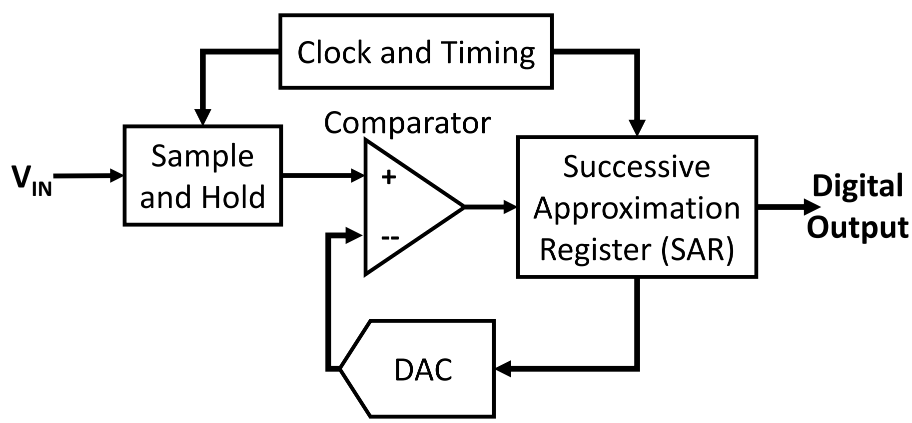
44.6.1. Principle of Operation

Important: When AVDD < 2.5v, for proper ADC operation it is critical that the internal charge pumps be enabled as necessary as define in SUPC.VREGCTRL.CPEN[1:0].

The basic architecture of an ADC Module is shown in the following figure.

At the start of a capture cycle the Sample and Hold capacitor is connected to the incoming voltage on a trigger event until the expiration of the sample time defined by CORCTRLx.SAMC. At the end of the sampling period the Sample and Hold is disconnected from the input signal and connected to the Comparator and the conversion sequence begins. ADC conversion time = (# bits +1)*TAD).

Figure 44-3. ADC Module Architecture



44.6.2. Basic Operation

44.6.2.1. Enabling, Disabling, and Resetting

The ADC is enabled by setting **CTRLA.ENABLE** = 1. Setting this bit to zero disables the ADC. The ADC can be reset by setting **CTRLA.SWRST** = 1 to initiate a software reset. (The ADC module is reset when the **SYNCYBUSY.SWRST** bit goes low.)

Note: The bit **DBGCTRL.DBGRUN** is unchanged by a software reset.

44.6.2.2. Operation

A trigger event initiates the start of the ADC sampling of the selected analog input. After the expiration of the SAMC sample time the conversion sequence is automatically initiated by the hardware assuming CTRLB.SWCNVEN = 0. For maximum efficiency, in the case of SCAN mode, at the conclusion of sampling for the current active analog input and while the conversion cycle is in progress, the analog channel input multiplexer will switch to the next analog input in the users scan sequence in preparation to initiate the next channel sampling sequence immediately once the previous channel conversion is complete. In SCAN mode, sample/conversion progression is always lower to higher numbered channels as selected by the user in the CHNCFG2n.CSSx registers for the respective ADCn module.

44.6.2.3. Clocks Setup

The interface to the Advanced Peripheral Bus (APB) is clocked by the system's Main Clock. The main clock (GCLK0 or MCLK) can operate up to 120 MHz.

The ADC Core operates on a clock provided by the GCLK module, GCLK_ADC max 100Mhz. The register field GCLK.PCHCTRL33.GEN[2:0] is configured to select one of the GCLK's clock generators. Clock generators are configured by the register GCLK.GENCTRLm , for m=0,1,...,11. This register selects the input clock for the generator and provides the clock divider to divide an input clock down to the clock output of GCLK Generator m . This clock is called the ADC Control Clock, or CTL_CLK in this chapter.

44.6.2.4. Conversion Timing and Sampling Rate

If the period of CTL_CLK is T_Q , then the period of the respective ADC_CLK module clock, T_{AD} , is given by:

$T_{AD} = 2 \cdot ADCDIV \cdot T_Q$.

The minimum ADC clock period is $T_{AD} = 2 \cdot T_Q$.

The maximum ADC throughput rate with N_{BITS} of resolution, is:

Provided CTRLB.SWCNVEN=0:

Maximum Throughput rate F_{TPR} = [1/ [(CORCTRL.SAMC+2)TAD + (N_{BITS}+1)TAD]] / #Active ADC Channels

Example:

(CORCTRL.SAMC=0x1, 12bit resolution and 2 AINx scan channels selected)

F_{TPR} = [1/ (3TAD + 13TAD)] / 2

= (1 / 16TAD) / 2

Important: The TAD clock must have the following characteristics:

1. TAD <= 75 MHz
2. The highest value possible given the integer divisor relationship with the clock source
   Examples are as follows:
   - Clock Source TAD CLOCK(min.):
     - 300 MHz to 75 MHz
     - 200 MHz to 50 MHz
     - 150 MHz to 75 MHz
     - 120 MHz to 60 MHz
     - 72 MHz to 72 MHz

44.6.2.5. Voltage Reference Selection

The ADC's voltage reference is selected by CTRLD.VREFSEL[2:0]:

VREFSEL[2:0]	ADREF+	ADREF-
000	A_{VDD}	A_{VSS}
001	External $V_{REF}H$	A_{VSS}
010 - 111	Reserved	Reserved

44.6.2.6. ADC Resolution

The output resolution is selected by CORCTRL.SELRES, to choose between 6 bits, 8 bits, 10 bits, or 12 bits (default). Changing the resolution of the ADC module will also change the ADC throughput rate.

Note: Required analog signal sample time in ADC clock cycles (TAD) as defined in CORCTRL.SAMC depend on source impedance of external signal source. Consult electrical specifications for required sample time based on external source impedance.

44.6.2.7. Differential and Single Ended Conversions

The ADC supports a maximum of three differential signal pairs.

44.6.2.8. Input Channel Conversion Priority

The selection of the next analog input for conversion uses a natural ascending analog input channel order priority. In scan mode, analog input channels are always processed lower to higher channels as selected by the user in the ADC CHNCFG2n.CSS register.

44.6.2.9. Output Data Formats

The Analog/Digital core provides up to 12-bits of resolution which are available in registers CHRDYDAT and PFFDATA as 16 bits. The 12-bit ADC output data is translated into a 16-bit format specified by CHNCFG2n.FRACTk. Additionally, the CHNCFG2n.SIGNk bit specifies the data format as unsigned, or two's complementary signed.

The following table shows the four cases of the channel index k 16-bit output data format depending on the values [FRACTk ,SIGNk]:

Table 44-2. ADC Analog Input Channel n Data Format Examples

FRACT*k*	SIGN*k*	16-bit Output Data Format
		s = sign bit d = mantissa value bit
0	0	0000 dddd dddd dddd
0	1	ssss sddd dddd dddd
1	0	dddd dddd dddd 0000
1	1	sddd dddd dddd 0000

44.6.2.10. Initialization

The following steps are required to configure the ADC for operation:

1. Configure input pins for operation as analog input pins, ADC_AINk.
2. Configure a GCLK generator to provide the GCLK_ADC clock (PCHCTRLm index = 33).
3. Configure ADC registers:
   a. Copy factory-provided calibration values from the CAL OTP Register FCCFG65 into CALCTRL for all the ADC in use by the application.
   b. Configure CTRLA as needed but do not set CTRLA.ENABLE.
   c. Configure CTRLD, choosing the clock divider (CTLCKDIV) for the CTL_CLK clock based on the GCLK_ADC, Voltage Reference (VREFSEL), and Wake-up Delay Exponent (WKUPEXP).
   d. (Optionally) Enable ADC interrupts for FIFO support using CTLINTENSET. Optionally enable the interrupts for VREFRDY and Core n ready. (Or these conditions can be monitored by watching CTLINTFLAG in a while(1) loop.) Enable interrupt vector 132 (ADC Global Interrupt) in the NVIC.
   e. Enable ADC interrupts using the INTENSET registers. Enable the corresponding NVIC interrupt vector 133.
   f. For each ADCn, configure its Control register CORCTRLn:
      - Set the clock divider, ADCDIV, for the ADC Module clock, CORE_CLK[n].
      - Set the sample count SAMC (ADCDIV and SAMC determine the ADC Module's sample rate fs).
      - If ADC resolution other than 12 bits is desired, set it using SELRES.

- Other settings in CORCTRLn as needed.

g. Setup differential channels using CHNCFG3n.DIFFk = 1 for signals pairs k and k+1 (k even).

h. Select the input channels for each scan using CHNCFG2n.CSSk = 1.

i. Set the data format for each sample using CHNCFG2n.FRACTk and CHNCFG3n.SIGNk.

j. Setup channel triggers.
   - (Optional: Setup up the Digital Filter and Digital Comparator associated with each of the four modules).

4. Enable the ADC by setting CTRLA.ENABLE = 1.
5. Wait until CTLINTFLAG.VREFRDY = 1, either using an ISR or polling in a while(1) loop. This signals that the chip's analog environment is ready.
6. Wait for CRRDYn to go high in CTLINTFLAG, indicating that the corresponding ADC Module is ready. This can be polled in a while(1) loop or trapped by the ISR attached to the ADC Global Interrupt.

44.6.2.11. Channel Triggers

The ADC controller starts the capture and conversion of an input channel when that channel is "triggered". The ADC Controller needs to know when a channel requests an analog-to-digital conversion. For this purpose, each input channel contains a trigger signal which is passed by the channel to the Module's Access Arbiter. The Access Arbiter drives the address selections to both the input Analog Multiplexer and to the output Digital De-multiplexer.

44.6.2.11.1. Trigger Priority

When two triggers arrive at the Access Arbiter at the same time the trigger associated with the input channel having the lower index *k* will be serviced. The second trigger will be queued for later service, provided it is not overwritten by a third trigger request. In that case, the third trigger will be lost and not serviced.

44.6.2.11.2. Channel Trigger Selection

The trigger used for the ADC, Input Channel k, is specified by CHNCFG{4|5}n.TRGSRCk[3:0]. (CHNCFG4n contains the trigger choices for k = 0,1,...,7. CHNCFG5n contains the trigger choices for k = 8,9,...,15. TRGSRCk for Sn ≤ k are not defined.)

The possible values for TRGSRCk are:

= 0000: No Trigger (NOP)

= 0001: Global Software Trigger (CTRLB.GSWTRG)

= 0010: Global Level Software Trigger (CTRLB.LSWTRG)

= 0011: SCANTRG - Scan Trigger

= 0100: STRIG Synchronous Trigger

= 0101 - 1111: ADC Trigger Event User 0 – 10

Input channels with no triggers will not be serviced during the Module's operation. For all trigger sources except the SCANTRG, setting the TRGSRCk will result in only input channel k being serviced by the Module. To collect a "scan" of input channels TRGSRCk must be set to SCANTRG and then SCANTRG is defined by CORCTRLn.STRGSRC. All the input channels to be included in a scan started by SCANTRG must have CHNCFG2n.CSSk set to one.

The SCANTRG source is selected by CORCTRLn.STRGSRC:

= 0000: No trigger (NOP)

= 0001: Global Software trigger (CTRLB.GSWTRG)

= 0010: Level Software trigger (CTRLB.LSWTRG)

= 0011: Reserved

= 0100: Synchronous Trigger (STRIG)

= 0101 - 1111: EVSYS Event Generator 0 – 10

The Synchronous Trigger (STRIG) is driven by a counter at the ADC Control Clock (CTL_CLK) frequency and fires when the counter reaches the value defined by CTRLC.CNT[7:0]. To enable this clock set CTRLB.STRGEN to one.

44.6.2.11.3. Software Triggers

ADC captures can be directly controlled from software by using the Global Software Trigger to start a single capture (when CTRLB.GSWTRG = 1) or by using the Global Level Software Trigger to start a burst of captures that will continue as long CTRLB.LSWTRG = 1 and stop when LSWTRG = 0.

44.6.2.11.4. ADC Debugging

Setting CTRLB.SWCNVEN = 1 allows two bits in CTRLB to control the Module and the input channel specified by CTRLB.ADCHSEL. For this to work all the other input channels for the Module must be disabled by setting TRGSRCk = 0, which can only be accomplished when the ADC is disabled. Sampling of the specified input channel starts when CTRLB.SAMP is set to one and stops when CTRLB.SAMP is reset to zero. Immediately after this start conversion by setting CTRLB.RQCNVRT to one.

The CTRLB register has a dedicated SYNCBUSY bit to allow the manipulation of these bits when the ADC is enabled. After writing to the CTRLB register the SYNCBUSY.CTRLB will go high. Wait until the bit goes low before writing to CTRLB again.

44.6.2.11.5. Trigger Limitations

<u>Trigger Rule 1</u>: In order to ensure synchronizing every single pulse on GSWTRG, the GSWTRG pulses must be at least 4 ADC Control Clock periods apart, positive edge to positive edge (because the GSWTRG lasts only 1 single APB clock period by construction).

<u>Trigger Rule 2</u>: If a channel k is effectively included in scan n by setting CHNCFG{4|5}n.TRGSRCk = 3 (SCANTRG) and CHNCFG2n.CSSk= 1, then the user MUST ensure that NO other triggers are generated for that channel using CTRLB.RQCNVRT or any digital filter. Otherwise the scan behavior is unpredictable.

<u>Trigger Rule 3:</u> In order to ensure synchronizing every single pulse on LSWTRG, the LSWTRG pulses must be at least 8 ADC Control Clock periods + 10 Main Clock clock periods positive edge to positive edge, and also at least 4 ADC control clock periods + 4 Main Clock periods negative edge to positive edge.

44.6.3. ADC Result Registers

Each of the 14 analog input channels has an associated results register. To read the converted data write the ADC Module index n and the channel index k in the CORCHDATAID register:

CORCHDATAID.CORDYID = n

CORCHDATAID.CHRDYID = k

Then read the data from the CHRDYDAT register. The captured data bits are stored in CHRDYDAT, together with the FRACT, SIGN, DIFF, and LVL settings associated with this input channel. CHRDYDAT is 16 bits wide. For more information on the data formats found in this field consult Output Data Formats.

When data from ADCn, input channel k, is ready to be read by the application the INTFLAGn.CHRDY[k] bit will be set. If the corresponding bit in INTENSETn has been set, then the interrupt associated with ADCn will fire. (To reset the CHRDY bit write one to it.) If the application does not read the new data before another sample arrives the Channel Overwritten Error flag (INTFLAGn.CHNERRC) bit will be set.

44.6.4. ADC Result FIFO

The ADC has a 16-sample deep FIFO. To capture ADCn results in this FIFO, first enable the FIFO: PFFCTRL.PFFEN = 1. Then enable the FIFO for ADCn: PFFCTRL.PFFCRn= 1. FIFO results are available in the PFFDATA register.

The status of the FIFO is reflected in the CTLINTFLAG register. Setting the corresponding bits in the CTLINTENSET register enables these bits to fire the ADC Global Interrupt.

44.6.5. Additional Features

44.6.5.1. Dedicated Digital Filter

The Dedicated Digital Filter enables the ADC Module to feed a contiguous (i.e. back-to-back) set of input samples into a digital filter to produce a single finished output sample that either “averages” or “oversamples” the input samples. All the captured samples are equally weighted in the accumulation of the result before scaling of the final result.

The main difference between the “averaging” and “oversampling” modes relates to how the final filter output is scaled. Collecting 16 samples in “oversampling” mode generates two additional bits of output (14 bits total) while in “averaging” mode the output is always exactly 12 bits wide for all burst sizes.

The Digital Filter trades ADC throughput for increased data output resolution, in the sense that if random noise contaminates the least significant bits of a sample then “oversampling” can increase the resolution of the results. Oversampling by a factor of N increases the number of effective bits (ENOB) by square root(N). The scaling in “oversampling” mode for small N conserves all the data but for larger N some resolution is lost in order to fit the filters results into just 16 bits.

On the other hand, if the noise contamination of the samples is not random, but is harmonically related to another signal, this filtering will not improve ADC performance (ENOB does not change.).

44.6.5.1.1. Operation of the Digital Filter

The Digital Filter for ADCn must be enabled by setting FLTCTRLn.FLTEN. The input channel to be filtered, with index k, is set with FLTCTRLn.FLTCHNID = k . The choice between “averaging” or “oversampling” is implemented by FLTCTRLn.FMODE.

Once the Digital Filter is configured, the filter’s control logic waits for an external trigger of channel k to initiate the contiguous scan of samples defined by FLTCTRLn.OVRSAM. The trigger signal for the channel k to be filtered causes the accumulator to be cleared and initiates the first conversion.

On being enabled, the filter control logic has forced the trigger sensitivity into Level Mode (by logic appended to the internal registers associated with the CHNCFG1n.LVNk registers), which enables the selected ADCn to work in burst mode collecting back-to-back samples. The filter control logic will force internally the trigger to one as long as the filter needs more samples. In this way the ADCn itself will collect a burst of samples spaced apart according to CORCTRLn.SAMC.

When the required number of samples have been received and processed by the filter logic, the filter control logic releases the forcing of the channel trigger and starts waiting for a new initial sample (when the external trigger determines the sampling event according to the trigger waveform and the trigger setting to edge / level sensitivity).

At the completion of a burst collection the filter control logic will set the INTFLAGn.FLTRDY flag, which will trigger the ADCn interrupt if the corresponding bit in INTENSET has been set. Then the new output will be available using the CORCHDATAID and CHRDYDAT registers as described in ADC Result Registers.

44.6.5.1.2. Digital Filters Usage

To sustain the back-to-back collection of input samples the digital filter module will block any lower priority ADC conversion requests until the required input samples have been completed. However, any higher priority ADC requests will be processed un-impeded. Therefore, if higher priority requests occur during the filter sequence, the collected samples will no longer be contiguous, injecting unacceptable levels of aperture jitter over the burst and therefore contaminating the filter output. *For this reason,the user should arrange the initiation trigger for the over burst collection to occur while there are no expected interruptions from higher priority ADC conversion requests.*

The register field FLTCTRLn.FMODE determines the filtering mode. For “averaging” mode, FMODE=1. For “oversampling” mode, FMODE=0. When FMODE=1, the register field FLTCTRLn.DATA16EN controls the format of the filter’s output:

- FMODE=1 and DATA16EN=1, then the filter index output data is always left justified with all 16 bits significant.
- If FMODE=1 and DATA16EN=0, then the filter output data is always left justified but with only the first 12 bits significant and the last 4 bits always zero.

If FMODE=0 (i.e. when the Digital Filter works in accumulation mode), DATA16EN bit has no effect, but CHNCFG2n.FRACTk and CHNCFG3n.SIGNk determines the format of the filter output data as shown in Output Data Formats.

44.6.5.1.3. Digital Filters and the APB Bus FIFO

If PFFCTRL.PFFCRn is set for the ADCn using its Digital Filter the FIFO will capture all the back-to-back samples fed into the Digital Filter, save for the last scanned sample, which will be replaced by the filter’s output.

44.6.5.1.4. Finite Bursts

If the trigger of a burst collection is not repeated, then the ADC will collect just one burst. As long as the number of samples defined in OVRSAM is less than 16 then the samples collected will not overflow the FIFO. The first 15 will be directly available from the FIFO and, while the 16 sample is overwritten by the filter, the 16 sample can be recovered using the first 15 samples and the filter’s output.

44.6.5.2. Dedicated Digital Comparator

Each of the four ADC modules has a dedicated Digital Comparator that can alert the application to the capture and conversion of input signals that match predefined criterion(s). When an input conversion produces a “hit” the flag INTFLAGn.CMPHIT is set, which can optionally fire the interrupt associated with ADCn if INTENSETn.CMPHIT is set. The index, k, of the input channel that produce the “hit” is identified in the INTENSETn.CMPINTID[3:0] field.

The Digital Comparator must be enabled by setting CMPCTRLn.CMPEN. Only those input channels that have the associated CHNCFG1n.CHNCMPENk bit set are included in the comparator scans.

The CMPCTRLn register supports these comparisons directly:

- `bit 29` **IEHIHI: Enable Comparison - High Limit, Active High:**
  Setting this bit enables comparison events ADCMPHI ≤ ADC value
- `bit 28` **IEHILO: Enable Comparison - High Limit, Active Low:**
  Setting this bit enables comparison events ADC value < ADCMPHI
- `bit 15` **IEBTWN: Enable Comparison - Active Between Limits:**
  Setting this bit enables comparison events ADCMPLO ≤ ADC Value < ADCMPHI
- `bit 14` **IELOHI: Enable Comparison - Low Limit, Active High:**
  Setting this bit enables comparison events ADCMPLO ≤ ADC Value
- `bit 13` **IELOLO: Enable Comparison - Low Limit, Active Low:**
  Setting this bit enables comparison events ADC Value < ADCMPLO

Since setting more than one of the IE* bits produces a condition that is the <u>and</u> of all conditions enabled, it is not a recommended configuration.

44.6.6. DMA Operation

Table 44-3. DMA Event Trigger Mapping

Instance	Channel	CHCTRLB.TRIG [index]	Presentation
ADC	PFFRDY	61	ADC DMA PFFRDY trigger

The DMA can be programmed to read the APB Bus FIFO when there is available data. The DMA can read from the FIFO when the FIFO is half full or when the FIFO is not empty (i.e. it is "Ready"). The bit **PFFCTRL.PFFRDYDMA** controls this choice.

For the FIFO to work, first enable bit by setting **PFFCTRL.PFFEN** to one. Next enable the FIFO to capture data from one or more ADC Modules in operation by setting **PFFCTRL.**

PFFCR*n* to one for all *n* in use.

44.6.7. Interrupts

The ADC module provides two interrupts to the NVIC: one Global Interrupt (REQ0) and one Module-specific interrupts (REQ1). Global interrupts are serviced by the **CTLINTENCLR, CTLINTENSET,** and **CTLINTFLAG** registers, while each ADC Module's interrupts are serviced by **INTENCLR**, **INTENSET**, and **INTFLAG** registers.

When the ADC's chosen voltage reference status changes (not ready to ready or the reverse) the **CTLINTFLAG.VREFUPD** bit will be set. For this bit to fire the global interrupt **CTLINTENSET**.**VREFUPD** should be set to one. The ADC's global interrupt ISR should always check this flag and disable ADC operation when **CTLINTFLAG.VREFUPD** = 1 and **CTLINTFLAG.VREFRDY** = 0, indicating that the voltage reference is no longer "ready" and therefore no longer useable.

When the startup delay for each ADC Module has elapsed a **CRRDY*n*** bit in the **CTLINTFLAG** register will go high. This can trigger a global interrupt as well if the corresponding bit in **CTLINTENSET** has been set. Note that the **CRRDY*n*** bit in INTFLAG will stay high and therefore continue to fire the global interrupt unless the global interrupt service routine disables the Core Ready Interrupts from firing by setting the **CTLINTENCLR.CRRDY*n*** bits to one before exiting.

Other bits in the **CTLINTFLAG** register are related to the operation of the APB Data FIFO. The **CTLINTFLAG.PFFHFUL** bit goes high when the FIFO is half full. The **CTLINTFLAG.PFFRDY** goes high when the FIFO is "ready", i.e. when it contains data to be ready. FIFO overflow and underflow will set the **CTLINTFLAG.PFFOVF** or **CTLINTFLAG.PFFUNF** bits, respectively.

44.6.8. Events

The ADC can generate events, which can be shared by the Event System (EVSYS) module with other modules, and it can be an event user of events generated by other modules and distributed by the EVSYS module. The ADC can generate an event when a Digital Comparator "Hit" occurs. It can also generate an event when a new data result is "ready" (available for reading.) As a user the ADC often uses events generated by a timer to trigger the capture and conversion of new samples.

There is one Results Ready (ADC_RESRDY) generators. Similarly, there is one Comparator Hit (ADC_CMP) generators. The ADC has eleven "trigger" users in the EVSYS (TRIG*n*, *n* = 0,1,...,10). This provides eleven separate event users which can be used to trigger up to eleven different types of ADC captures.

The ADC Module has a dedicated "ADC Event Control" register (EVCTRL). The register controls the Compare Hit and Results Ready generator. To use generated events as a capture trigger **EVCTRL*n*.STARTEI** must be set. The polarity of trigger inputs can be inverted by setting **EVCTRL*n*.STARTINV.**

44.6.9. Power Management

The ADC will continue to operate in any sleep mode where the selected source clock is running. The ADC's interrupts can be used to wake up the device from sleep modes. Events connected to the event system can trigger other operations in the system without exiting sleep modes.

Note: If several events are connected to the ADC, the enabled action will be taken on any of the incoming events.

44.6.10. Idle and Standby Sleep Mode Operation

The host device supports several modes of operation, starting at Active and ending with Off:

Mode of Operation:	Behavior:	Wake-Up Signal:
Active	CPU running. All memory is active. ADC can run.	None
(CPU) Idle	CPU halted. All memory is active. ADC [1]	All Interrupts
Standby	All logic content retained. ADC [1] NVM in sleep mode. Configurable RAM retention support.	All Interrupts
Hibernate	Only backup domain remains active. ADC cannot run. Configurable RAM retention. Only 32 kHz clock can run.	Backup Interrupts
Backup	Only backup domain remains active. ADC cannot run. Only 32 kHz clock can run.	Backup Interrupts
Off	Everything is off.	External Reset

Note:

1. The CTRLA.ONDEMAND and CTRLA.RUNSTDBY bits control the behavior of the ADC during Idle and Standby sleep modes, in cases where the ADC is enabled (CTRLA.ENABLE = 1). See the following table:

Table 44-4. ADC Idle and Standby Mode Behavior

CTRLA.ENABLE	CTRLA.RUNSTDBY	CTRLA.ONDEMAND	Description
0	x	x	Disabled.
1	0	0	ADC inactive in Idle and Standby mode
1	0	1	Run in Idle and Standby modes only on EVSYS trigger request.
1	1	0	Run in Idle and Standby modes.
1	1	1	Run in Idle and Standby modes.
Note: When CTRLA.ONDEMAND=1, the analog block is always powered-off. When a start request is detected, the system returns from sleep and starts a new conversion. When RUNSTDBY is enabled the GCLK to the ADC continues to run when ONDEMAND=0. With ONDEMAND and RUNSTDBY set the GCLK will be requested. Therefore, there will be an additional delay until the analog front end is ready (CTLINTFLAG.CRRDYn=1).			

44.6.11. Debug Operation

Setting **DBGCTRL.DBGRUN** = 1 enables the ADC to continue operation when the device is halted in debug mode. If this bit is not set, then the ADC is halted in debug mode.

If **DBGCTRL.DBGRUN**=0, then during debug the ADC macro will capture all the triggers that fire and on exit from the debug operation to normal running operation the ADC will perform conversions on the captured event or events based on ADC channel priorities.

During a debug stop all the ADC registers are still write protected as long as **CTRLA.ENABLE**=1.

44.6.12. Register Synchronization

The **SYNCBUSY** register supports the **CTRLA** and **CTRLB** registers. When **CTRLA.ENABLE** has been set no additional writes to this bit are allowed as long as the SYNCYBUSY.ENABLE bit remains high. Additional writes to the **CTRLB** register are not allowed as long as the **SYNCBUSY.CTRLB** bit remains high.

Register synchronization is required for other registers in the GLCK clock domain. These registers can be modified while CTRLA.ENABLE=0, i.e. as long as the ADC is not enabled. Once the ADC is enabled (CTRLA.ENABLE=1) these registers are synchronized to the APB_CLK (Main Clock) domain and are write protected.

44.7. Register Summary

For descriptions and definitions of both Register and bitfield properties, refer to Register Properties.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRLA	31:24								
		23:16								
		15:8								
		7:0	ONDEMAND	RUNSTDBY				ANAEN	ENABLE	SWRST
0x04	CTRLB	31:24								
		23:16								
		15:8	SWCNVEN				STRGEN	TRGSUSP	LSWTRG	GSWTRG
		7:0	SAMP	RQCNVRT			ADCHSEL[3:0]			
0x08	CTRLC	31:24								
		23:16								
		15:8	CNT[15:8]							
		7:0	CNT[7:0]							
0x0C ... 0x0F	Reserved									
0x10	CTRLD	31:24		VREFSEL[2:0]			WKUPCLKCNT[3:0]			
		23:16				ANLEN				CHNEN
		15:8			CTLCKDIV[5:0]					
		7:0								
0x14 ... 0x1F	Reserved									
0x20	CONFIG[0].CORCTRL	31:24		ADCDIV[6:0]						
		23:16		SCNRTDS	STRGLVL		STRGSRC[3:0]			
		15:8	EIRQOVR	EIS[2:0]			SELRES[1:0]		SAMC[9:8]	
		7:0	SAMC[7:0]							
0x24	CONFIG[0].CHNCFG1	31:24			LVL13	LVL12	LVL11	LVL10	LVL9	LVL8
		23:16	LVL7	LVL6	LVL5	LVL4	LVL3	LVL2	LVL1	LVL0
		15:8			CHNCMPE13	CHNCMPE12	CHNCMPE11	CHNCMPE10	CHNCMPE9	CHNCMPE8
		7:0	CHNCMPE7	CHNCMPE6	CHNCMPE5	CHNCMPE4	CHNCMPE3	CHNCMPE2	CHNCMPE1	CHNCMPE0
0x28	CONFIG[0].CHNCFG2	31:24			FRAC13	FRAC12	FRAC11	FRAC10	FRAC9	FRAC8
		23:16	FRAC7	FRAC6	FRAC5	FRAC4	FRAC3	FRAC2	FRAC1	FRAC0
		15:8			CSS13	CSS12	CSS11	CSS10	CSS9	CSS8
		7:0	CSS7	CSS6	CSS5	CSS4	CSS3	CSS2	CSS1	CSS0
0x2C	CONFIG[0].CHNCFG3	31:24			SIGN13	SIGN12	SIGN11	SIGN10	SIGN9	SIGN8
		23:16	SIGN7	SIGN6	SIGN5	SIGN4	SIGN3	SIGN2	SIGN1	SIGN0
		15:8								
		7:0				DIFF4		DIFF2		DIFF0
0x30	CONFIG[0].CHNCFG4	31:24	TRGSRC7[3:0]				TRGSRC6[3:0]			
		23:16	TRGSRC5[3:0]				TRGSRC4[3:0]			
		15:8	TRGSRC3[3:0]				TRGSRC2[3:0]			
		7:0	TRGSRC1[3:0]				TRGSRC0[3:0]			
0x34	CONFIG[0].CHNCFG5	31:24	TRGSRC15[3:0]				TRGSRC14[3:0]			
		23:16	TRGSRC13[3:0]				TRGSRC12[3:0]			
		15:8	TRGSRC11[3:0]				TRGSRC10[3:0]			
		7:0	TRGSRC9[3:0]				TRGSRC8[3:0]			
0x38	CONFIG[0].CALCTRL	31:24	CALBITS[31:24]							
		23:16	CALBITS[23:16]							
		15:8	CALBITS[15:8]							
		7:0	CALBITS[7:0]							
0x3C	EVCTRL0	31:24								
		23:16								
		15:8								
		7:0			CMPEO	RESRDYEO	STARTINV			STARTEI
0x40 ... 0xAF	Reserved									

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0xB0	CMPCTRL	31:24			IEHIHI	IEHILO	ADCMPHI[11:8]			
		23:16	ADCMPHI[7:0]							
		15:8	IEBTWN	IELOHI	IELOLO	CMPEN	ADCMPLO[11:8]			
		7:0	ADCMPLO[7:0]							
0xB4 ... 0xBF	Reserved									
0xC0	FLTCTRL0	31:24								
		23:16								
		15:8			FLTCHNID[3:0]					FLTEN
		7:0				DATA16EN	FMODE	OVRSAM[2:0]		
0xC4 ... 0xCF	Reserved									
0xD0	CORCHDATAID	31:24								
		23:16								
		15:8								
		7:0					CHRDYID[3:0]			
0xD4	CHRDYDAT	31:24					FRACT	SIGN	DIFF	LVL
		23:16								
		15:8	CHRDYDAT[15:8]							
		7:0	CHRDYDAT[7:0]							
0xD8	PFFDATA	31:24	PFFCNT[7:0]							
		23:16	PFFFRACT	PFFSIGN			PFFCHNID[3:0]			
		15:8	PFFDATA[15:8]							
		7:0	PFFDATA[7:0]							
0xDC ... 0xE3	Reserved									
0xE4	PFFCTRL	31:24								
		23:16								PFFRDYDMA
		15:8								
		7:0				PFFCR			PFFEN	
0xE8	SYNCBUSY	31:24								
		23:16								
		15:8								
		7:0						CTRLB	ENABLE	SWRST
0xEC ... 0xFB	Reserved									
0xFC	CTLINTENSET	31:24								
		23:16								
		15:8					PFFHFUL	PFFRDY	PFFOVF	PFFUNF
		7:0	VREFRDY	VREFUPD			CRRDY3	CRRDY2	CRRDY1	CRRDY0
0x0100	CTLINTENCLR	31:24								
		23:16								
		15:8					PFFHFUL	PFFRDY	PFFOVF	PFFUNF
		7:0	VREFRDY	VREFUPD						CRRDY
0x0104	CTLINTFLAG	31:24								
		23:16								
		15:8					PFFHFUL	PFFRDY	PFFOVF	PFFUNF
		7:0	VREFRDY	VREFUPD						CRRDY
0x0108 ... 0x011F	Reserved									
0x0120	INTENCLR	31:24	CHRDY[15:8]							
		23:16	CHRDY[7:0]							
		15:8					EOSRDY	CHNERRC	FLTRDY	CHRDYC
		7:0	SOVFL			CMPHIT				

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0124	INTENSET	31:24	CHRDY[15:8]							
		23:16	CHRDY[7:0]							
		15:8					EOSRDY	CHNERRC	FLTRDY	CHRDYC
		7:0	SOVFL			CMPHIT				
0x0128	INTFLAG	31:24	CHRDY[15:8]							
		23:16	CHRDY[7:0]							
		15:8	CRDYID[3:0]				EOSRDY	CHNERRC	FLTRDY	CHRDYC
		7:0	SOVFL			CMPHIT	CMPINTID[3:0]			
0x012C ... 0x0167	Reserved									
0x0168	DBGCTRL	31:24								
		23:16								
		15:8								
		7:0								DBGRUN

44.7.1. ADC Control Register A

Name: CTRLA
Offset: 0x0
Reset: 0x00000000
Property: PAC Write-Protected

Table 44-5. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	ONDEMAND	RUNSTDBY				ANAEN	ENABLE	SWRST
Access	R/W	R/W				R/W	R/W	R/W
Reset	1	0				0	0	0

Bit 7 – ONDEMAND On Demand Control (idle/standby mode)

The On Demand operation mode allows the ADC to be normally disabled, lowest power mode, until enabled by an EVSYS trigger event. The consequences of this however is that the analog logic is completely powered down when ONDEMAND=1 and clocks are suspended. Upon a EVSYS trigger event, it will require up to 20-50μs for the ADC to warm up meaning that the actual sample/convert sequence will not happen for:

ADC analog input signal sample point in time = (EVSYS Trigger event + CTRLD.WKUPEXP)

Which can be up to 50μs after the initial EVSYS trigger event. After the conversion is complete, if there are no pending EVSYS trigger events the ADC will again power down and the sequence will repeat as before.

When ONDEMAND = 1, the ADC works only on a rising edge EVSYS events. In addition, all other ADC trigger sources other than EVSYS triggers will not be acknowledged and therefore will render the ADC inactive. (See table below)

In Idle and Standby mode, if ONDEMAND = 1 and CTRLA.RUNSTDBY = 1, the ADC will be active and not powered down. If CTRLA.RUNSTDBY=0, the ADC is powered down and ADC clocks are suspended.

Note: CTRLD.WKUPEXP should be set by user to the equivalent of 50 μs or CTRLD.WKUPEXP[xxx] = 50 μs / TAD.

Table 44-6. ONDEMAND Matrix

SYSTEM STATUS	ONDEMAND	RUNSTDBY	ADCn Trigger Source	ADC BEHAVIOR
ACTIVE	1	x	ADCn Global Software Trigger.	Do Nothing
		x	ADCn Global Level Software Trigger	Do Nothing
		x	ADCn SCANTRG	Do Nothing except only when SCAN trigger source selected is EVSYS trigger, then RUN only when EVSYS trigger request is received.
		x	ADCn STRIG Synchronous Trigger	Do Nothing
		x	ADCn Trigger Event(s) from Event System (EVSYS)	RUN only when EVSYS trigger request is received.
	0	x	ADCn Global Software Trigger.	ADC active, Trigger sample/conv and wait for next trigger
		x	ADCn Global Level Software Trigger	ADC active, Trigger sample/conv continuously.
		x	ADCn SCANTRG	ADC active, Trigger sample/conv and wait for next trigger
		x	ADCn STRIG Synchronous Trigger	ADC active, Trigger sample/conv and wait for next trigger
		x	ADCn Trigger Event(s) from Event System (EVSYS)	ADC active, Trigger sample/conv and wait for next trigger.
IDLE, STANDBY	0	1	All triggers working	ADC active
	1	0	n/a	Do nothing
	1	1	ADCn Trigger Event(s) from Event System (EVSYS)	RUN only when EVSYS trigger request is received.

Value	Description
0	The ADC is always on, if ENABLE = 1. (Highly Recommended)
1	See table below.

Bit 6 – RUNSTDBY Run in Idle/Standby Mode

This bit controls how the ADC behaves during Standby Sleep mode.
This bit is not synchronized.

Value	Description
0	Discontinue module operation when device enters idle/standby mode (but do not stop the CTL_CLK to preserve the analog biasing). The ADC will complete the ongoing conversion before disabling the module.
1	The ADC is not stopped in Idle or 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 2 – ANAEN Analog Cores Enable bit

Note: This bit must be set prior to setting any of the CTRLD.ANLENn bits.

Value	Description
0	The analog and bias circuitry for all ADC modules are powered down and clocks disabled. Trigger events will not be serviced.
1	Enables and powers up the individual ADC modules analog logic based on the settings of CTRLD.ANLENn = 1, (where n = 0,1,2,3). Whenever the ADC module"n" exits a power down state to a powered-up state, CTRLA.ANAEN = 1 and CTRLD.ANLENn = 1, then the system hardware will wait until CTRLD.WKUPEXP time has expired before starting an ADCn sample/conversion if a trigger event is pending. The ADCn required warm-up time is 20-50us before it will stabilize for a valid ADC sample/conversion sequence to start. CTRLA.ENABLE does not need to be set to power-up the analog logic and begin the CTRLD.WKUPEXP time out.

Bit 1 – ENABLE A/D Module Operating Enable bit

Notes:

1. The ENABLE bit should be set only after the ADC module has been configured. Changing the configuration bits after enabling the ADC could result in unpredictable behavior.
2. When ENABLE = 0 the internal control logic is reset, and all status generated by the module is cleared. All ADC Module clocks are disabled unless ANAEN = 1. In this case ANLENn = 0 disable clocks and ANLENn = 1 enables them. All ADC registers are available for reading and writing.

Value	Description
0	A/D converter is off.
1	A/D converter module is enabled.

Bit 0 – SWRST Software Reset

Writing a one to the SWRST bit resets the state of the ADC and all the registers to their initial state. The only exception is the DBGCTRL.DBGRUN bit, which will keep its value after a SWRST. The module will be disabled after the reset (ENABLE = 0). When writing a one to SWRST, no other bits in the same register will be written, as SWRST will clear all the bits in the same register.

After writing a one to SWRST, SWRST will read back one until the module and the registers are reset. Any register write access during the ongoing reset will be discarded and an error will be generated. Read access can be performed without an error generated and must return reset value. Writing a one to SWRST will have priority above all other actions and will always happen immediately.

Writing a zero to SWRST has no effect.

Note: During a SWRST, access to registers/bits without SWRST are disallowed until SYNCBUSY.SWRST is cleared by hardware.

Value	Description
0	There is no reset operation ongoing.
1	The reset operation is ongoing.

44.7.2. ADC Control Register B

Name: CTRLB
Offset: 0x4
Reset: 0x00000000
Property: Write-Synchronized via SYNCBUSY.CTRLB

Table 44-7. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
	SWCNVEN				STRGEN	TRGSUSP	LSWTRG	GSWTRG
Access	R/W				R/W	R/W	R/W	R/W/HC
Reset	0				0	0	0	0

Bit	7	6	5	4	3	2	1	0
	SAMP	RQCNVRT			ADCHSEL[3:0]			
Access	R/W	R/W/HC			R/W	R/W	R/W	R/W
Reset	0	0			0	0	0	0

Bit 15 – SWCNVEN Software - Controlled Conversion Enable bit

Note: When **CTRLB.SWCNVEN**=1, all channels sample and conversions are controlled exclusively by SAMP and RQCNVRT. All hardware triggers and GSWTRG and LSWTRG are disabled. It will place all of its channels under the exclusive control of the CTRLB.SAMP and CTRLB.RQCNVRT bits.

Value	Description
0	Software controlled conversions disable. Traditional ADC hardware triggers as defined by CHNCFG4n and CHNCFG5n will be active if enabled.
1	SAMP & RQCNVRT bits control entire ADC module sample and convert respectively for all channels on ADC module. Setting this bit blocks all other hardware triggers events defined by CHNCFG4n and CHNCFG5n.

Bit 11 – STRGEN Synchronous Trigger Enable bit

Note: This bit is ignored if CTRLB.SWCNVEN=1.

Value	Description
0	Disable automatic ADC hardware CTRLC.CNT counter driven synchronous ADC triggers.
1	Enable automated hardware synchronous trigger period defined by CTRLC.CNT register.

Bit 10 – TRGSUSP Trigger Suspend bit

Value	Description
0	Trigger suspend disabled. Triggers if enabled and selected will occur.

Value	Description
1	Blocks triggers from starting new ADC conversions but does not disable the ADC Modules or disable trigger capture, (i.e. persistent last trigger event is latched). If trigger capture during trigger suspend is not desired for any channels connected to a certain ADCn, then CTRLD.CHNENn must be cleared then set in that order prior to resetting TRGSUSP. Pulsing down and up CTRLD.CHNENn prior to resetting TRGSUSP will clear ALL pending triggers for ADCn.

Bit 9 – LSWTRG Level Global Software Trigger bit

Notes:

1. This bit is ignored if SWCNVEN=1.
2. If SWCNVEN=0, this bit is NOT self-clearing and is meant to allow the user software to implement continuous sample/conversions on the associated analog input channel.

Value	Description
0	Global Level Software Trigger disabled.
1	Trigger A/D conversions for ADC analog input(s) "y" that have selected LSWTRG bit as the trigger signal via the associated CHNCFG4n.TRGSRC[y] or ADC CHNCFG50.TRGSRC[y] = 0b0010 or via the CORCTRLn.STRGSRC value where (n=ADC 0,1,2,3).

Bit 8 – GSWTRG Global Software Trigger bit

This software settable bit will trigger ADC sample/conversion sequences for ADC inputs that have selected the GSWTRG bit as the trigger signal via the associated CHNCFG4n.TRGSRC[y]= 0b0001 value, or via the CORCTRLn.STRGSRC value. This bit is auto cleared on the next APB clock cycle and is meant to implement single conversions on trigger edge-sensitive channels.

Notes:

1. This bit is ignored if SWCNVEN=1.
2. If SWCNVEN=0, this bit is auto cleared by hardware after sampling cycle has been triggered and is meant to implement single sample/conversions on trigger edge-sensitive channels.

Value	Description
0	Disable Global Software Trigger.
1	If SWCNVEN=0, this bit is auto cleared by hardware after sampling cycle has been triggered and is meant to implement single sample/conversions on trigger edge-sensitive channels.

Bit 7 – SAMP Enable the Analog Mux Input and Start Sampling

Notes:

1. This bit is ignored if SWCNVEN = 0.
2. ADCHSEL[3:0] must be initialized at or before when the SAMP bit is set.
3. The SAMP bit will keep the S&H circuit in Sample mode until the bit is cleared by the users software. Also, usage of the SAMP bit will cause settings of respective CORCTRLx.SAMC<9:0> bits to be ignored.
4. The SAMP bit is not a self-clearing bit and it is the responsibility of application software to clear this bit but only after setting the RQCNVRT bit to start the analog-to-digital conversion.
5. When the SAMP and RQCNVRT bits are used by software routines, all TRGSRCx<4:0> bits and STRGSRC<4:0> bits for the designated ADC module defined by ADCHSEL[3:0] should be set to ‘00000’ to disable all hardware triggers and prevent them from interfering with the software-controlled sampling command signal SAMP and with the software controlled trigger RQCNVRT.
6. The SAMP bit should only be used in conjunction with RQCNVRT for user software-controlled sampling and triggering.

Value	Description
0	ADC is not software-controlled sampling the ADC module and channel selected by ADCHSEL[3:0].
1	ADC is sampling and remain sampling for as long as user has this bit set in SW. The sampled analog input for a given ADCn module is defined by the channel selected by ADCHSEL[3:0] provided SWCNVEN = 1.

Bit 6 – RQCNVRT Request Individual ADC Conversion

Notes:

1. If user's software sets RQCNVRT = 1, an ADC conversion will begin immediately and terminate the sampling period defined by CORCTRLn.SAMC. This bit is cleared by hardware after the conversion is complete.
2. If SWCNVEN = 1, after setting this bit the users software must immediately clear the SAMP bit.
3. This bit is ignored if SWCNVEN = 0, (i.e. if software-controlled sample/conversions are disabled).

Value	Description
0	ADC is not converting if SWCNVEN = 1 or user previously set this bit and the previous ADC conversion is complete for the analog channel that was defined by ADCHSEL[3:0]
1	Terminate sampling and begin conversion of ADC and analog channel defined by ADCHSEL[3:0]. This bit is cleared by hardware when the conversion is complete.

Bits 3:0 – ADCHSEL[3:0] ADC Module Channel Input Select bits

This binary encoded bit field selects the ADC analog input to be sampled and converted respectively by the SAMP and RQCNVRT bit if SWCNVEN = 1.

Note: These bits are ignored if SWCNVEN = 0.

ADC Modules	ADCHSEL[3:0]	Analog Channel Input
ADC (AIN12:0)	0x0	VINP0
	0x1	VINP1
	0x2	VINP2
	0x3	VINP3
	0x4	VINP4
	0x5	VINP5
	0X6	VINP6
	0x7	VINP7
	0x8	VINP8
	0x9	VINP9
	0xA	VINP10
	0xB	VINP11
	0xC	Temp Sensor
	0xD	1.2v IVREF

44.7.3. ADC Control Register C

Name: CTRLC
Offset: 0x8
Reset: 0x00000000
Property: Write-Protected, Enable-Protected

Table 44-8. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
	CNT[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
	CNT[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 – CNT[15:0] This bit-field selects an alternate trigger source delay counter
Free-running counter based on CTL_CLK times out when it reaches this value. At time out, the STRIG synchronous trigger will fire.
Notes:
1. This register is not valid unless either [CORCTRLn.STRGSRC=0x4 plus CHNCFG40/50.TRGSRCx=0x3 plus CHNCFG20.CSSx=1] or [CHNCFG40/50.TRGSRCx=0x4 plus CTRLB.SWCNVEN=0] for Synchronous Trigger from CTRLC.CNT.
2. CTL_CLK = GCLK_ADC / (CTRLD.CTLCKDIV+1)
3. This bit is Enabled Protected. (Writes are ignored when CTRLA.ENABLE = 1 and return a bus error).

44.7.4. ADC Control Register D

Name: CTRLD
Offset: 0x10
Reset: 0x00000000
Property: Write-Protected, Enable-Protected

Table 44-9. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
		VREFSEL[2:0]			WKUPCLKCNT[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
				ANLEN				CHNEN
Access				R/W				R/W
Reset				0				0

Bit	15	14	13	12	11	10	9	8
			CTLCKDIV[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
Access								
Reset								

Bits 30:28 – VREFSEL[2:0] VREF Input Selection

VREFSEL[2:0]	ADREF+	ADREF-
`000`	A_{VDD}	A_{VSS}
`001`	External V_{REF}H	A_{VSS}
`010-111`	Reserved	Reserved

Note:
1. This bit is Enabled Protected. Writes are ignored when CTRLA.ENABLE = 1 and return a bus error.

Bits 27:24 – WKUPCLKCNT[3:0] Wake-Up TAD Clock Count bits

These bits represent the number of ADC TAD clocks required to warm-up the ADC module before it can perform conversion. Although the clocks are specific to each ADC, the WKUPCLKCNT bit is common to all ADC modules.
1111 = 215 = 32,768 TAD clocks
* * *
* * *
0110 = 26 = 64 TAD clocks
0101 = 25 = 32 TAD clocks
0100 = 24 = 16 TAD clocks
0011 = 24 = 16 TAD clocks

0010 = 24 = 16 TAD clocks
0001 = 24 = 16 TAD clocks
0000 = 24 = 16 TAD clocks

Notes:

1. Minimum required ADC warm-up time, (i.e., WKUPCLKCNT), is 50 µs. (i.e., TAD * WKUPCLKCNT).
2. TAD = [1 / ((GCLK_ADC Freq / (CTRLD.CTLCKDIV+1)) / (2 * CORCTRL.ADCDIV))] or (GCLK_ADC Period * (CTRLD.CTLCKDIV+1)) * (2 * ADCDIV).
3. These bits are Enabled Protected. Writes are ignored when CTRLA.ENABLE = 1 and return a bus error.

Bit 20 – ANLEN Analog and Bias Circuitry Enable for the ADC Module

Notes:

1. The CALCTRL register must be initialized by the users software to the factory-provided values in the CAL OTP Register FCCFG65 before setting ANLEN=1.
2. CTRLA.ANAEN=1 must be set prior to setting ANLEN=1.
3. This bit is Enabled Protected. Writes are ignored when CTRLA.ENABLE = 1 and return a bus error.

Value	Description
0	ADC Analog and bias circuitry powered down and clocks suspended. Analog logic power saving mode.
1	Analog and bias circuitry enabled. Once the analog and bias circuit is enabled CTRLA.ANLEN=1 and ANLEN, the ADC module needs a warm-up time, as defined by the WKUPCLKCNT<3:0> bits.

Bit 16 – CHNEN ADC Module Digital Enable

Notes:

1. This bit is Enabled Protected. Writes are ignored when CTRLA.ENABLE = 1 and return a bus error.
2. ANLEN qualifies CHNEN: If ANLEN =0, then digital logic is also disabled.

Value	Description
0	ADC digital logic disabled. No trigger, sample or conversion events will be processed. (power-saving mode with fast 2 TAD clock wakeup provided ANLEN = 1).
1	ADC digital logic enabled (required for active operation).

Bits 13:8 – CTLCKDIV[5:0] A/D Clock Source to Control Clock Divider

The CTLCKDIV bit field divides the GCLK_ADC input clock into the ADC Module control clock CTL_CLK scaled by CTRLD.CTLCKDIV with period T_Q.

Note: This bit is Enabled Protected. Writes are ignored when CTRLA.ENABLE = 1 and returns a bus error.

Value	Description
111111	$T_{GCLK\_ADCx}\cdot(CTLCKDIV[5:0]+1) = 64\cdot T_{GCLK\_ADCx} = T_Q$
111110	$T_{GCLK\_ADCx}\cdot(CTLCKDIV[5:0]+1) = 63\cdot T_{GCLK\_ADCx} = T_Q$
. . .	...
000100	$T_{GCLK\_ADCx}\cdot(CTLCKDIV[5:0]+1) = 5\cdot T_{GCLK\_ADCx} = T_Q$
000011	$T_{GCLK\_ADCx}\cdot(CTLCKDIV[5:0]+1) = 4\cdot T_{GCLK\_ADCx} = T_Q$
000010	$T_{GCLK\_ADCx}\cdot(CTLCKDIV[5:0]+1) = 3\cdot T_{GCLK\_ADCx} = T_Q$
000001	$T_{GCLK\_ADCx}\cdot(CTLCKDIV[5:0]+1) = 2\cdot T_{GCLK\_ADCx} = T_Q$
000000	$T_{GCLK\_ADCx}\cdot(CTLCKDIV[5:0]+1) = 1\cdot T_{GCLK\_ADCx} = T_Q$

44.7.5. ADC Module Core Control Registers

Name: CONFIG[0].CORCTRL
Offset: 0x20
Reset: 0x00000C00
Property: Write-Protected, Enable-Protected

Table 44-10. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
		ADCDIV[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
		SCNRTDS	STRGLVL		STRGSRC[3:0]			
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
	EIRQOVR	EIS[2:0]			SELRES[1:0]		SAMC[9:8]	
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	1	1	0	0

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

Bits 30:24 – ADCDIV[6:0] Division Ratio for ADC Clock

ADCDIV provides the ADC analog functional clock referred to as the ADC clock or TAD, (i.e., the period of the ADC clock). It divides CTL_CLK output clock from (GCLK_ADC Period * (CTRLD.CTLCKDIV+1)) with period = TQ to produce the resulting ADC_CLK for ADC sampling and converting. It will have the period TAD given by the formula:
TAD = (GCLK_ADCx Period * (CTRLD.CTLCKDIV+1)) * (2 * ADCDIV)
Note: The ADC throughput rate F_{TPR} = [1 / ((SAMC+2) * TAD) + ((#bits of resolution selected+1)*TAD)] / # of active AINx inputs.

Example 1:
TAD = 1/ADC_CLK = 1/(GCLK_ADCx Period * (CTRLD.CTLCKDIV+1)) * (2 * ADCDIV)
TAD = 1/ADC_CLK = 1/60 MHz = 16.6667 ns, SAMC = 0x4, SELRES = 0x3 = 12 bits, AIN0 active only.
ADC throughput rate:
FTPR = [1 / (6TAD + 13TAD)] / 1
= 1 / 19TAD
= 1 / (19 * 16.6667e-9)
= 3.157894 msps
Example 2: (Non Interleaved Mode)
TAD = 1/ADC_CLK = 1/(GCLK_ADCx Period * (CTRLD.CTLCKDIV+1)) * (2 * ADCDIV)
TAD = 1/ADC_CLK = 1/60 MHz = 16.6667 ns, SAMC = 0x4, SELRES = 0x3 = 12 bits, AIN0 and AIN3 in single ended mode active.

ADC throughput rate:
FTPR = [1 / (6TAD + 13TAD)] / 2
= (1 / 19TAD) / 2
= (1 / (16 * 16.6667e-9)) / 2
= 1.578947 msps

Notes:

1. This bit is Enabled Protected : (Writes are ignored when CTRLA.ENABLE = 1) . Returns a bus error.
2. The minimum sample rate for ADC is Ftpr >= 100 ksps.

Value	Description
1111111	$254 \cdot T_Q = T_{AD}$
. . .	...
0000011	$6 \cdot T_Q = T_{AD}$
0000010	$4 \cdot T_Q = T_{AD}$
0000001	$2 \cdot T_Q = T_{AD}$
0000000	Reserved

Bit 22 – SCNRTDS SCAN Retrigger Disable for ADC

Note:

1. This bit is Enabled Protected: (Writes are ignored when CTRLA.ENABLE = 1). Returns a bus error.

Value	Description
0	Allows the scan cycle to restart from the beginning, lowest CSSn channel even before all of the scan channels have been measured if a new scan trigger arrives before then. If this happens then the INTENCLRn.EOSRDY flag will be set when the current scan is interrupted, and the new scan starts from the beginning of the lowest CSSn channel selected.
1	Prevents an early scan trigger (arriving before of the end of the current scan) from starting a new scan cycle. The scan will includes all channels associated with ADCn which have their CHNCFG4n.TRGSRCk set to point to the Scan Trigger, i.e., CHNCFG4n.TRGSRCy[3:0] = 4'b0011, and have their associated CHNCFG2n.CSSk bit set.

Bit 21 – STRGLVL Scan Trigger High Level Sensitivity for the ScanTrigger of the ADCn (STRIGn)

Note:

1. This bit is Enabled Protected (Writes are ignored when CTRLA.ENABLE = 1). Returns a bus error.

STRGLVL functions as follows:

Value	Description
0	SCANTRG is Positive Edge Active(the power-up value for backwards compatibility). A positive edge on the SCANTRG will initiate a single but complete scan of all included channels.
1	SCANTRG is High Level Active. So long as SCANTRG stays high, the entire scan will re-trigger.

Bits 19:16 – STRGSRC[3:0] SCAN Trigger Source Select for the SCANTRG

These bits select the trigger source for the scan trigger SCANTRG serving ADCn. The trigger STRIGn serves all the channels k which are associated to ADCn, have their CHNCFG{4|5}n.TRGSRCk = 4'h0011, and have their Channel Scan Select bit set (CHNCFG2n.CSSk = 1'b1).

Note:

1. This bit is Enabled Protected: (Writes are ignored when CTRLA.ENABLE = 1). Returns a bus error.

Value	Description
0000	No trigger (NOP)
0001	Global Software Trigger (CTRLB.GSWTRG) - self-cleared on the next APB clock cycle.
0010	Level Software Trigger (CTRLB.LSWTRG) - NOT self-cleared.
0011	Reserved

Value	Description
0100	Synchronous Trigger (CTRLC.CNT) **Notes:** In order to utilize CORCTRL.STRGSRC = 0x4, the user must configure the following registers: • CTRLC.CNT • CHNCFG2n.CSSx • CHNCFG4n.TRGSRCx = 0x3 • CTRLB.SWCNVEN = 0
0101 – 1111	ADC Trigger Event User 0 – 10

Bit 15 – EIRQOVR Interrupt Type Selection

Note:

1. This bit is Enabled Protected: (Writes are ignored when CTRLA.ENABLE = 1). Returns a bus error.

Value	Description
0	Use normal ADC interrupts
1	Use early ADC interrupts as defined by CORCTRLn.EIS bits

Bits 14:12 – EIS[2:0] Early Interrupt Select Bits for ADCn

These bits select the number of core clocks and TAD clocks prior to the end of conversion at which the early interrupt is generated. All channels serviced by ADCn share the same EIS setting.
The interrupt is generated ((EIS +1) x TAD) ADC Module clocks prior to end of conversion.

Notes:

1. This bit is Enabled Protected: (Writes are ignored when CTRLA.ENABLE = 1). Returns a bus error.
2. Early interrupt is a feature that can be useful in ADC very high speed data acquisitions to mask the MCU interrupt latency delay.
   Example: CPU MCLK =200 MHz, ADC CTL_CLK = 60 MHz and ADC TAD Clk = 30 MHz.

   Hypothetically, if it took the CPU 20 MCLK cycles to service an interrupt, then theoretically the ADC early interrupt EIS could be set to 4, so that the ISR latency was masked. The CPU could then read the ADC result almost immediately from within the Interrupt Service Routine (ISR), when the result was ready, without any concern for overwriting the current result or FIFO with the next ADC conversion result.

Notes: Depending on the bit resolution selection field SELRES, the allowed maximum EIS values are as follows:

- 12-bit Resolution, all 8 possible settings are allowed, 0 - 7
- 10-bit Resolution, all 8 possible settings are allowed, 0 - 7
- 8-bit Resolution, only the 6 lowest settings are allowed, 0 - 5
- 6-bit Resolution, only the 4 lowest settings are allowed, 0 - 3

The hardware will utilize the maximum allowed EIS setting if the user sets the EIS value that is too big.

CAUTION Do not set the EIS to large or the interrupt can occur before the ADC conversion is complete and the CPU could read the previous ADC conversion result instead of the one in progress.

Value	Description
0x0	Generate interrupt 1 TAD clock before end of ADC conversion
0x1	Generate interrupt 2 TAD clocks before end of ADC conversion
0x2	Generate interrupt 3 TAD clocks before end of ADC conversion
0x3	Generate interrupt 4 TAD clocks before end of ADC conversion

Value	Description
`0x4`	Generate interrupt 5 TAD clocks before end of ADC conversion
`0x5`	Generate interrupt 6 TAD clocks before end of ADC conversion
`0x6`	Generate interrupt 7 TAD clocks before end of ADC conversion
`0x7`	Generate interrupt 8 TAD clocks before end of ADC conversion

Bits 11:10 – SELRES[1:0] ADC Resolution for the ADCn

Value	Description
`00`	6 bits
`01`	8 bits
`10`	10 bits
`11`	12 bits (power-on default)

Note:
1. This bit is Enabled Protected: (Writes are ignored when CTRLA.ENABLE = 1). Returns a bus error.

Bits 9:0 – SAMC[9:0] Sample Count
The sample time required depends on the external analog signal source impedance. Refer to the ADC electrical characteristics of SAMC values required based on external source impedance. If the external analog source impedance is unknown or if your getting inconsistent ADC results consider increasing the SAMC sample time at the cost of a lower ADC throughput rate of course.
ADCn Throughput Rate:
F_{TPR} = [1 / ((SAMC value +2) * TAD) + ((#bits of resolution selected+1)*TAD)] / # of active AINx inputs on ADCn

Value	Description
`1111111111`	1025 T_{AD}n
`...`	...
`0000000001`	3 T_{AD}n
`0000000000`	Reserved

Notes:
1. All channels serviced by ADCn share the same SAMC setting.
2. The sampling sequence starts with a trigger event.
3. The internal sampling cap is not discharged between samples. These bits are Enabled Protected: Writes are ignored when CTRLA.ENABLE = 1 returns a bus error.
4. When IVREF is to be sampled (Internal - ADC3 Module - Channel 6), (SAMC value + 2) * TAD must be ≧ 30 us.

44.7.6. ADC Channel Configuration Registers 1

Name: CONFIG[0].CHNCFG1
Offset: 0x24
Reset: 0x00000000
Property: PAC Write-Protected, Enable-Protected

Table 44-11. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
			LVL13	LVL12	LVL11	LVL10	LVL9	LVL8
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
	LVL7	LVL6	LVL5	LVL4	LVL3	LVL2	LVL1	LVL0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
			CHNCMPE13	CHNCMPE12	CHNCMPE11	CHNCMPE10	CHNCMPE9	CHNCMPE8
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
	CHNCMPE7	CHNCMPE6	CHNCMPE5	CHNCMPE4	CHNCMPE3	CHNCMPE2	CHNCMPE1	CHNCMPE0
Access	R/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 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29 – LVLk Trigger Level for Input k
Bit 29 :LVL13 Scan Level Trigger mode select for ADC Module 0, analog input for the 1.2v IVREF
Bit 28 :LVL12 Scan Level Trigger mode select for ADC Module 0, analog input from Temperature Sensor
Bit 27 :LVL11 Scan Level Trigger mode select for ADC Module 0, analog input AIN11 (3)
Bit 26 :LVL10 Scan Level Trigger mode select for ADC Module 0, analog input AIN10 (3)
Bit 25 :LVL9 Scan Level Trigger mode select for ADC Module 0, analog input AIN9
Bit 24 :LVL8 Scan Level Trigger mode select for ADC Module 0, analog input AIN8
Bit 23 :LVL7 Scan Level Trigger mode select for ADC Module 0, analog input AIN7
Bit 22 :LVL6 Scan Level Trigger mode select for ADC Module 0, analog input AIN6
Bit 21 :LVL5 Scan Level Trigger mode select for ADC Module 0, analog input AIN5
Bit 20 :LVL4 Scan Level Trigger mode select for ADC Module 0, analog input AIN4
Bit 19 :LVL3 Scan Level Trigger mode select for ADC Module 0, analog input AIN3
Bit 18 :LVL2 Scan Level Trigger mode select for ADC Module 0, analog input AIN2
Bit 17 :LVL1 Scan Level Trigger mode select for ADC Module 0, analog input AIN1
Bit 16 :LVL0 Scan Level Trigger mode select for ADC Module 0, analog input AIN0

Notes:
1. These bits are ignored if CTRLB.SWCNVEN = 1.
2. These bits are Enabled Protected (Writes are ignored when CTRLA.ENABLE = 1 returns a bus error).
3. These bits not available in 64-pin package.

Value	Description
0	Rising edge scan trigger mode select. A rising edge trigger event will initiate a single but complete scan of all included scan channels defined in CHNCFG21.CSSy (default).
1	Level scan trigger mode select. As long as the trigger event stays a logic high when The corresponding CHNCFG40/ CHNCFG50 TRGSRCx = 0b0011, (SCANTRG - Scan Trigger Select), the entire scan will re-trigger continuously.

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13 – CHNCMPEk Channel k Compare Enable

Bit 13 :CHNCMPE13 Enable digital comparator for processing ADC conversion results for the 1.2v IVREF
Bit 12 :CHNCMPE12 Enable digital comparator for processing ADC conversion results for the Temperature Sensor
Bit 11 :CHNCMPE11 Enable digital comparator for processing ADC conversion results for AIN11 (3)
Bit 10 :CHNCMPE10 Enable digital comparator for processing ADC conversion results for AIN10 (3)
Bit 9 :CHNCMPE9 Enable digital comparator for processing ADC conversion results for AIN9
Bit 8 :CHNCMPE8 Enable digital comparator for processing ADC conversion results for AIN8
Bit 7 :CHNCMPE7 Enable digital comparator for processing ADC conversion results for AIN7
Bit 6 :CHNCMPE6 Enable digital comparator for processing ADC conversion results for AIN6
Bit 5 :CHNCMPE5 Enable digital comparator for processing ADC conversion results for AIN5
Bit 4 :CHNCMPE4 Enable digital comparator for processing ADC conversion results for AIN4
Bit 3 :CHNCMPE3 Enable digital comparator for processing ADC conversion results for AIN3
Bit 2 :CHNCMPE2 Enable digital comparator for processing ADC conversion results for AIN2
Bit 1 :CHNCMPE1 Enable digital comparator for processing ADC conversion results for AIN1
Bit 0 :CHNCMPE0 Enable digital comparator for processing ADC conversion results for AIN0
Notes:
1. In addition to setting the CHNCMPENn bit in this register, the associated ADC Digital Comparator must be also properly configured in its Digital Comparator Control Register, CMPCTRL0 as well as EVCTRL0.CMPEO.
2. These bits are ignored if CTRLB.SWCNVEN = 1 and are Enabled Protected (Writes are ignored when CTRLA.ENABLE = 1. Returns a bus error.)
3. These bits not available in 64-pin package.

Value	Description
0	ADC analog channel is not monitored by ADC internal digital comparator
1	ADC analog channel conversion result is monitored by ADC internal digital comparator

44.7.7. ADC Channel Configuration Registers 2

Name: CONFIG[0].CHNCFG2
Offset: 0x28
Reset: 0x00000000
Property: PAC Write-Protected, Enable-Protected

Table 44-12. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
			FRAC13	FRAC12	FRAC11	FRAC10	FRAC9	FRAC8
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
	FRAC7	FRAC6	FRAC5	FRAC4	FRAC3	FRAC2	FRAC1	FRAC0
Access	R/W	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
			CSS13	CSS12	CSS11	CSS10	CSS9	CSS8
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
	CSS7	CSS6	CSS5	CSS4	CSS3	CSS2	CSS1	CSS0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29 – FRACk Fractional Data Output Format Enable for Channel k

Bit 29: FRAC13 Fractional Data Output Format select for ADC Module 0, analog input AIN13
Bit 28: FRAC12 Fractional Data Output Format select for ADC Module 0, analog input AIN12
Bit 27: FRAC11 Fractional Data Output Format select for ADC Module 0, analog input AIN11 (4)
Bit 26: FRAC10 Fractional Data Output Format select for ADC Module 0, analog input AIN10 (4)
Bit 25: FRAC9 Fractional Data Output Format select for ADC Module 0, analog input AIN9
Bit 24: FRAC8 Fractional Data Output Format select for ADC Module 0, analog input AIN8
Bit 23: FRAC7 Fractional Data Output Format select for ADC Module 0, analog input AIN7
Bit 22: FRAC6 Fractional Data Output Format select for ADC Module 0, analog input AIN6
Bit 21: FRAC5 Fractional Data Output Format select for ADC Module 0, analog input AIN5
Bit 20: FRAC4 Fractional Data Output Format select for ADC Module 0, analog input AIN4
Bit 19: FRAC3 Fractional Data Output Format select for ADC Module 0, analog input AIN3
Bit 18: FRAC2 Fractional Data Output Format select for ADC Module 0, analog input AIN2
Bit 17: FRAC1 Fractional Data Output Format select for ADC Module 0, analog input AIN1
Bit 16: FRAC0 Fractional Data Output Format select for ADC Module 0, analog input AIN0

Notes:
1. These bits are ignored if CTRLB.SWCNVEN = 1.
2. These bits are Enabled Protected (Writes are ignored when CTRLA.ENABLE = 1 returns a bus error).
3. Fractional format is very useful in Multiply and Accumulate (MAC) operations because result overruns cannot happen as any given result is less than 1.
4. These bits are not available in the 64-pin package.

Value	Description
0	ADC channel "AINn" result output format is unsigned integer
1	ADC channel "AINn" result output format is fractional

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13 – CSSk Channel Scan Select for Channel k

Bit 13: CSS13 Channel Scan Select for ADC Module 0, analog input AIN13
Bit 12: CSS12 Channel Scan Select for ADC Module 0, analog input AIN12
Bit 11: CSS11 Channel Scan Select for ADC Module 0, analog input AIN11 (4)
Bit 10: CSS10 Channel Scan Select for ADC Module 0, analog input AIN10 (4)
Bit 9: CSS9 Channel Scan Select for ADC Module 0, analog input AIN9
Bit 8: CSS8 Channel Scan Select for ADC Module 0, analog input AIN8
Bit 7: CSS7 Channel Scan Select for ADC Module 0, analog input AIN7
Bit 6: CSS6 Channel Scan Select for ADC Module 0, analog input AIN6
Bit 5: CSS5 Channel Scan Select for ADC Module 0, analog input AIN5
Bit 4: CSS4 Channel Scan Select for ADC Module 0, analog input AIN4
Bit 3: CSS3 Channel Scan Select for ADC Module 0, analog input AIN3
Bit 2: CSS2 Channel Scan Select for ADC Module 0, analog input AIN2
Bit 1: CSS1 Channel Scan Select for ADC Module 0, analog input AIN1
Bit 0: CSS0 Channel Scan Select for ADC Module 0, analog input AIN0

Notes:
1. Scan mode requires programming of CORCTRL0.SCANTRG and CORCTRL0.STRGSRC to select scan trigger source.
2. ADC Scan sequence is always least to most significant analog input.
3. These register bits are ignored if CTRLB.SWCNVEN = 1 and they are enabled Protected (Writes are ignored when CTRLA.ENABLE = 1 returns a bus error).
4. These bits are not available in the 64-pin package.

Value	Description
0	ADC analog input channel "AINn" is not part of ADC scan list
1	Add ADC analog input channel "AINn" to scan list

44.7.8. ADC Channel Configuration Registers 3

Name: CONFIG[0].CHNCFG3
Offset: 0x2C
Reset: 0x00000000
Property: PAC Write-Protected, Enable-Protected

Table 44-13. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
			SIGN13	SIGN12	SIGN11	SIGN10	SIGN9	SIGN8
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
	SIGN7	SIGN6	SIGN5	SIGN4	SIGN3	SIGN2	SIGN1	SIGN0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
				DIFF4		DIFF2		DIFF0
Access				R/W		R/W		R/W
Reset				0		0		0

Bits 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29 – SIGNk Signed Data Output Format Enable for channel k

Bit 29: SIGN13 Signed Data Output Format Enable for ADC Module 0, analog input AIN13
Bit 28: SIGN12 Signed Data Output Format Enable for ADC Module 0, analog input AIN12
Bit 27: SIGN11 Signed Data Output Format Enable for ADC Module 0, analog input AIN11 (1)
Bit 26: SIGN10 Signed Data Output Format Enable for ADC Module 0, analog input AIN10 (1)
Bit 25: SIGN9 Signed Data Output Format Enable for ADC Module 0, analog input AIN9
Bit 24: SIGN8 Signed Data Output Format Enable for ADC Module 0, analog input AIN8
Bit 23: SIGN7 Signed Data Output Format Enable for ADC Module 0, analog input AIN7
Bit 22: SIGN6 Signed Data Output Format Enable for ADC Module 0, analog input AIN6
Bit 21: SIGN5 Signed Data Output Format Enable for ADC Module 0, analog input AIN5
Bit 20: SIGN4 Signed Data Output Format Enable for ADC Module 0, analog input AIN4
Bit 19: SIGN3 Signed Data Output Format Enable for ADC Module 0, analog input AIN3
Bit 18: SIGN2 Signed Data Output Format Enable for ADC Module 0, analog input AIN2
Bit 17: SIGN1 Signed Data Output Format Enable for ADC Module 0, analog input AIN1
Bit 16: SIGN0 Signed Data Output Format Enable for ADC Module 0, analog input AIN0
Note:

1. These bits are not available in the 64-pin package.

Value	Description
0	Output format is unsigned

Value	Description
1	Output format is signed

Bit 4 – DIFF4 Differential Mode Enable for channel k

Bit 4: DIFF4 Differential Mode Enable for ADC Module 0, analog input AIN4

Value	Description
0	Channel is connected in single-ended mode
1	Channel is connected in differential mode

Bit 2 – DIFF2 Differential Mode Enable for channel k

Bit 2: DIFF2 Differential Mode Enable for ADC Module 0, analog input AIN2

Value	Description
0	Channel is connected in single-ended mode
1	Channel is connected in differential mode

Bit 0 – DIFF0 Differential Mode Enable for channel k

Bit 0: DIFF0 Differential Mode Enable for ADC Module 0, analog input AIN0

Value	Description
0	Channel is connected in single-ended mode
1	Channel is connected in differential mode

44.7.9. ADC Channel Configuration Registers 4

Name: CONFIG[0].CHNCFG4
Offset: 0x30
Reset: 0x00000000
Property: PAC Write-Protected, Enable-Protected

Table 44-14. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	TRGSRC7[3:0]				TRGSRC6[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
	TRGSRC5[3:0]				TRGSRC4[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
	TRGSRC3[3:0]				TRGSRC2[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
	TRGSRC1[3:0]				TRGSRC0[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

Bits 0:3, 4:7, 8:11, 12:15, 16:19, 20:23, 24:27, 28:31 – TRGSRCk Conversion Trigger Source for channel k

Bits 31-28: TRGSRC7 ADC Sample/Conversion Trigger Source for analog input channel AIN7
Bits 27-24: TRGSRC6 ADC Sample/Conversion Trigger Source for analog input channel AIN6
Bits 23-20: TRGSRC5 ADC Sample/Conversion Trigger Source for analog input channel AIN5
Bits 19-16: TRGSRC4 ADC Sample/Conversion Trigger Source for analog input channel AIN4
Bits 15-12: TRGSRC3 ADC Sample/Conversion Trigger Source for analog input channel AIN3
Bits 11-8: TRGSRC2 ADC Sample/Conversion Trigger Source for analog input channel AIN2
Bits 7-4: TRGSRC1 ADC Sample/Conversion Trigger Source for analog input channel AIN1
Bits 3-0: TRGSRC0 ADC Sample/Conversion Trigger Source for analog input channel AIN0

TRGSRCn	Description (Trigger events start ADC sample/conversion sequence)
0000	No Trigger
0001	ADC Global Software Trigger. (Requires CTRLB.GSWTRG =1, CTRLB.GSWTRG is self-clearing after trigger event.)
0010	ADC Global Level Software Trigger (Requires CTRLB.LSWTRG=1)
0011	ADC SCANTRG - Scan Trigger. (Requires CHNCFG20.CSSy be configured accordingly)
0100	ADC STRIG Synchronous Trigger
0101-1111	ADC Trigger Event(s) from Event System (EVSYS)

If CHNCFG4.TRGSRCy = 0b101 thru 0b1111:

Event System (EVSYS)			ADC
CHANNELx.EVGEN	CHANNELn.PATH(2)	USERm.CHANNEL	CHNCFG4.TRGSRCy
User Selected ADC Trigger Event Source	= 0x2 Asynchronous path	78	0b0101
		79	0b0110
		80	0b0111
		81	0b1000
		82	0b1001
		83	0b1010
		84	0b1011
		85	0b1100
		86	0b1101
		87	0b1110
		88	0b1111

Notes:

1. SCANTRG in turn requires programming of CORCTRL0.STRGSRC to select its trigger source. Also, the appropriate CHNCFG20.CSSy bit must be set to include channel "y" in the scan started by the STRIGn trigger.
2. Requires EVCTRL0.STARTEI = 1 to enable any trigger event(s) from Event System (EVSYS).
3. If using the Event System (EVSYS) trigger for ADC, Asynchronous CHANNELn.PATH = 0x2 must be used to guarantee deterministic ADC sample/convert trigger timing.
4. These bits are Enabled Protected (Writes are ignored when CTRLA.ENABLE = 1 . Returns a bus error.)
5. If CTRLB.SWCNVEN=1, all of these register bits are ignored.

44.7.10. ADC Channel Configuration Registers 5

Name: CONFIG[0].CHNCFG5
Offset: 0x34
Reset: 0x00000000
Property: PAC Write-Protected, Enable-Protected

Table 44-15. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	TRGSRC15[3:0]				TRGSRC14[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
	TRGSRC13[3:0]				TRGSRC12[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
	TRGSRC11[3:0]				TRGSRC10[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
	TRGSRC9[3:0]				TRGSRC8[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	1	1	0	0	1

Bits 0:3, 4:7, 8:11, 12:15, 16:19, 20:23, 24:27, 28:31 – TRGSRCk Conversion Trigger Source for channel k
Bits 15-12: TRGSRC11 ADC Sample/Conversion Trigger Source for analog input channel AIN11 [(1)]
Bits 11-8: TRGSRC10 ADC Sample/Conversion Trigger Source for analog input channel AIN10 [(1)]
Bits 7-4: TRGSRC9 ADC Sample/Conversion Trigger Source for analog input channel AIN9
Bits 3-0: TRGSRC8 ADC Sample/Conversion Trigger Source for analog input channel AIN8

TRGSRCn	Description (Trigger events start ADC sample/conversion sequence)
0000	No Trigger
0001	ADC Global Software Trigger. (Requires CTRLB.GSWTRG =1, CTRLB.GSWTRG is self-clearing after trigger event.)
0010	ADC Global Level Software Trigger (Requires CTRLB.LSWTRG = 1)
0011	ADC SCANTRG - Scan Trigger. (Requires CHNCFG20.CSSy be configured accordingly)
0100	ADC STRIG Synchronous Trigger
0101-1111	ADC Trigger Event(s) from Event System (EVSYS)

Note:
1. This bit is not available in the 64-pin package.

If CHNCFG5.TRGSRCy = 0b101 thru 0b1111:

Event System (EVSYS)			ADC
CHANNELx.EVGEN	CHANNELn.PATH(2)	USERm.CHANNEL	CHNCFG5.TRGSRCy
User Selected ADC Trigger Event Source	= 0x2 Asynchronous path	78	0b0101
		79	0b0110
		80	0b0111
		81	0b1000
		82	0b1001
		83	0b1010
		84	0b1011
		85	0b1100
		86	0b1101
		87	0b1110
		88	0b1111

Notes:

1. SCANTRG in turn requires programming of CORCTRL0.STRGSRC to select its trigger source. Also, the appropriate CHNCFG20.CSSy bit must be set to include channel "y" in the scan started by the STRIGn trigger.
2. Requires EVCTRL0.STARTEI = 1 to enable any trigger events from Event System (EVSYS).
3. If using Event System (EVSYS) trigger for ADC, Asynchronous CHANNELn.PATH = 0x2 must be used to guarantee deterministic ADC sample/convert trigger timing.
4. These bits are Enabled Protected (Writes are ignored when CTRLA.ENABLE = 1 returns a bus error).
5. If CTRLB.SWCNVEN = 1, all of these register bits are ignored.

44.7.11. ADCn Module CALIBRATION Values Register

Name: CONFIG[0].CALCTRL
Offset: 0x38
Reset: 0x00000000
Property: PAC Write-Protected, Enable-Protected

Table 44-16. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	CALBITS[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
	CALBITS[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
	CALBITS[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
	CALBITS[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 – CALBITS[31:0] Calibration Data to the ADCn

Where n=0,1,2,3:

These register bits can only change when the applicable CTRLD.ANLENn bit is 0. Once the applicable CTRLD.ANLENn bit rising edge occurs, those bit values must remain unchanged until after the same CTRLD.ANLENn bit falling edge occurs.

This register must be initialized in user software to the factory-provided values in the CAL OTP Register FCCFG65 before setting CTRLD.ANLENn to 1.

Note:

1. This bit is Enabled Protected (Writes are ignored when CTRLA.ENABLE = 1. Returns a bus error).

44.7.12. ADC Event Control

Name: EVCTRL0
Offset: 0x3C
Reset: 0x00000000
Property: PAC Write-Protected, Enable-Protected

Table 44-17. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			CMPEO	RESRDYEO	STARTINV			STARTEI
Access			R/W	R/W	R/W			R/W
Reset			0	0	0			0

Bit 5 – CMPEO Window Event Out Enable

This bit indicates whether the Digital Comparator Window event output is enabled or not and whether an output event will be generated when the Digital Comparator Window is met.

Note:

1. This bit is Enabled Protected. (Writes are ignored when CTRLA.ENABLE = 1. Returns a bus error).

Value	Description
0	Window event output is disabled, and an event will not be generated.
1	Window event output is enabled, and an event will be generated.

Bit 4 – RESRDYEO Result Ready Event Out Enable

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.

Note:

1. This bit is Enabled Protected. (Writes are ignored when CTRLA.ENABLE = 1. Returns a bus error).

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 Sample/Conversion Event/Trigger Invert Enable

Notes:

1. Only rising edge EVSYS ADC Trigger events are acknowledged.
2. This bit is Enabled Protected. (Writes are ignored when CTRLA.ENABLE = 1. Returns a bus error).

Value	Description
0	Start event input source is not inverted.
1	Start event input source is inverted.

Bit 0 – STARTEI Start Conversion Event/Trigger Enable

Note:

1. This bit is Enabled Protected. (Writes are ignored when CTRLA.ENABLE = 1. Returns a bus error.)

Value	Description
0	Not enabled. ADC Event System events cannot trigger start of conversions.
1	Enabled. ADC Event System events can trigger start of conversions.

44.7.13. ADC Digital Comparator Control Register

Name: CMPCTRL
Offset: 0xB0
Reset: 0x00000000
Property: PAC Write-Protected, Enable-Protected

Table 44-18. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
			IEHIHI	IEHILO	ADCMPHI[11:8]			
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
	ADCMPHI[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
	IEBTWN	IELOHI	IELOLO	CMPEN	ADCMPLO[11:8]			
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
	ADCMPLO[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 29 – IEHIHI Enable Comparison - High Limit, Active High

Setting this bit enables comparison events ADCMPHI≤ ADC value.

Notes:

1. This bit is Enabled Protected : (Writes are ignored when CTRLA.ENABLE = 1) returns a bus error.
2. The Digital Comparator works on the final value of the filter data.
3. When using channel using FRACT (16bits left justified) the lower 4 bits are ‘0, therefor in this mode 15:4 will be in 11:0 and user must account for the lower 4 bits of 0’s.
4. In any case where the results value is of greater resolution than 12 bits, the comparison is only performed on upper 12 bits of the results value according to the settings of register CMPCTRLn.

Bit 28 – IEHILO Enable Comparison - High Limit, Active Low

Setting this bit enables comparison events ADC value < ADCMPHI.

Notes:
1. This bit is Enabled Protected : (Writes are ignored when CTRLA.ENABLE = 1) returns a bus error.
2. The Digital Comparator works on the final value of the filter data.
3. When using channel using FRACT (16bits left justified) the lower 4 bits are '0, therefor in this mode 15:4 will be in 11:0 and user must account for the lower 4 bits of 0's.
4. In any case where the results value is of greater resolution than 12 bits, the comparison is only performed on upper 12 bits of the results value according to the settings of register CMPCTRLn.

Bits 27:16 – ADCMPHI[11:0] High limit of Digital Analog Comparator n

This register stores the limit value which is used for comparisons with the ADC Module output data when
IEHIHI = 1, IEHILO = 1, or IEBTWN = 1.
The user is responsible for formatting the data in ADCMPHI[11:0] as signed or unsigned to match the data format as specified by the CHNCFG3n.SIGNk and CHNCFG2n.FRACTk bits for all the analog input channels k which are enabled by CHNCFG1n.CHNCMPENk .
Note: In Filter accumulation mode, the comparison is done on the upper 12 of the 16 bits of filter data.

Notes:
1. This bit is Enabled Protected : (Writes are ignored when CTRLA.ENABLE = 1) returns a bus error.
2. The Digital Comparator works on the final value of the filter data.
3. When using channel using FRACT (16bits left justified) the lower 4 bits are '0, therefor in this mode 15:4 will be in 11:0 and user must account for the lower 4 bits of 0's.
4. In any case where the results value is of greater resolution than 12 bits, the comparison is only performed on upper 12 bits of the results value according to the settings of register CMPCTRL*n.*

Bit 15 – IEBTWN Enable Comparison - Active Between Limits

Setting this bit enables comparison events ADCMPLO ≤ ADC Value < ADCMPHI.
Notes:
1. This bit is Enabled Protected : (Writes are ignored when CTRLA.ENABLE = 1) returns a bus error.
2. The Digital Comparator works on the final value of the filter data.
3. When using channel using FRACT (16bits left justified) the lower 4 bits are '0, therefor in this mode 15:4 will be in 11:0 and user must account for the lower 4 bits of 0's.
4. In any case where the results value is of greater resolution than 12 bits, the comparison is only performed on upper 12 bits of the results value according to the settings of register CMPCTRL*n.*

Bit 14 – IELOHI Enable Comparison - Low Limit, Active High

Setting this bit enables comparison events ADCMPLO≤ ADC Value.
Notes:
1. This bit is Enabled Protected : (Writes are ignored when CTRLA.ENABLE = 1) returns a bus error.
2. The Digital Comparator works on the final value of the filter data.
3. When using channel using FRACT (16bits left justified) the lower 4 bits are '0, therefor in this mode 15:4 will be in 11:0 and user must account for the lower 4 bits of 0's.
4. In any case where the results value is of greater resolution than 12 bits, the comparison is only performed on upper 12 bits of the results value according to the settings of register CMPCTRL*n.*

Value	Description
0	Use normal interrupts
1	use early interrupts

Bit 13 – IELOLO Enable Comparison - Low Limit, Active Low

Setting this bit enables comparison events ADC Value < ADCMPLO.

Notes:

1. This bit is Enabled Protected : (Writes are ignored when CTRLA.ENABLE = 1) returns a bus error.
2. The Digital Comparator works on the final value of the filter data.
3. When using channel using FRACT (16bits left justified) the lower 4 bits are ‘0, therefor in this mode 15:4 will be in 11:0 and user must account for the lower 4 bits of 0’s.
4. In any case where the results value is of greater resolution than 12 bits, the comparison is only performed on upper 12 bits of the results value according to the settings of register CMPCTRLn.

Bit 12 – CMPEN Digital Comparator n Enable

Setting this bit enables digital comparisons for the inputs to ADCn.
For each channel input channel k to ADCn to be compared the corresponding bit CHNCFG1n.CHNCMPENk must be set for the channel to be monitored.

Notes:

1. This bit is Enabled Protected : (Writes are ignored when CTRLA.ENABLE = 1) returns a bus error.
2. The Digital Comparator works on the final value of the filter data.
3. When using channel using FRACT (16bits left justified) the lower 4 bits are ‘0, therefor in this mode 15:4 will be in 11:0 and user must account for the lower 4 bits of 0’s.
4. In any case where the results value is of greater resolution than 12 bits, the comparison is only performed on upper 12 bits of the results value according to the settings of register CMPCTRL*n.*

Bits 11:0 – ADCMPLO[11:0] Low limit of Digital Analog Comparator

This register stores the limit value which is used for comparisons with the ADC Module output data when
IELOHI = 1, IELOLO = 1, or IEBTWN = 1.
The user is responsible for formatting the data in ADCMPLO[11:0] as signed or unsigned to match the data format as specified by the CHNCFG3n.SIGNk and CHNCFG2n.FRACTk bits for all the analog input channels k which are enabled by CHNCFG1n.CHNCMPENk .
Note: In Filter accumulation mode, the comparison is done on the upper 12 of the 16 bits of filter data.

Notes:

1. This bit is Enabled Protected : (Writes are ignored when CTRLA.ENABLE = 1) returns a bus error.
2. The Digital Comparator works on the final value of the filter data.
3. When using channel using FRACT (16bits left justified) the lower 4 bits are ‘0, therefor in this mode 15:4 will be in 11:0 and user must account for the lower 4 bits of 0’s.
4. In any case where the results value is of greater resolution than 12 bits, the comparison is only performed on upper 12 bits of the results value according to the settings of register CMPCTRL*n.*

44.7.14. ADC Digital Filter Control Register

Name: FLTCTRL0
Offset: 0xC0
Reset: 0x00000000
Property: PAC Write-Protected, Enable-Protected

Table 44-19. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
			FLTCHNID[3:0]					FLTEN
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
				DATA16EN	FMODE	OVRSAM[2:0]		
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bits 13:10 – FLTCHNID[3:0] ADC Channel ID To Be Filtered

Identifies which input channel, k , is to be filtered by the Digital Filter.

FLTCTRL0:

FLTCHNID[3:0] = Defines which single ADC Analog input channel VINP[14:0] to be filtered.
VINP[11:10] not available on 64 pin package.

Note:

1. This bit is Enabled Protected. (Writes are ignored when CTRLA.ENABLE = 1) returns a bus error.

Bit 8 – FLTEN Digital Filter Enable

When set, this bit enables the Digital Filter to filter the output data generated by the ADC. The input channel to be filtered is determined by FLTCHNID.

Note:

1. This bit is Enabled Protected: (Writes are ignored when CTRLA.ENABLE = 1) returns a bus error.

Bit 4 – DATA16EN Data 16 Bits Enable

Note:

1. This bit is Enabled Protected: (Writes are ignored when CTRLA.ENABLE = 1) returns a bus error.

This bit is significant only if FMODE = 1 (Averaging Mode) and CHNCFG2n.FRACTk= 1 (Fractional Output Mode, where k = FLTCHNID[3:0] is the chosen input for filtering) as follows:

Value	Description
0	Only the first 12 bits are significant, followed by 4 zeros.
1	All 16 bits of the filter output data are significant

Bit 3 – FMODE ADC Filter Mode

Note:

1. This bit is Enabled Protected: (Writes are ignored when CTRLA.ENABLE = 1) returns a bus error.

Value	Description
0	Filtering in Oversampling Mode (power-up default)
1	Filtering in Averaging Mode

Bits 2:0 – OVRSAM[2:0] Oversampling Ratio

Note:

1. This bit is Enabled Protected: (Writes are ignored when CTRLA.ENABLE = 1) returns a bus error.

Determines the number of samples generated in the burst mode used for computing one single filter output value.
The OVRSAM encoding depends on the FMODE setting as follows:
If FMODE = 0 (Oversampling Mode) then OVRSAM is encoded as follows:

Value	Descripton
000	4 samples, shift sum 1-bit to right, output data is 13-bits
001	16 samples, shift sum 2-bits to right, output data is 14-bits
010	64 samples, shift sum 3-bits to right, output data is 15-bits
011	256 samples, shift sum 4-bits to right, output data is 16-bits
100	2 samples, shift sum 0-bits to right, output data is in 12.1 format
101	8 samples, shift sum 1-bit to right, output data is in 13.1 format
110	32 samples, shift sum 2-bits to right, output data is in 14.1 format
111	128 samples, shift sum 3-bits to right, output data is in 15.1 format

If FMODE=1 (Averaging Mode), then OVRSAM is encoded as follows:

Value	Descripton
000	2 samples to be averaged
001	4 samples to be averaged
010	8 samples to be averaged
011	16 samples to be averaged
100	32 samples to be averaged
101	64 samples to be averaged
110	128 samples to be averaged
111	256 samples to be averaged

44.7.15. ADC COR Channel Ready DATA ID Register

Name: CORCHDATAID
Offset: 0xD0
Reset: 0x00000000
Property: PAC Write-Protected, Enable-Protected

Table 44-20. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

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

Bits 3:0 – CHRDYID[3:0] ADC Channel Read ID
Input Channel Index k:
For ADC, set value to input channel index, k,0 ≤ k ≤ (Sn-1) for status register CHNRDYDAT to display the current values of configuration bits and the last converted output data or written by the user to display the channel.
Notes:
1. ADC supports external analog inputs VINP[11:0] and internal inputs VINP[14:12]. VINP[11:10] are not available in the 64-pin package.
2. Selecting unimplemented input channels on a given ADCn will return a bus error with the data (32'h00000000).

Value	Description
`1110`	analog input channel 14(1)
`...`	...
`0010`	analog input channel 2
`0001`	analog channel 1
`0000`	analog channel 0

44.7.16. ADC Channel Ready DATA Register

Name: CHRDYDAT
Offset: 0xD4
Reset: 0x00000000

Note: All the data read in this register pertains to the Module defined by CORCHDATAID.CORDYID and its analog input channel defined by CORCHDATAID.CHRDYID.

Table 44-21. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
					FRACT	SIGN	DIFF	LVL
Access					R	R	R	R
Reset					0	0	0	0

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	CHRDYDAT[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
	CHRDYDAT[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 27 – FRACT Current FRACT setting for the analog input channel

Bit 26 – SIGN Current SIGN setting for the analog input channel

Bit 25 – DIFF Current DIFF setting for the analog input channel

Bit 24 – LVL Current LVL setting for the analog input channel

Bits 15:0 – CHRDYDAT[15:0] ADC Channel Output Data [15:0] for the analog input channel
Note: A read of CHRDYDAT will generate a read bus error on analog input channels which have not been implemented on ADCn, and the returned data will be 0x0000_0000.

44.7.17. ADC FIFO Output Data Register

Name: PFFDATA
Offset: 0xD8
Reset: 0x00000000

Note: Reading any part of the PFFDATA register advances the FIFO pointer, therefore it is recommended to read all 32 bits of the register and then parse out the values of each field.

Table 44-22. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	PFFCNT[7:0]							
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
	PFFFRACT	PFFSIGN			PFFCHNID[3:0]			
Access	R	R			R	R	R	R
Reset	0	0			0	0	0	0

Bit	15	14	13	12	11	10	9	8
	PFFDATA[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
	PFFDATA[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:24 – PFFCNT[7:0] Current number of data entries to be read in the APB FIFO

Bit 23 – PFFFRACT CHNCFG2.FRACTk setting associated with data in the PFFDATA register

Bit 22 – PFFSIGN CHNCFG3.SIGNk setting associated with data in the PFFDATA register

Bits 19:16 – PFFCHNID[3:0] Analog Input Channel Index k associated with PFFCORID and data in the PFFDATA register, k = 14-0 which represents VINP[14:0]

Bits 15:0 – PFFDATA[15:0] 16-bit Output Data of the FIFO
Data is in the format given by PFFFRACT and PFFSIGN.

44.7.18. ADC FIFO Control Register

Name: PFFCTRL
Offset: 0xE4
Reset: 0x00000000

The ADC FIFO is useful in applications that stream out ADC data at very high transfer rates to relive CPU bandwidth. Individual high data rate ADC result interrupts and CPU reads may slow bus access transfer requests.

Table 44-23. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
								PFFRDYDMA
Access								R/W
Reset								0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

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

Bit 16 – PFFRDYDMA DMA FIFO Data Ready Interrupt selection

Note:
1. This bit is Enabled Protected: (Writes are ignored when CTRLA.ENABLE = 1. Returns a bus error).

Value	Description
0	Selects CTLINTFLAG.PFFHFUL for the ADC DMA PFFRDY trigger signal to the DMAC
1	Selects CTLINTFLAG.PFFRDY for the ADC DMA PFFRDY trigger signal to the DMAC

Bit 4 – PFFCR FIFO Enable for ADCn

When PFFEN = 1, setting this bit for the ADC enables the conversion output data of any channel k associated to the ADC to be stored into the optional data FIFO.

Note:
1. This bit is Enabled Protected: (Writes are ignored when CTRLA.ENABLE = 1. Returns a bus error).

Bit 1 – PFFEN FIFO General Enable

When the FIFO is disabled no data is being saved into the FIFO and the its logic is being kept in reset state.

Note:

1. This bit is Enabled Protected: (Writes are ignored when CTRLA.ENABLE = 1. Returns a bus error).

Value	Description
0	FIFO is disabled
1	FIFO is enabled

44.7.19. ADC Core Synchronization Register

Name: SYNCBUSY
Offset: 0xE8
Reset: 0x00000000
Property: -

Notes: The following GCLK-clocked registers are “Enable Write Protected”:

- CTRLC
- CTRLD
- CORCTRL0
- CHNCFG10
- CHNCFG20
- CHNCFG30
- CHNCFG40
- CHNCFG50
- CALTRL0
- FLTCTRL0

Therefore, they do not require a SyncBusy bit. These registers are write-disabled when the CTRLA.ENABLE bit is set to enable the ADC. They can only be changed when the ADC is disabled (CTRLA.ENABLE = 0). The user must completely configure the ADC and then enable the ADC by setting CTRLA.ENABLE = 1.

Table 44-24. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						CTRLB	ENABLE	SWRST
Access						R	R	R
Reset						0	0	0

Bit 2 – CTRLB Synchronization Busy bit
For the GCLK-based register CTRLB:
When Hardware sets this bit, no writes are permitted to the CTRLB register.

Bit 1 – ENABLE ENABLE Synchronization Busy bit
For GCLK-based register bit CTRLA.ENABLE:
When Hardware sets this bit, no writes are permitted to the CTRLA.ENABLE register.

Bit 0 – SWRST Software Reset Busy bit
Notes:
1. Typically, when the SWRST is written, the bit is auto-cleared the next APB clock cycle after. However, the SYNCBUSY.SWRST bit is set and stays set until the reset in the GCLK domain is completed. Therefore, the user must poll the SYNCBUSY register to know when the operation is complete.
2. Care must be taken during the APB reset phase, because potentially the external clock (GCLK) may not present.
3. During a SWRST, access to registers or bits without SWRST are disallowed until SYNCBUSY.SWRST is cleared by hardware.

44.7.20. ADC Control Interrupt Enable Clear Register

Name: CTLINTENCLR
Offset: 0x100
Reset: 0x00000000
Property: PAC Write-Protection

Note: Writing a zero to these bits has no effect. Writing a one to these any of these bits will CLEAR the ENABLE bit.

Note: A read of this register provides whether the Interrupt is Enabled (bit=1) or Disabled (bit=0), i.e. a write of a 1 to the bit then a read of the bit will return the interrupt is disabled (=0).

Table 44-25. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
					PFFHFUL	PFFRDY	PFFOVF	PFFUNF
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	VREFRDY	VREFUPD						CRRDY
Access	R/W	R/W						R/W
Reset	0	0						0

Bit 11 – PFFHFUL ADC FIFO Half Full Disable
Writing a 1 to this bit will disable the ADC FIFO Half Full as an interrupt request.
Reading this bit returns whether the PFFHFUL interrupt is enabled (1 = enabled).

Bit 10 – PFFRDY ADC FIFO Data Ready Disable
Writing a 1 to this bit will disable the ADC FIFO Data Ready as an interrupt request.
Reading this bit returns whether the PFFRDY interrupt is enabled (1 = enabled).

Bit 9 – PFFOVF ADC FIFO Write Overflow Error Disable
Writing a 1 to this bit will disable the ADC FIFO Overflow Error as an interrupt request,
Reading this bit returns whether the PFFOVF interrupt is enabled (1 = enabled).

Bit 8 – PFFUNF ADC FIFO Read Underflow Error Disable
Writing a 1 to this bit will disable the ADC FIFO Read Underflow Error as an interrupt request.
Reading this bit returns whether the PFFUNF interrupt is enabled (1 = enabled).

Bit 7 – VREFRDY Voltage Reference Ready Interrupt Disable
Writing a 1 to this bit will disable the ADC Voltage Reference Ready as an interrupt request.
Reading this bit returns whether the VREFRDY interrupt is enabled (1 = enabled).

Bit 6 – VREFUPD Voltage Reference Ready Updated Interrupt Disable
Writing a 1 to this bit will disable the ADC Voltage Reference Ready Updated as an interrupt request.
Reading this bit returns whether the VREFUPD interrupt is enabled (1 = enabled).

Bit 0 – CRRDY Core n Ready Interrupt Disable
Writing a 1 to this bit will disable the Core n Ready as an interrupt request.
Reading this bit returns whether the CRRDYn interrupt is enabled (1 = enabled).

44.7.21. ADC Control Interrupt Enable Set Register

Name: CTLINTENSET
Offset: 0xFC
Reset: 0x00000000
Property: PAC Write-Protection

Note: Writing a zero to these bits has no effect. Writing a one to these any of these bits will SET the ENABLE bit.

Note: A read of this register provides whether the Interrupt is Enabled (bit=1) or Disabled (bit=0), i.e. a write of a 1 to the bit then a read of the bit will return the interrupt is enabled (=0).

Table 44-26. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
					PFFHFUL	PFFRDY	PFFOVF	PFFUNF
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	VREFRDY	VREFUPD			CRRDY3	CRRDY2	CRRDY1	CRRDY0
Access	R/W	R/W			R/W	R/W	R/W	R/W
Reset	0	0			0	0	0	0

Bit 11 – PFFHFUL ADC FIFO Half Full Interrupt Enable
Writing a 1 to this bit will enable the ADC FIFO Half Full interrupt request.
Reading this bit returns whether the PFFHFUL interrupt is enabled (1 = enabled).

Bit 10 – PFFRDY ADC FIFO Data Ready Interrupt Enable
Writing a 1 to this bit will enable the ADC FIFO Data Ready interrupt request.
Reading this bit returns whether the PFFRDY interrupt is enabled (1 = enabled).

Bit 9 – PFFOVF ADC FIFO Write Overflow Error Interrupt Enable
Writing a 1 to this bit will enable the ADC FIFO Overflow Error interrupt request,
Reading this bit returns whether the PFFOVF interrupt is enabled (1 = enabled).

Bit 8 – PFFUNF ADC FIFO Read Underflow Error Interrupt Enable
Writing a 1 to this bit will enable the ADC FIFO Read Underflow Error interrupt request.
Reading this bit returns whether the PFFUNF interrupt is enabled (1 = enabled).

Bit 7 – VREFRDY Voltage Reference Ready Interrupt Enable
Writing a 1 to this bit will Enable the ADC Voltage Reference Ready as an interrupt request.
Reading this bit returns whether the VREFRDY interrupt is enabled (1 = enabled).

Bit 6 – VREFUPD Voltage Reference Ready Updated Interrupt Enable
Writing a 1 to this bit will Enable the ADC Voltage Reference Ready Updated as an interrupt request.
Reading this bit returns whether the VREFUPD interrupt is enabled (1 = enabled).

Bits 0, 1, 2, 3 – CRRDYn Core n Ready Interrupt Enable
Writing a 1 to this bit will enable the Core n Ready as an interrupt request.
Reading this bit returns whether the CRRDYn interrupt is enabled (1 = enabled).

44.7.22. ADC Control Interrupt Flags Register

Name: CTLINTFLAG
Offset: 0x104
Reset: 0x00000000
Property: -

Note: Interrupt flags must be cleared and then read back to confirm the clear before exiting the ISR to avoid double interrupts.

Table 44-27. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
					PFFHFUL	PFFRDY	PFFOVF	PFFUNF
Access					R/HS/HC	R/HS/HC	R/W/HS	R/W/HS
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	VREFRDY	VREFUPD						CRRDY
Access	R/W/HS/HC	R/W/HS						R/HS/HC
Reset	0	0						0

Bit 11 – PFFHFUL ADC FIFO Half Full Interrupt Flag
This bit is set when the FIFO is at least half full of data to be read. When set, this bit enables this condition to trigger the ADC interrupt if the corresponding bit in CTLINTSET is set. This bit is cleared by hardware when the FIFO output data has been read and there is less than a half full FIFO left to be read.
This bit is NOT reset by software writing a 1 to it, it is only reset by hardware.

Bit 10 – PFFRDY ADC FIFO Data Ready Interrupt Flag
This bit is set when the FIFO has data to be read. When set, this bit enables the trigger of the ADC interrupt if the corresponding bit in CTLINTSET is set.
This bit is cleared by hardware when the FIFO output data in has been read and there is no additional data ready in the FIFO (that is the APB FIFO is empty). This bit is NOT reset by software writing a 1 to it.

Bit 9 – PFFOVF Write Overflow Error in the FIFO
This bit is set by hardware when the FIFO is full and new output data overwrites data to be read. When set, this bit enables this condition to trigger an ADC interrupt if the corresponding bit in CTLINTSET is set.

This bit is reset by software writing a 1 to it.

Bit 8 – PFFUNF Read Underflow Error in the FIFO
This bit is set by hardware when the FIFO is empty. When set, this bit enables the trigger of an ADC interrupt if the corresponding bit in CTLINTSET is set.
This bit is reset by software writing a 1 to it.

Bit 7 – VREFRDY Read-only ADC Voltage Reference Ready Status Bit
Hardware sets and clears this bit according to the status of the ADC Voltage Reference. ADC Conversion Data is valid only after VREFRDY is set by hardware, so the ADC interrupt service routine in charge of data processing should always check first that VREFRDY is set to ensure the data validity. This bit will only be updated to the VREFRDY condition set or cleared when the CTRLA.ENABLE is on. Therefore, once the ISR has verified that VREFRDY=1, it should disable the corresponding interrupt by setting CTLINTENCLR.VREFRDY = 1 to prevent continuous firing of the ISR. Alternately, the software setup routine for the ADC can wait in a while(1) loop, polling VREFRDY until it goes high.

Bit 6 – VREFUPD Voltage Reference Ready Update Interrupt Flag
This bit is set by hardware on both the positive and negative edges of the bit VREFRDY. This means that the hardware will set this bit when the ADC Voltage Reference is ready, but also when it fails, that is on any change. When set, this bit enables the trigger of an ADC interrupt if the corresponding bit in CTLINTSET is set.
This bit is reset by software writing a 1 to it. It is NOT cleared by a software read.
Software must read the value of VREFRDY to ascertain if the ADC analog reference circuits are in order or not when the CPU is servicing the interrupt prompted by VREFUPD. The ADC Voltage Reference analog signals are required to be ready during operation of the ADC. If an ADC Voltage Reference fault is detected, the ADC module must be re-calibrated. Most likely an ADC Reference Voltage fault is caused by a brown-out of the analog Vdd supply.
Hardware sets this bit to zero when CTRLA.ENABLE=0.

Bit 0 – CRRDY Read-only ADCn Ready Status Bit
It is set by hardware when $2^{WKUPEXP}$ ADCn clocks have elapsed after software setting CTRLD.ANLENn to one. It is cleared by hardware when CTRLD.ANLENn is de-asserted (but does NOT depend on CTRLD.CHNENn). When set, this bit enables the trigger of an ADC interrupt if the corresponding bit in CTLINTSET is set.
It is NOT cleared after a software read.
Note: Since this bit cannot be cleared except by setting CTRLD.ANLENn to zero, which would then disable the analog and bias circuitry for ADCn, the ISR that services CRRDYn firing should then disable the interrupt by setting CTLINTENCLR.CRRDYn = 1.

Note: These bits are only cleared when the Analog Channel is disabled.

44.7.23. ADCn Interrupt Enable Clear

Name: INTENCLR
Offset: 0x120
Reset: 0x00000000
Property: PAC Write-Protected, Enable-Protected

Note: Writing a zero to any of these bits has no effect, but writing a one to these bits will CLEAR the ENABLE bit.
A read of this register provides the current status of interrupts, that is whether each interrupt is enabled (bit = 1) or disabled (bit = 0).

Table 44-28. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	CHRDY[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
	CHRDY[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
					EOSRDY	CHNERRC	FLTRDY	CHRDYC
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

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

Bits 31:16 – CHRDY[15:0] ADCn Channel Ready Interrupt Disable for Input Channel k
Writing a 1 to bit will k disable the data ready flag for Core n channel k as an interrupt request.
CHRDY[k] = 14-0 individual channels which corresponds to VINP[14:0]. K = 11 and 10 do not exist for the 64-pin package.
Note:
1. This bit is Enabled Protected: (Writes are ignored when CTRLA.ENABLE = 1. Returns a bus error).

Bit 11 – EOSRDY ADCn Module End-Of-Scan Interrupt Disable
Writing a 1 to this bit will disable the flag bit EOSRDY as an interrupt request.
Note:
1. This bit is Enabled Protected: (Writes are ignored when CTRLA.ENABLE = 1. Returns a bus error).

Bit 10 – CHNERRC ADCn Module Channel Overwritten Error Flag Interrupt Disable
Writing a 1 to this bit will disable the flag bit CHNERRCas an interrupt request.

Note:
1. This bit is Enabled Protected: (Writes are ignored when CTRLA.ENABLE = 1. Returns a bus error.)

Bit 9 – FLTRDY ADCn Digital Filter Ready Flag Disable
Writing a 1 to this bit will disable the filter ready flag for Filter as an interrupt request.
Note:
1. This bit is Enabled Protected: (Writes are ignored when CTRLA.ENABLE = 1. Returns a bus error.)

Bit 8 – CHRDYC ADCn Module Current Channel ready Disable
Writing a 1 to this bit will disable the flag bit CHRDYC as an interrupt request.
Note:
1. This bit is Enabled Protected: (Writes are ignored when CTRLA.ENABLE = 1. Returns a bus error.)

Bit 7 – SOVFL ADCn Clock Synchronizer Overflow Disable
Writing a 1 to this bit will disable the flag SOVFL as an interrupt request.
Note:
1. This bit is Enabled Protected: (Writes are ignored when CTRLA.ENABLE = 1. Returns a bus error.)

Bit 4 – CMPHIT ADCn Digital Comparator Hit Disable
Writing a 1 to this bit will disable the comparator hit flag for Comparator n as an interrupt request.
Note:
1. This bit is Enabled Protected: (Writes are ignored when CTRLA.ENABLE = 1. Returns a bus error.)

44.7.24. ADCn Interrupt Enable Set Register

Name: INTENSET
Offset: 0x124
Reset: 0x00000000
Property: PAC Write-Protected, Enable-Protected

Note: Writing a zero to any of these bits has no effect, but writing a one to these bits will SET the ENABLE bit.
A read of this register provides the current status of interrupts, i.e., whether each interrupt is enabled (bit=1) or disabled (bit=0).

Table 44-29. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	CHRDY[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
	CHRDY[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
					EOSRDY	CHNERRC	FLTRDY	CHRDYC
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

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

Bits 31:16 – CHRDY[15:0] ADCn Channel Ready Interrupt Enable for Input Channel k of Core n
Writing a 1 to bit k will enable the data ready flag for Core n channel k as an interrupt request.
CHRDY[k] is defined only for k = 0,1,2,...,(Sn-1).
Note:
1. This bit is Enabled Protected : (Writes are ignored when CTRLA.ENABLE = 1. Returns a bus error.)

Bit 11 – EOSRDY ADCn Module End-Of-Scan Interrupt Enable
Writing a 1 to this bit will disable the flag bit EOSRDY as an interrupt request.
Note:
1. This bit is Enabled Protected : (Writes are ignored when CTRLA.ENABLE = 1. Returns a bus error.)

Bit 10 – CHNERRC ADCn Module Channel Overwritten Error Flag Interrupt Enable
Writing a 1 to this bit will enable the flag bit CHNERRC as an interrupt request.
Note:
1. This bit is Enabled Protected : (Writes are ignored when CTRLA.ENABLE = 1. Returns a bus error.)

Bit 9 – FLTRDY ADCn Digital Filter Ready Flag Enable

Writing a 1 to this bit will enable the filter ready flag for Filter as an interrupt request.

Note:

1. This bit is Enabled Protected: (Writes are ignored when CTRLA.ENABLE = 1. Returns a bus error.)

Bit 8 – CHRDYC ADCn Module Current Channel Ready Enable

Writing a 1 to this bit will enable the flag bit CHRDYC as an interrupt request.

Note:

1. This bit is Enabled Protected : (Writes are ignored when CTRLA.ENABLE = 1. Returns a bus error.)

Bit 7 – SOVFL ADCn Clock Synchronizer Overflow Enable

Writing a 1 to this bit will enable the flag SOVFL as an interrupt request.

Note:

1. This bit is Enabled Protected: (Writes are ignored when CTRLA.ENABLE = 1. Returns a bus error.)

Bit 4 – CMPHIT ADCn Digital Comparator Hit Enable

Writing a 1 to this bit will enable the comparator hit flag for Comparator n as an interrupt request.

Note:

1. This bit is Enabled Protected: (Writes are ignored when CTRLA.ENABLE = 1. Returns a bus error.)

44.7.25. ADC Interrupt Flags Register

Name: INTFLAG
Offset: 0x128
Reset: 0x00000000
Property: -

Note: Interrupt flags must be cleared and then read back to confirm they are cleared before exiting the ISR to avoid double interrupts.

Table 44-30. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	CHRDY[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
	CHRDY[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
	CRDYID[3:0]				EOSRDY	CHNERRC	FLTRDY	CHRDYC
Access	R/W	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
	SOVFL			CMPHIT	CMPINTID[3:0]			
Access	R/W			R/W	R/W	R/W	R/W	R/W
Reset	0			0	0	0	0	0

Bits 31:16 – CHRDY[15:0] ADC Channel Ready Interrupt Flag Channel k
CHRDY[k] = 1 indicates that ADC has completed its last A/D conversion for channel k.
Note: CHRDY[k] = 14-0 individual channels which corresponds to VINP[14:0]. K=11 and 10 do not exist for the 64-pin package.
If INTENSET.CHRDY[k] is set, then CHRDY[k] = 1 will trigger an ADC interrupt.

CHRDY[k] is reset by software writing a 1 to it.

Bits 15:12 – CRDYID[3:0] ADC Module Current Channel ID Ready Bits
The value of CRDYID indicates the input channel index, k , that ADC has just completed in its current scan. This if for information only since these bits cannot be used to trigger an ADC interrupt. (CHRDY[k] are intended for that purpose.)
These bits are reset by software writing a 1 to them.

Bit 11 – EOSRDY ADC Module End-Of-Scan Interrupt Flag
This bit is set by hardware at the end of the scan of all channels included in the scan performed by ADC in response to a SINGLE event of the Scan Trigger 0 (STRIG0).
If INTENSET.EOSRDY is set, then EOSRDY = 1 will trigger an ADC interrupt.

This bit is reset by software writing a 1 to it.

Bit 10 – CHNERRC ADC Module Channel Overwritten Error Flag

When set, this bit indicates that the ADC has completed its last A/D conversion for channel CRDYID[3:0], but at the time CRDYID[3:0] was updated, the status bit CHRDYC was still set, which indicates that the software may not have had the time to read the previous data, which may be now lost.
If INTENSET.CHNERRC is set, then CHNERRC = 1 will trigger an ADC interrupt.
This bit is reset by software writing a 1 to it.

Bit 9 – FLTRDY ADC Digital Filter Ready for Filter Flag

When set, this bit indicates that the digital filter has issued a new output sample for the input channel defined by FLTCTRL.FLTCHNID.
If INTENSET.FLTRDY is set, then FLTRDY = 1 will trigger an ADC interrupt.
This bit is reset by software writing a 1 to it.

Bit 8 – CHRDYC ADC Current Channel Ready Flag

0 = ADC busy or idle.
1 = When set, this bit signifies that the ADC has completed its current A/D conversion for the channel identified in CRDYID[3:0].

Notes:

1. If INTENSET.CHRDYC is set, then CHRDYC = 1 will trigger an ADC interrupt.
2. This bit is reset by software writing a 1 to it.

Bit 7 – SOVFL ADC Clock Synchronizer Overflow into the APB Clock Domain

When set this bit signifies the ADC data was lost due to a slow APB_CLK.
If INTENSET.SOVFL is set, then SOVFL = 1 will trigger an ADC interrupt.
This bit is reset by software writing a 1 to it.

Bit 4 – CMPHIT ADC Digital Comparator Hit Interrupt Flag for Comparator

When set, this bit signifies that the Digital Comparator associated with ADC has issued a condition hit interrupt for channel identified in CMPINTID[5:0].
If INTENSET.CMPHIT is set, then CMPHIT = 1 will trigger an ADC interrupt.
This bit is reset by software writing a 1 to it.

Bits 3:0 – CMPINTID[3:0] ADC Module Digital Comparator Channel ID Bits

When set, this signifies that the ADC Channel ID for which the digital comparator has issued the condition hit interrupt (CMPHIT=1). These bits cannot be used as an interrupt request flag. This is for information only. (CMPHIT is intended for that purpose.)
This bit is reset by software writing a 1 to it.

44.7.26. ADC Debug Control Register

Name: DBGCTRL
Offset: 0x168
Reset: 0x00000000

Table 44-31. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

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

Bit 0 – DBGRUN Debug Running State
This bit is not affected by software reset and should not be changed by software while the ADC is enabled.
Note: During debug operation with DBGRUN=0 the ADC captures input trigger events and on exit from the debug the ADC will (based on priority) perform conversions on the captured event or events.

Value	Description
0	The ADC is halted when the CPU is halted in debug mode.
1	The ADC continues normal operation when the CPU is halted in debug mode.

45. Analog Comparators (AC)

45.1. Overview

The Analog Comparator (AC) consists of two 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. The two comparators can combine to support a window function, discussed below.

Hysteresis is user selectable to achieve the optimal operation for each application.

By default, the input selection includes four shared analog port pins and several internal signals. The flexible internal architecture allows additional input selections. For details, refer to each chip-specific documentation. 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, called Comparator 0 (COMP0) and Comparator 1 (COMP1). The pair can be set in window mode to compare a signal to a voltage range instead of a single voltage level.

Important: If AVDD < 2.5v, users must enable analog charge pumps in SUPC.VREGCTRL.CPEN[1:0].

45.2. Features

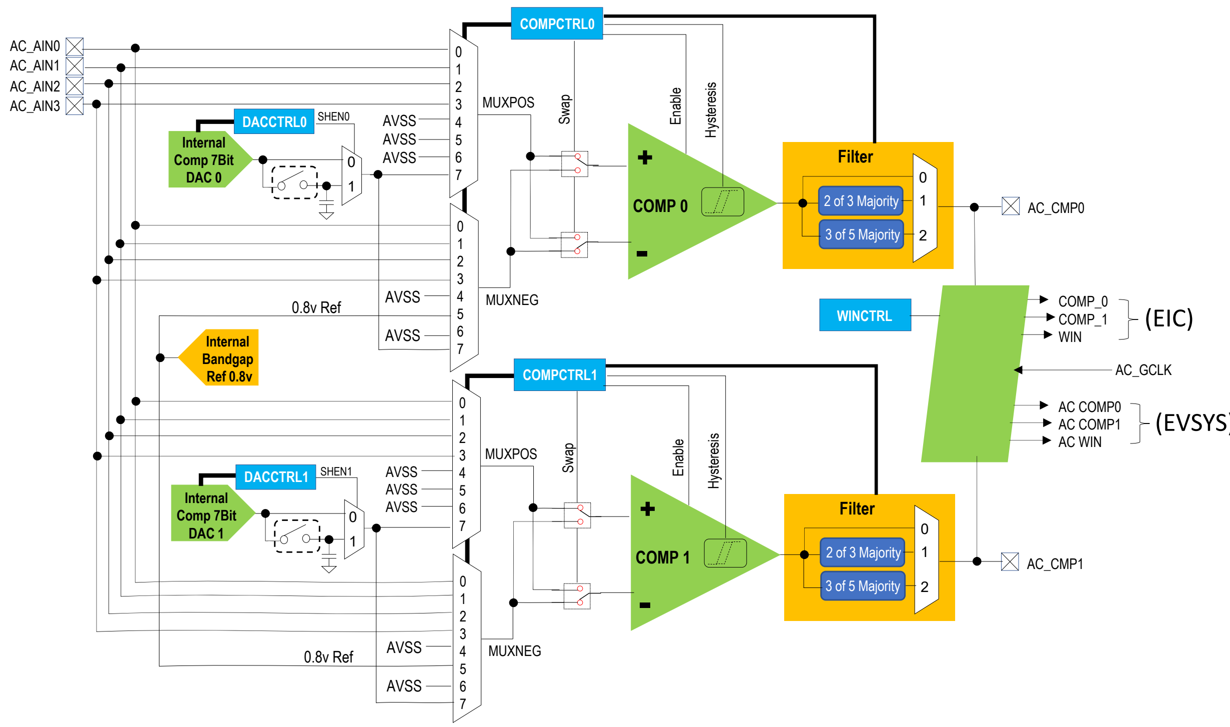
The following are key features of the AC module:

- Two individual comparators
- Selectable propagation delay versus current consumption
- Selectable hysteresis
  - 4-levels , Always ON
- Analog comparator outputs available on pins
  - Asynchronous or synchronous
- Flexible input selection:
  - Four pins selectable for positive or negative inputs
  - Ground (for zero crossing)
  - One dedicated 7-bit DAC for each comparator, enabling up to 128-level programmable VDD scaler
  - Extensions available to connect additional internal or external pins on each device
- 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
- Low-power option
  - Single-shot support

45.3. Block Diagram

Figure 45-1. Analog Comparator Block Diagram



45.4. Analog Connections

Each comparator has minimum four I/O pins that can be used as analog inputs. Each pair of comparators shares four pins. These pins must be configured for analog operation before using them as comparator inputs.

The Analog comparator embeds additional capabilities, enabling the option to connect more I/O pins from device to device, or other internal references. This selection is device specific, and for further details refer to the device specification.

Any internal reference source, such as a bandgap voltage reference or internal DAC must be configured and enabled prior to its use as a comparator input. Any external reference must be enabled and ready to be used, before the comparator is enabled.

45.5. Peripheral Dependencies

Peripheral Name	Base Address	NVIC IRQ Index: Source	MCLK AHBx/APBx Clock Enable Mask Bit	GCLK Peripheral Channel Clock Name:Register	PAC Peripheral Identifier (PAC.WRCTRL.PERIDx)	EVSYS Users (EVSYS.USERm)	EVSYS Generator (EVSYS.CHANNELn.EVGENx)	Power Domain
AC	0x4501_2000	190 : COMP_0, COMP_1, WIN	MCLK.CLKMSK3[9]	GCLK_AC: GCLK.PCHCTRL[34]	45	89 : SOC_0 90 : SOC_1	108 : AC COMP_0 109 : AC COMP_1 110 : WIN	VDDREG, AVDD

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. 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 Starting a Comparison for more details.
- Select the desired hysteresis with COMPCTRLn.HYSTEN and COMPCTRLn.HYST. See Input Hysteresis for more details.
- Write COMPCTRLn.SPEED to ‘1’
- Select the interrupt source with COMPCTRLn.INTSEL
- Select the positive and negative input sources with the COMPCTRLn.MUXPOS and COMPCTRLn.MUXNEG bits. See 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.

45.6.2.2. Starting a Comparison

Each comparator channel can be in one of two different measurement modes, determined by the Single bit in the Comparator (n) Control register (COMPCTRLn.SINGLE):

- Continuous measurement
- Single-shot

After being enabled, a start-up delay (COMPCTRLn.SUT) 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. For details on how the start-up time is calculated, refer to COMPCTRLn.SUT description.

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.

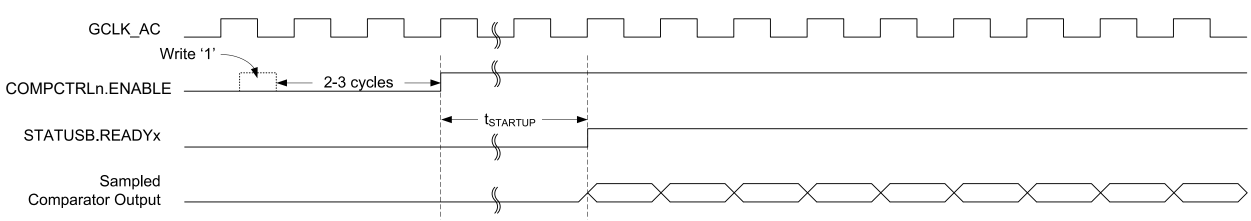
45.6.2.2.1. Continuous Measurement

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 n 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 45-2. 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.

45.6.2.2.2. Single-Shot

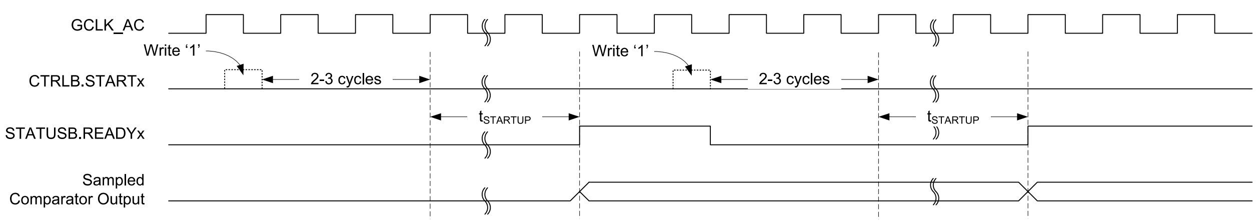
Single-shot operation is selected by writing COMPCTRLn.SINGLE to '1'. During single-shot operation, the comparator is normally idle. The user starts a single comparison by writing '1' to the respective Start Comparison bit in the write-only Control B register (CTRLB.STARTx). The comparator is enabled, 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 n 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 n Event Input bit in the Event Control Register (EVCTRL.COMPEIx) enables triggering on incoming peripheral events. Each comparator can be triggered independently by separate events. Event-triggered operation is similar to user-triggered operation; the difference is that a peripheral event from another hardware module causes the hardware to automatically start the comparison and will not clear STATUSB.READYx.

To detect an edge of the comparator output in single-shot operation for the purpose of interrupts, the result of the current measurement is compared with the result of the previous measurement (one sampling period earlier). An example of single-shot operation is shown in the following figure.

Figure 45-3. 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

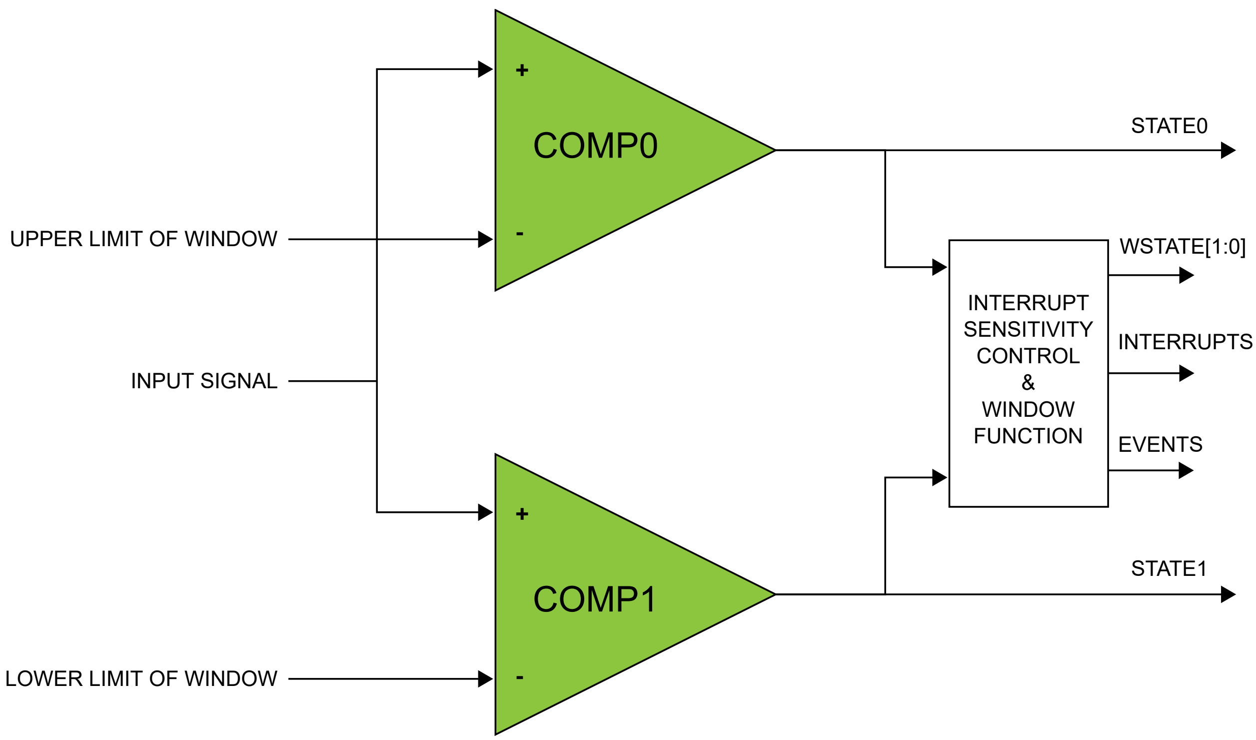
Each comparator pair can be configured to work together in Window mode. In this mode, a voltage range is defined, and the comparators give information about whether an input signal is within this range or not. Window mode is enabled by the Window Enable x bit in the Window Control register (WINCTRL.WENx). 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 the following figure, 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 45-4. Comparators in Window Mode



45.6.5. Internal DAC Operation

Each Analog Comparator pair, includes two DACs, each connected to its respective ACx.

The DACx is enabled when the analog comparator is enabled (COMPCTRLn = 1), and the DAC is used as positive or negative input for the respective comparator (MUXNEG = INTDAC or MUXPOS = INTDAC).

The DAC is disabled when the analog comparator is disabled (COMPCTRLn.ENABLE = 0).

The DAC configuration registers are cleared when the analog comparator is reset (CTRLA.SWRST = 1).

The DAC generates a reference voltage that is a fraction of the device's supply voltage, with 128 levels. One independent DAC channel is dedicated for each comparator. The voltage of each channel is selected by the Value x bit field in the DACCTRL registers (DACCTRL.VALUEx).

The DAC can be enabled in continuous operating mode (DACCTRL.SHENx = 0), or in sample mode (DACCTRL.SHENx = 1). When set in continuous mode, no clocks are required for operation. In sampling mode, each DAC controller includes a dedicated counter, generating the sampling clock.

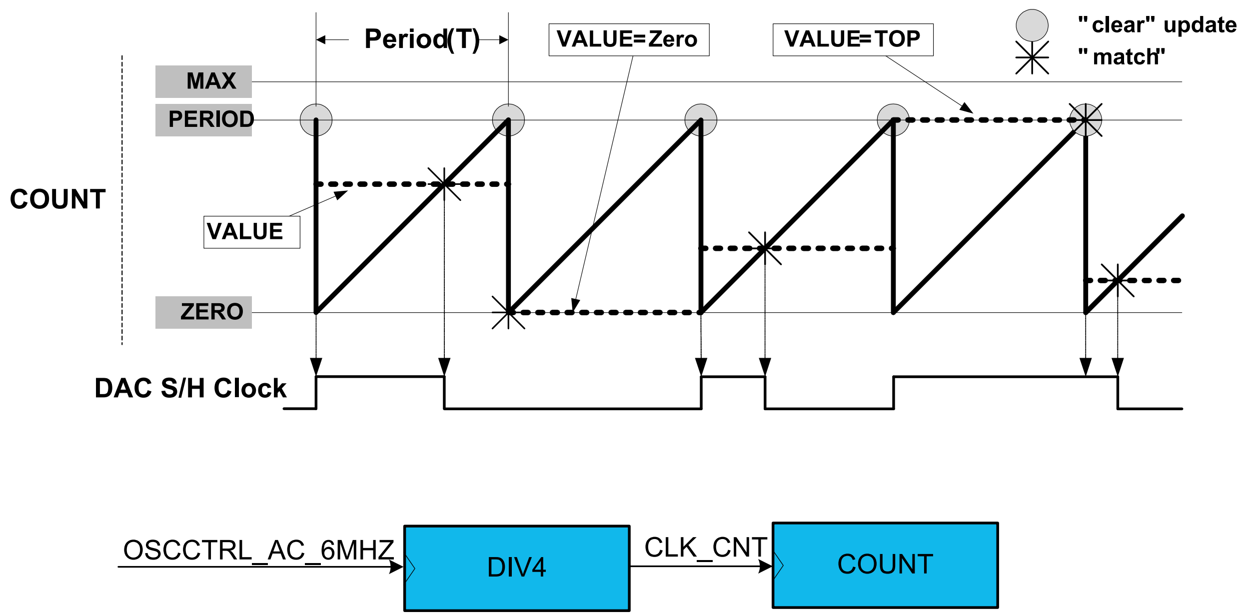
The counter operation is started when the analog comparator is ready (STATUSB.READYx = 1), and stopped when the analog comparator is disabled. As a consequence, in single shot mode, the DAC is enabled in continuous mode when the AC is enabled (COMPCTRLn.ENABLE = 1), and switches to the sample and hold operation when the comparator is ready.

Note: the DAC S/H operating mode and AC single-shot mode are two independent operations, and not linked together. Both S/H and continuous DAC operating modes can be selected when the AC is used either continuous or single-shot mode. The operating modes must be selected by the application, depending on power consumption and/or response time requirements.

The counter counts up and restarted when software PERIOD programmable value (CTRLC.PER) is reached. The counter operation is stopped and the internal counter is cleared when the sampling and hold operation is not required. When the AC output toggles, the counter is restarted.

The sampling clock is set when the counter operation starts, and cleared when the counter reaches the WIDTH software programmable value (CTRLC.WIDTH).

Figure 45-5. Sample and Hold Clock Generation



45.6.6. Input Hysteresis

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 always enabled for each comparator. The level of hysteresis is programmable through the Hysteresis Level bits also in the Comparator n Control register (COMPCTRLn.HYST).

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 n 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 Continuous Mode Filtering. For Single-shot mode, the comparison completes after the Nth filter sample, as shown in Single-Shot Filtering.

Figure 45-6. Continuous Mode Filtering

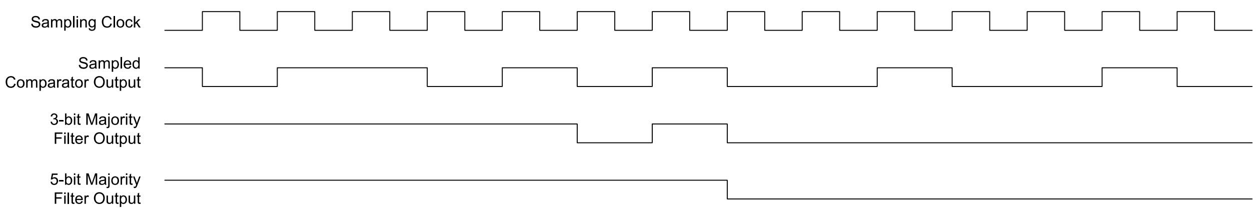
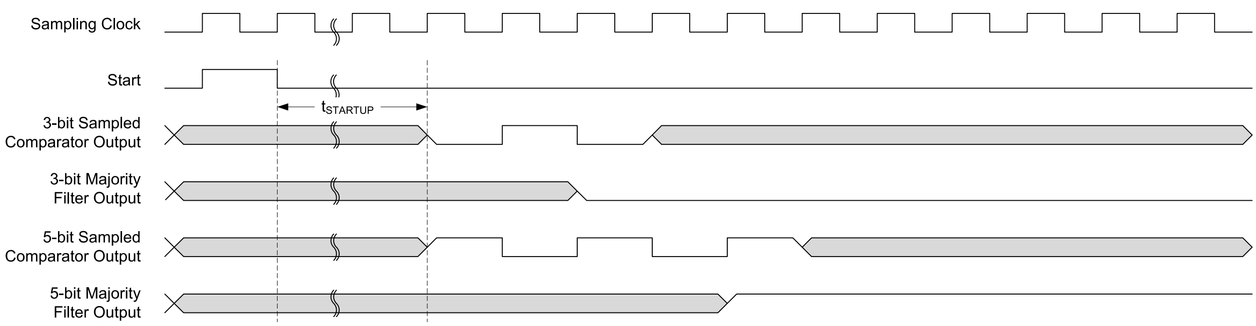


Figure 45-7. Single-Shot Filtering



During Sleep modes, filtering is supported only for single-shot measurements. Filtering must be disabled if continuous measurements will be done during Sleep modes, or the resulting interrupt/ event may be generated incorrectly.

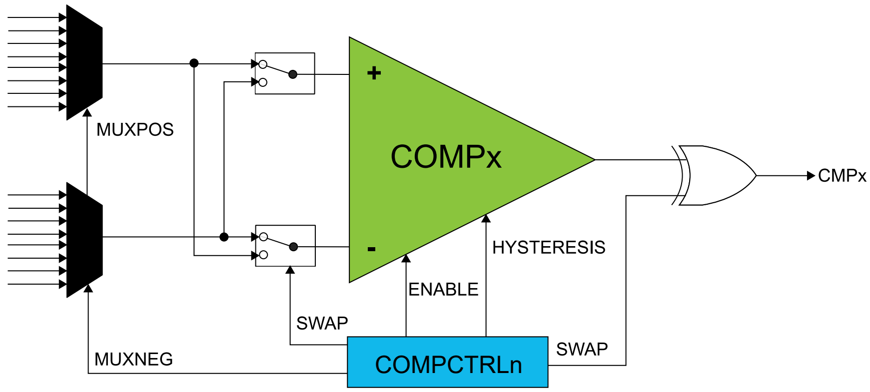
45.6.9. Comparator Output

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. Inputs Swapping

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 figure below. As part of the input selection, COMPCTRLn.SWAP can be changed only while the comparator is disabled.

Figure 45-8. Input Swapping



45.6.11. Interrupts

The AC has the following interrupt sources:

- Comparator (COMPx, x = {0,1}): Indicates a change in comparator status.
- Window (WIN0): Indicates a change in the window status.

Comparator interrupts are generated based on the conditions selected by the Interrupt Selection bit group in the Comparator Control registers (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).

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.

45.6.12. Events

The AC can generate the following output events:

- Comparator (COMPx, x={0,1}): Generated as a copy of the comparator status
- Window (WIN0): Generated as a copy of the window inside/outside status

Writing a one to an Event Output bit in the Event Control Register (EVCTRL.xxEO) enables the corresponding output event. Writing a zero to this bit disables the corresponding output event. Refer to the Event System chapter for details on configuring the event system.

The AC can take the following action on an input event:

- Start comparison (STARTx, x={0,1}): Start a 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 n 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 45-1. Sleep Mode Operation

COMPCTRLn.MODE	RUNSTDBY=0	RUNSTDBY=1
0 (Continuous)	GCLK_AC stopped, COMPx disabled	GCLK_AC stopped or running (depends on filtering options), COMPx enabled
1 (Single-shot)	GCLK_AC stopped, COMPx disabled	GCLK_AC stopped, COMPx enabled only when triggered by an input event

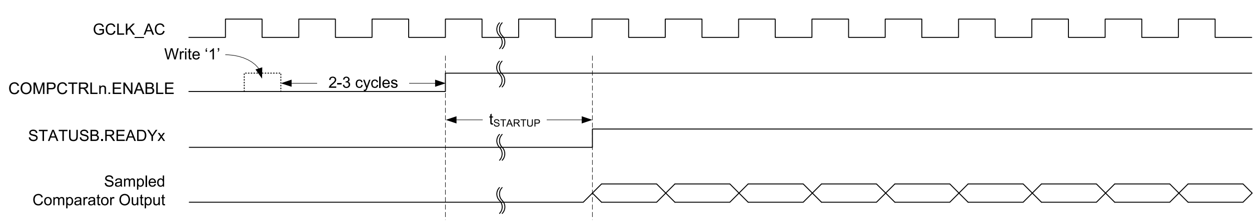
The Run in Standby bits in the Comparator n Control registers (COMPCTRLn.RUNSTDBY) also controls the behavior of the internal DAC during standby sleep mode, if the DAC is selected a positive or negative multiplexor input (COMPCTRLn.MUXPOS = INTDAC, COMPCTRLn.MUXNEG = INTDAC). When the bit is zero, the DAC is disabled during sleep, but maintains its current configuration. When the bit is one, the DAC continues to operate during sleep. Note that when RUNSTDBY is zero, the DAC blocks are powered off for the lowest power consumption. This necessitates a start-up time delay when the system returns from sleep.

45.6.13.1. Continuous Measurement during Sleep

When a comparator is enabled in continuous measurement mode and the filtering options are disabled, the GCLK_AC is disabled during sleep and 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.

If the filtering is enabled, the GCLK_AC is kept running, the comparator will remain continuously enabled and will function synchronously. The current state of the comparator is synchronously monitored for changes. If an edge matching the interrupt condition is found and the interrupt is enabled in the Interrupt Enable registers (INTENCLR/SET), the AC can wake up the device.

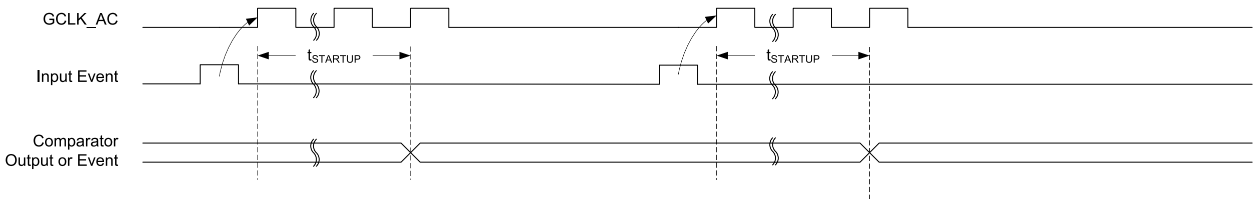
Figure 45-9. Continuous Mode SleepWalking



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 the following figure. The comparator and GCLK_AC are then disabled again automatically, unless configured to wake the system from sleep. Filtering is allowed with this configuration.

Figure 45-10. Single-Shot SleepWalking



45.7. Register Summary

For descriptions and definitions of both Register and bitfield properties, refer to Register Properties.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRLA	31:24								
		23:16								
		15:8								
		7:0							ENABLE	SWRST
0x04	CTRLB	31:24								
		23:16								
		15:8								
		7:0							START1	START0
0x08	CTRLC	31:24						PRESCALER[2:0]		
		23:16			PER[9:4]					
		15:8	PER[3:0]						WIDTH[9:8]	
		7:0	WIDTH[7:0]							
0x0C	EVCTRL	31:24							INVEI1	INVEI0
		23:16							COMPEI1	COMPEI0
		15:8								WINEO0
		7:0							COMPEO1	COMPEO0
0x10	INTENCLR	31:24								
		23:16								
		15:8								WIN0
		7:0							COMP1	COMP0
0x14	INTENSET	31:24								
		23:16								
		15:8								WIN0
		7:0							COMP1	COMP0
0x18	INTFLAG	31:24								
		23:16								
		15:8								WIN0
		7:0							COMP1	COMP0
0x1C	STATUSA	31:24								
		23:16							WSTATE0[1:0]	
		15:8								
		7:0							STATE1	STATE0
0x20	STATUSB	31:24								
		23:16								
		15:8								
		7:0							READY1	READY0
0x24	DBGCTRL	31:24								
		23:16								
		15:8								
		7:0								DBGRUN
0x28	SYNCBUSY	31:24								
		23:16								
		15:8						WINCTRL0		
		7:0					COMPCTRL1	COMPCTRL0	ENABLE	SWRST
0x2C ... 0x33	Reserved									
0x34	COMPCTRL0	31:24	SUT[5:0]						OUT[1:0]	
		23:16	FLEN[2:0]			HYST[1:0]			SPEED	SWAP
		15:8		MUXPOS[2:0]				MUXNEG[2:0]		
		7:0		RUNSTDBY	INTSEL[1:0]		SINGLE		ENABLE	
0x38	DACCTRL	31:24	SHEN1							
		23:16		VALUE1[6:0]						
		15:8	SHEN0							
		7:0		VALUE0[6:0]						

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x3C	WINCTRL	31:24								
		23:16								
		15:8								
		7:0						WINTSEL[1:0]		WEN

45.7.1. Comparator Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00000000
Property: PAC Write-Protection, Write-Synchronized

Table 45-2. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

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

Bit 1 – ENABLE Enable

Due to synchronization, there is delay from updating the register until the peripheral is enabled/ disabled. The value written to CTRL.ENABLE will read back immediately and the corresponding bit in the Synchronization Busy register (SYNCBUSY.ENABLE) will be set. SYNCBUSY.ENABLE is cleared when the peripheral is enabled/disabled.

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 1. When the CTRLA.SWRST is written, the user should poll the SYNCB.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.

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.
Due to synchronization, there is a delay from writing CTRLA.SWRST until the reset is complete. CTRLA.SWRST and SYNCBUSY.SWRST will both be cleared when the reset is complete.

Note: To avoid spurious interrupts from enable/disable cycles, use the SWRST bit to reset the comparator module.

Value	Description
0	There is no reset operation ongoing.
1	The reset operation is ongoing.

45.7.2. Comparator Control B

Name: CTRLB
Offset: 0x04
Reset: 0x00000000
Property: -

Table 45-3. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
							START1	START0
Access							R/S	R/S
Reset							0	0

Bits 0, 1 – STARTn Comparator n Start Comparison (n=0,1)
Writing a '0' to this field has no effect.
Writing a '1' to STARTn starts a single-shot comparison on COMPn if both the Single-Shot and Enable bits in the Comparator n Control Register are '1' (COMPCTRLn.SINGLE and COMPCTRLn.ENABLE), if enabled in single-shot mode. Writing a '1' has no effect.
This bit always reads as zero.
Note: If COMPCTRLn.SINGLE=1 and WINCTRL.WIN=1, window single shot mode, it is only necessary to set one of the CTRLB.STARTn bits for the paired window comparators.

45.7.3. Comparator Control C

Name: CTRLC
Offset: 0x08
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Table 45-4. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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

Bit	23	22	21	20	19	18	17	16
			PER[9:4]					
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
	PER[3:0]						WIDTH[9:8]	
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
	WIDTH[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 26:24 – PRESCALER[2:0] Prescaling Factor

These bits define the prescaling factor for the AC clock source (GCLK_AC) to generate the DAC sampling clock as shown in the following table.

Value	Name	Description
`0x0`	DIV1	Sampling rate is GCLK_AC (No division)
`0x1`	DIV2	Sampling rate is GCLK_AC/2
`0x2`	DIV4	Sampling rate is GCLK_AC/4
`0x3`	DIV8	Sampling rate is GCLK_AC/8
`0x4`	DIV16	Sampling rate is GCLK_AC/16
`0x5`	DIV32	Sampling rate is GCLK_AC/32
`0x6`	DIV64	Sampling rate is GCLK_AC/64
`0x7`	DIV128	Sampling rate is GCLK_AC/128

Bits 21:12 – PER[9:0] DAC Sample and Hold Clock Period

These bits configure the sample and hold clock period. If PER is set to zero, no sample and hold DAC clock is generated.

Note: These bits are ignored if DACCTRLn.SHENn=0 (i.e. DAC continuous operation mode is enabled).

Bits 9:0 – WIDTH[9:0] DAC Sample and Hold Clock Pulse Width

These bits configure the sample and hold clock pulse width. If WIDTH is set to zero, no sample and hold DAC clock is generated.

Note: These bits are ignored if DACCTRLn.SHENn=0 (i.e. DAC continuous operation mode is enabled).

45.7.4. Comparator Event Control

Name: EVCTRL
Offset: 0xC
Reset: 0x0000
Property: PAC Write-Protection, Enable-Protected

Table 45-5. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
							INVEI1	INVEI0
Access							R/W	R/W
Reset							0	0

Bit	23	22	21	20	19	18	17	16
							COMPEI1	COMPEI0
Access							R/W	R/W
Reset							0	0

Bit	15	14	13	12	11	10	9	8
								WINEO0
Access								R/W
Reset								0

Bit	7	6	5	4	3	2	1	0
							COMPEO1	COMPEO0
Access							R/W	R/W
Reset							0	0

Bits 24, 25 – INVEIx Inverted Event Input Enable x

Value	Description
0	Incoming event is not inverted for comparator x.
1	Incoming event is inverted for comparator x.

Bits 16, 17 – COMPEIx Comparator x Event Input

Note that several actions can be enabled for incoming events. If several events are connected to the peripheral, the enabled action will be taken for any of the incoming events. There is no way to tell which of the incoming events caused the action.
These bits indicate whether a comparison will start or not on any incoming event.

Value	Description
0	Comparison will not start on any incoming event.
1	Comparison will start on any incoming event.

Bit 8 – WINEO0 Window 0 Event Output Enable

These bits indicate whether the window function can generate a peripheral event or not.

Value	Description
0	Window 0 Event is disabled.
1	Window 0 Event is enabled.

Bits 0, 1 – COMPEOx Comparator x Event Output Enable
These bits indicate whether the comparator x output can generate a peripheral event or not.

Value	Description
0	COMPx event generation is disabled.
1	COMPx event generation is enabled.

45.7.5. Comparator Interrupt Enable Clear

Name: INTENCLR
Offset: 0x10
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).

Table 45-6. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
								WIN0
Access								R/W
Reset								0

Bit	7	6	5	4	3	2	1	0
							COMP1	COMP0
Access							R/W	R/W
Reset							0	0

Bit 8 – WIN0 Window 0 Interrupt Enable

Reading this bit returns the state of the Window 0 interrupt enable.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit disables the Window interrupt.

Value	Description
0	The Window 0 interrupt is disabled.
1	The Window 0 interrupt is enabled.

Bits 0, 1 – COMPx Comparator x Interrupt Enable

Reading this bit returns the state of the Comparator x interrupt enable.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit disables the Comparator x interrupt.

Value	Description
0	The Comparator x interrupt is disabled.
1	The Comparator x interrupt is enabled.

45.7.6. Comparator Interrupt Enable Set

Name: INTENSET
Offset: 0x14
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 register (INTENCLR).

Table 45-7. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
								WIN0
Access								R/W
Reset								0

Bit	7	6	5	4	3	2	1	0
							COMP1	COMP0
Access							R/W	R/W
Reset							0	0

Bit 8 – WIN0 Window 0 Interrupt Enable
Reading this bit returns the state of the Window 0 interrupt enable.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit enables the Window 0 interrupt.

Value	Description
0	The Window 0 interrupt is disabled.
1	The Window 0 interrupt is enabled.

Bits 0, 1 – COMPx Comparator n Interrupt Enable (x=0,1)
Reading this bit returns the state of the Comparator n 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.

45.7.7. Comparator Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x18
Reset: 0x00000000
Property: –

Note: Interrupt flags must be cleared and then read back to confirm they are cleared before exiting the ISR to avoid double interrupts.

Table 45-8. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
								WIN0
Access								R/W/HS
Reset								0

Bit	7	6	5	4	3	2	1	0
							COMP1	COMP0
Access							R/W/HS	R/W/HS
Reset							0	0

Bit 8 – WIN0 Window 0
This flag is set according to the Window 0 Interrupt Selection bit group in the WINCTRL register (WINCTRL.WINTSEL0) and will generate an interrupt if INTENCLR/SET.WIN0 is also one.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Window 0 interrupt flag.

Bits 0, 1 – COMPn Comparator n Interrupt Status (n=0,1)
Reading this bit returns the status of the Comparator n interrupt flag.
This flag is set according to the Interrupt Selection bit group in the Comparator n Control register (COMPCTRLn.INTSEL) and will generate an interrupt if INTENCLR/SET.COMPn is also one.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Comparator n interrupt flag.

45.7.8. Comparator Status A

Name: STATUSA
Offset: 0x1C
Reset: 0x00000000
Property: Read-Only

Table 45-9. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
							WSTATE0[1:0]	
Access							R	R
Reset							0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
							STATE1	STATE0
Access							R	R
Reset							0	0

Bits 17:16 – WSTATE0[1:0] Window 0 Current State
These bits show the current state of the signal if the window 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 – STATEn Comparator n Current State (n=0,1)
This bit shows the current state of the output signal from COMPn. STATEn is valid only when STATUSB.READYn=1.

45.7.9. Comparator Status B

Name: STATUSB
Offset: 0x20
Reset: 0x00000000
Property: Read-Only

Table 45-10. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
							READY1	READY0
Access							R/HS/HC	R/HS/HC
Reset							0	0

Bits 0, 1 – READYx Comparator x Ready
This bit is cleared when the comparator n output is not ready.
This bit is set when the comparator n output is ready.
Note: In single shot mode, the READY is also cleared when a comparison is triggered by software (CTRLB.STARTx = 1). When the comparison is triggered by an event, the READY is set after the first comparison and stay set as long as the comparator is enabled.

45.7.10. Comparator Debug Control

Name: DBGCTRL
Offset: 0x24
Reset: 0x00000000
Property: PAC Write-Protection

Table 45-11. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

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.

45.7.11. Comparator Synchronization Busy

Name: SYNCBUSY
Offset: 0x28
Reset: 0x00000000
Property: -

Table 45-12. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
						WINCTRL0		
Access						R		
Reset						0		

Bit	7	6	5	4	3	2	1	0
					COMPCTRL1	COMPCTRL0	ENABLE	SWRST
Access					R	R	R	R
Reset					0	0	0	0

Bit 10 – WINCTRL0 WINCTRL0 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.

Bits 2, 3 – COMPCTRLn COMPCTRLn Synchronization Busy (n=0,1)
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 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.

45.7.12. Comparator Control 0

Name: COMPCTRL0
Offset: 0x34
Reset: 0x00000000
Property: PAC Write-Protection, Enable Protected

Table 45-13. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	SUT[5:0]						OUT[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	23	22	21	20	19	18	17	16
	FLEN[2:0]			HYST[1:0]			SPEED	SWAP
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
		MUXPOS[2:0]				MUXNEG[2:0]		
Access		R/W	R/W	R/W		R/W	R/W	R/W
Reset		0	0	0		0	0	0

Bit	7	6	5	4	3	2	1	0
		RUNSTDBY	INTSEL[1:0]		SINGLE		ENABLE	
Access		R/W	R/W	R/W	R/W		R/W	
Reset		0	0	0	0		0	

Bits 31:26 – SUT[5:0] Start-up Time

Each time a comparator is enabled, the comparison will be enabled after the startup time specified by these bits using the formula:
SUT = (20us / ((2^PRESCALER) x t(GCLK_AC))).
Refer to electrical specifications for minimum analog comparator start-up time required to initialize COMPCTRLn.SUT bits to.

Notes:

1. If COMPCTRLn.SINGLE=1 even if comparator COMPCTRLn.ENABLE=1 the comparator is turned off to conserve power. When CTRLB.STARTn is set by the user to start the comparison the comparator is enabled, after COMPCTRLn.SUT time has expired the comparison is made and then the comparator is again shut off to conserve power ready for the next time.
2. These bits can be written only while COMPCTRLn.ENABLE is zero.
3. Zero value is not allowed and can lead to unpredictable behavior.

Bits 25:24 – 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: If the asynchronous path is selected, the filter settings are ignored.

Note: If OUT = 0x2 = Sync, Filtering must be enabled, FLEN>0.

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

Bits 23:21 – FLEN[2:0] Filter Length

These bits configure the filtering for comparator n. COMPCTRLn.FLEN can only be written while COMPCTRLn.ENABLE is zero.

Note: If COMPCTRLn.OUT=0x1 (Asyncronous Mode), these bits are ignored.

Value	Name	Description
0x0	OFF	No filtering (Not valid if COMPCTRLn.OUT = 0x2 = sync)
0x1	MAJ3	3-bit majority function (2 of 3)
0x2	MAJ5	5-bit majority function (3 of 5)
0x3-0x7	N/A	Reserved

Bits 20:19 – 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.

Value	Name	Description
0x0	HYST10	10mV (Refer to AC Electrical Specifications)
0x1	HYST20	20mV (Refer to AC Electrical Specifications)
0x2	HYST40	40mV (Refer to AC Electrical Specifications)
0x3	HYST60	60mV (Refer to AC Electrical Specifications)

Bit 17 – SPEED Speed Selection

This bit indicates the speed/propagation delay mode of comparator n. COMPCTRLn.SPEED can be written only while COMPCTRLn.ENABLE is zero.

Note: High speed equates to higher operating current as well as faster response time.

Value	Name	Description
0x0	HIGH	High speed, high power
0x1	LOW	Low speed, low power

Bit 16 – 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.

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.

Value	Name	Description
0x0	AC_AIN0	Comparator(n) Positive analog input AIN0
0x1	AC_AIN1	Comparator(n) Positive analog input AIN1
0x2	AC_AIN2	Comparator(n) Positive analog input AIN2
0x3	AC_AIN3	Comparator(n) Positive analog input AIN3
0x4	AVSS	Internal AVSS connection

Value	Name	Description
0x5	Reserved	Reserved
0x6	Reserved	Reserved
0x7	INTDAC	Internal DACn

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.

Value	Name	Description
0x0	AC_AIN0	Comparator(n) Negative analog input AIN0
0x1	AC_AIN1	Comparator(n) Negative analog input AIN1
0x2	AC_AIN2	Comparator(n) Negative analog input AIN2
0x3	AC_AIN3	Comparator(n) Negative analog input AIN3
0x4	AVSS	Internal AVSS connection
0x5	BANDGAP	Internal 0.8v Bandgap
0x6	AVSS	AVSS
0x7	INTDAC	Internal DACn

Bit 6 – RUNSTDBY Run in Standby

This bit controls the behavior of the comparator during standby sleep mode. This bit can only be written while COMPCTRLn.ENABLE is zero.

Value	Description
0	The comparator n is disabled during sleep.
1	The comparator n continues to operate during sleep.

Bits 5:4 – INTSEL[1:0] Interrupt Selection

These bits select the condition for comparator n (n=0,1) to generate an interrupt or event. COMPCTRLn.INTSEL can be written only while COMPCTRLn.ENABLE is zero.

Value	Name	Description
0x0	TOGGLE	Interrupt on comparator n output toggle
0x1	RISING	Interrupt on comparator n output rising
0x2	FALLING	Interrupt on comparator n output falling
0x3	EOC	Interrupt on end of comparison (single-shot mode only)

Bit 3 – SINGLE Single-Shot Mode

This bit determines the operation of comparator n. COMPCTRLn.SINGLE can be written only while COMPCTRLn.ENABLE is zero.

Note: To initiate a single-shot comparison the user's software must write the respective CTRLB.STARTn = 1. Be aware that If COMPCTRLn.SINGLE=1 even if comparator COMPCTRLn.ENABLE=1 the comparator is turned off to conserve power.

When CTRLB.STARTn is set by the user to start the comparison the comparator is enabled, after COMPCTRLn.SUT time has expired the comparison is made and then the comparator is again shut off to conserve power ready for the next time.

Value	Description
0	Comparator n operates in continuous measurement mode.
1	Comparator n operates in single-shot mode.

Bit 1 – ENABLE Enable

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.

Value	Description
0	Writing a zero to this bit disables comparator n.
1	Writing a one to this bit enables comparator n.

45.7.13. Comparator DAC Control

Name: DACCTRL
Offset: 0x38
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Note: This register can only be written while COMPCTRLn.ENABLE =0.

Table 45-14. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	SHEN1							
Access	R/W							
Reset	0							

Bit	23	22	21	20	19	18	17	16
		VALUE1[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
	SHEN0							
Access	R/W							
Reset	0							

Bit	7	6	5	4	3	2	1	0
		VALUE0[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit 31 – SHEN1 Comparator1 DAC1 Sample and Hold Enable Operating Mode
This bit enables the odd DAC low-power operation.
Note: If DACCTRL0.SHEN0=1 user must initialize AC.CTRLC.PER and AC.CTRLC.WIDTH accordingly.

Note: This bit can be written only while COMPCTRL1.ENABLE is zero.

Value	Description
0	Continuous operation mode is enabled.
1	Sample-and-hold operation mode is enabled.

Bits 22:16 – VALUE1[6:0] Comparator1 DAC1 Output Value
These bits define the scaling factor for odd DAC channel voltage reference. The output voltage, VOUT, is:

$$V_{OUT} = \frac{AVDD \bullet (VALUE)}{128}$$

Note: This bit can be written only while COMPCTRL1.ENABLE is zero.

Bit 15 – SHEN0 Comparator0 DAC0 Sample and Hold Enable Operating Mode
This bit enabled the even DAC low-power operation.

Note: If DACCTRLn.SHENn=1 user must initialize AC.CTRLC.PER and AC.CTRLC.WIDTH accordingly.

Note: This bit can be written only while COMPCTRL0.ENABLE is zero.

Value	Description
0	Continuous operation mode is enabled.
1	Sample-and-hold operation mode is enabled.

Bits 6:0 – VALUE0[6:0] Comparator0 DAC0 Output Value

These bits define the scaling factor for even DAC channel voltage reference. The output voltage, VOUT, is:

$$V_{OUT} = \frac{AVDD \bullet (VALUE)}{128}$$

These bits can be written only while COMPCTRL0.ENABLE is zero.

45.7.14. Comparator Window Control

Name: WINCTRL
Offset: 0x3C
Reset: 0x00000000
Property: PAC Write-Protection, Write-Synchronized

Table 45-15. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

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

Bits 2:1 – WINTSEL[1:0] Window n Interrupt Selection

These bits configure the interrupt mode for the comparator n window 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 (above or below) window

Bit 0 – WEN Window n Mode Enable

Note: If WINCTRL.WIN=1 then both pair of appropriate comparators must have identical COMPCTRLn.SINGLE bit value settings.

Value	Description
0	Window mode is disabled for Comparator 0 and 1
1	Window mode is enabled for Comparator 0 and 1

46. Position Decoder (PDEC)

46.1. Overview

The PDEC consists of a Quadrature/Hall decoder, followed by a counter, with two compare channels. The counter can be split into two parts to report the angular position and the number of revolutions. If the quadrature decoder feature is not suitable for specific applications, the PDEC module can be used as an additional time base.

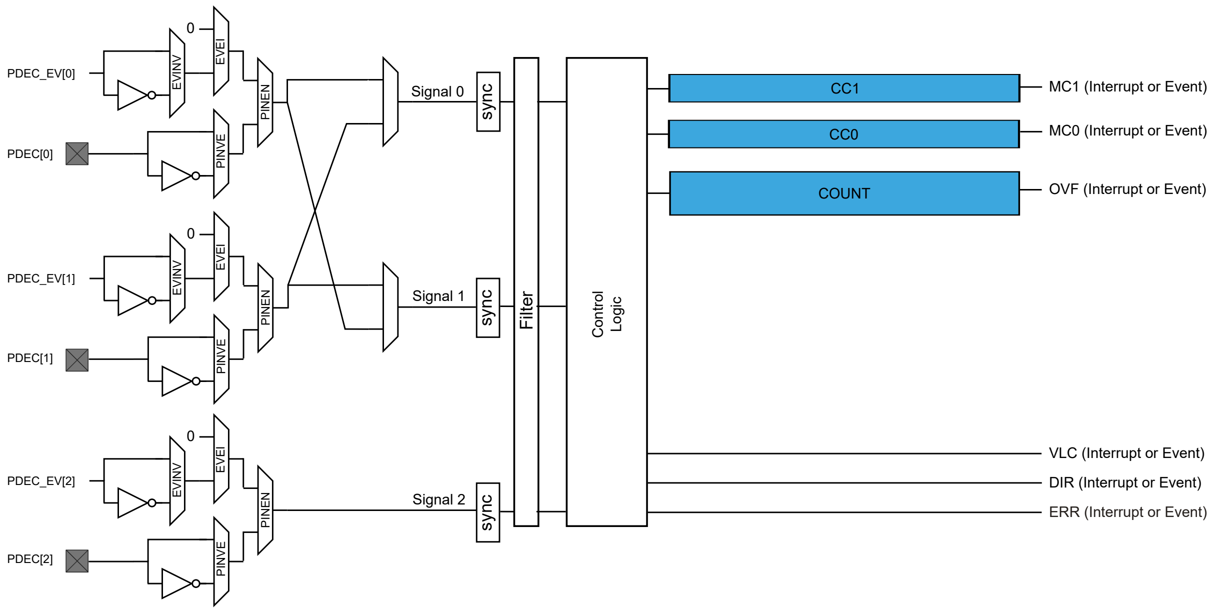
46.2. Features

The following are key features of the module:

- Internal prescaler
- Selectable mode of operation:
  - QDEC, HALL or COUNTER
- QDEC
  - Angular and revolution counts
  - Synchronous and asynchronous velocity measurements
  - Direction change detection
  - Check valid quadrature transitions
  - Check index position versus angular position
  - Auto correction mode
- HALL
  - Window validation of Hall transitions
  - Hall code detection
  - Direction change detection
  - Check valid Hall transitions
  - Programmable event generation delay after a Hall transition
- COUNTER
  - 16-bit counter with two compare channels
  - One of the compare channels can be configured with period settings
  - Counter overflow interrupt and event generation option
  - Compare match interrupt and event generation option

46.3. Block Diagram

Figure 46-1. Block Diagram



46.4. Signal Description

Signal Name	Type	Description
PDEC[2:0]	Digital input	PDEC inputs

Note: One signal can be mapped on one of several pins.

46.5. Peripheral Dependencies

Peripheral Name	Base Address	NVIC IRQ Index: Source	MCLK AHBx/APBx Clock Enable Mask Bit	GCLK Peripheral Channel Clock Name:Register	PAC Peripheral Identifier (PAC.WRCTRL.PERIDx)	EVSYS Users (EVSYS.USERm)	EVSYS Generator (EVSYS.CHANNELn.EVGENx)	Power Domain
PDEC	0x4501_C000	142 : DIR, ERR, MC_0, MC_1, OVF, VLC	MCLK.CLKMSK3[14]	GCLK_PDEC : GCLK.PCHCTRL[38]	50	97-99 : EVU_x	118 : DIR 119 : ERR 120 : MCX_0 121 : MCX_1 122 : OVF 123 : VLC	VDDREG

46.6. Functional Description

46.6.1. Principle of Operation

The PDEC control logic can be driven by a set of three inputs signal coming from Event System channels or I/O input pins. These three inputs can be filtered prior to down-stream processing. The input polarity, phase definition and other factors are configurable. QDEC, HALL or COUNTER mode of operation are supported.

Depending of the mode configuration, specific input sequences can generate:

- State change
- Counter increment or decrement
- Interrupts
- Output events

46.6.2. Basic Operation

46.6.2.1. Initialization

The following PDEC registers are enable-protected, meaning they can only be written when the PDEC is disabled (CTRLA.ENABLE is zero):

- Event Control register (EVCTRL)

Enable-protection is denoted by the 'Enable-Protected' property in the register description.

The following register bits are enable-protected, meaning that they can only be written when the PDEC is disabled (CTRLA.ENABLE=0):

- Maximum Consecutive Missing Pulses bits in Control A register (CTRLA.MAXCMP[3:0])
- Angular Counter Length bits in Control A register (CTRLA.ANGULAR[2:0])
- I/O Pin x Invert Enable bits in Control A register (CTRLA.PINVEN[2:0])
- PDEC Input From Pin x Enable bits in Control A register (CTRLA.PINEN[2:0])
- Period Enable bit in Control A register (CTRLA.PEREN)
- PDEC Phase A and B Swap bit in Control A register (CTRLA.SWAP)
- Auto Lock bit in Control A register (CTRLA.ALOCK)
- PDEC Configuration bits in Control A register (CTRLA.CONF[2:0])
- Run in Standby bit in Control A register (CTRLA.RUNSTDBY)
- Operation Mode bits in Control A register (CTRLA.MODE[1:0])

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'.

46.6.2.2. Enabling, Disabling, and Resetting

The PDEC must be configured before it is enabled by the following steps:

1. Enable the PDEC bus clock (CLK_PDEC_APB).
2. Select the mode of operation by writing the Mode bits in the Control A register (CTRLA.MODE).
3. Select the PDEC mode configuration by writing the Configuration bits in the Control A register (CTRLA.CONF).
4. Select the PDEC event or pin input signal source by writing the Event Enable Input bit in the Event Control register (EVCTRL.EVEI) or the Pin Enable bit in Control A register (CTRLA.PINEN).
5. Select the angular counter length value by writing the Angular bits in the Control A register (CTRLA.ANGULAR).

Optionally, the following configurations can be set before enabling PDEC:

- The GCLK_PDEC clock can be prescaled by writing to the Prescaler register (PRESC).
- A filter can be applied to the input signal by writing a corresponding value to the Filter register (FILTER).

- If the resolution of the rotary sensor is not a power of 2, an Angular period can be set (CTRLA.PEREN and CC0 register).

The PDEC is enabled by writing a '1' to the Enable bit in the Control A register (CTRLA.ENABLE). The PDEC is disabled by writing a '0' to CTRLA.ENABLE.

In QDEC or HALL operation modes, PDEC decoding is enabled writing a START command in the Control B Set register (CTRLBSET.CMD=START). The PDEC decoding is disabled writing a STOP command in the Control B Set register (CTRLBSET.CMD=STOP).

The PDEC is reset by writing a '1' to the Software Reset bit in the Control A register (CTRLA.SWRST). All registers in the PDEC, except DBGCTRL, will be reset to their initial state, and the PDEC will be disabled.

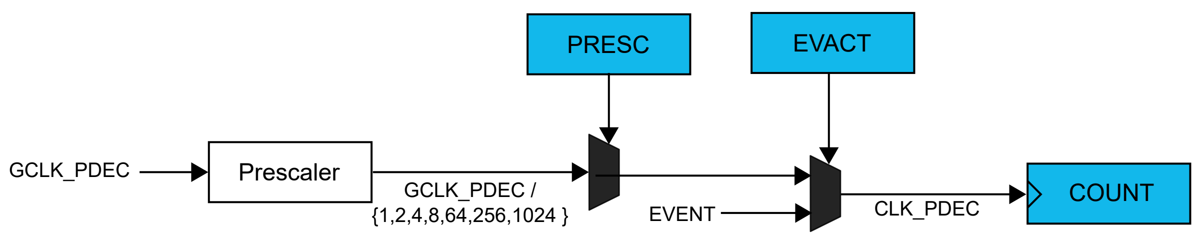
The PDEC should be disabled before the PDEC is reset to avoid undefined behavior.

46.6.2.3. Prescaler Selection

The GCLK_PDEC is fed into the internal prescaler. Prescaler outputs from 1 to 1/1024 are directly available for selection by the counter and all selections are available in Prescaler register (PRESC). If the prescaler value is higher than 0x01, the counter update condition is executed on the next prescaled clock pulse.

The prescaler clock is also enabled when the input filtering is required.

Figure 46-2. Prescaler Selection



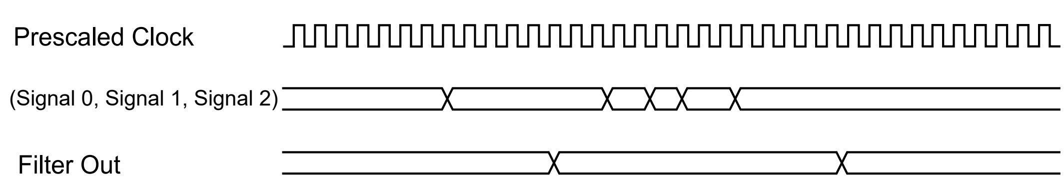
46.6.2.4. Input Selection and Filtering

The QDEC and HALL operations require three inputs, as shown in the Block Diagram. Each input can either be a dedicated I/O pin or an Event system channel. This is selected by writing to the corresponding Event x Enable bit in the Event Control register (EVCTRL.EVEIx) or the Pin x Enable bit in the Control A register (CTRLA.PINENx).

The I/O input pin active level can be inverted by writing to the corresponding Pin x Inversion Enable bit in the Control A register (CTRLA.PINVENx). Similarly, the event input active level can be inverted by writing to the corresponding Inverted Event x Input Enable bit in the Event Control register (EVCTRL.EVINVx).

All input signals can be filtered before they are fed into the control logic. The FILTER register is used to configure the minimum duration for which the input signal must be valid. The input signal minimum duration must be (FILTER +1)* t_{GCLK_PDEC} .

Figure 46-3. Input Signal Filtering



Only the first two input signals can be swapped by writing to the SWAP bit in the Control A register (CTRLA.SWAP).

46.6.2.5. Period Control

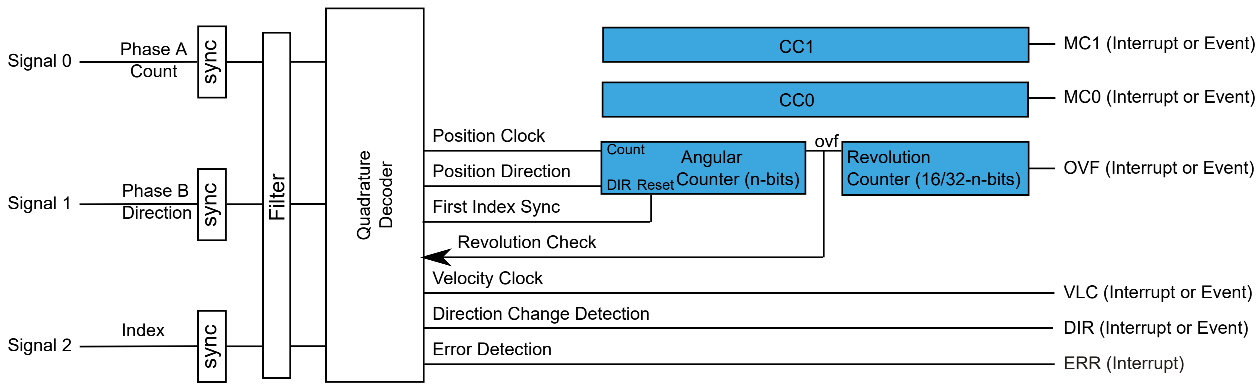
The Channel Compare 0 register (CC0) can act as a period register (PER) by writing the PEREN bit in the Control A register (CTRLA.PEREN) to '1'. The PER can be used to control the top value (TOP) of the counting operation:

When up-counting and the counter reaches the value of CC0, the counter is cleared to zero. When down-counting and the counter reaches zero, the counter is reloaded with the CC0 value.

46.6.2.6. QDEC Operation Mode

In QDEC mode of operation, Signal 0 and Signal 1 control logic inputs refer to Phase A and Phase B in X4 mode, and to count/direction in X2 mode. The Signal 2 control logic input refers to the Index, in both X4 and X2 mode of operation. In X4 mode, a simultaneous transition on Phase A and Phase B will cause a QDEC error detection (STATUS.QERR).

Figure 46-4. QDEC Block Diagram

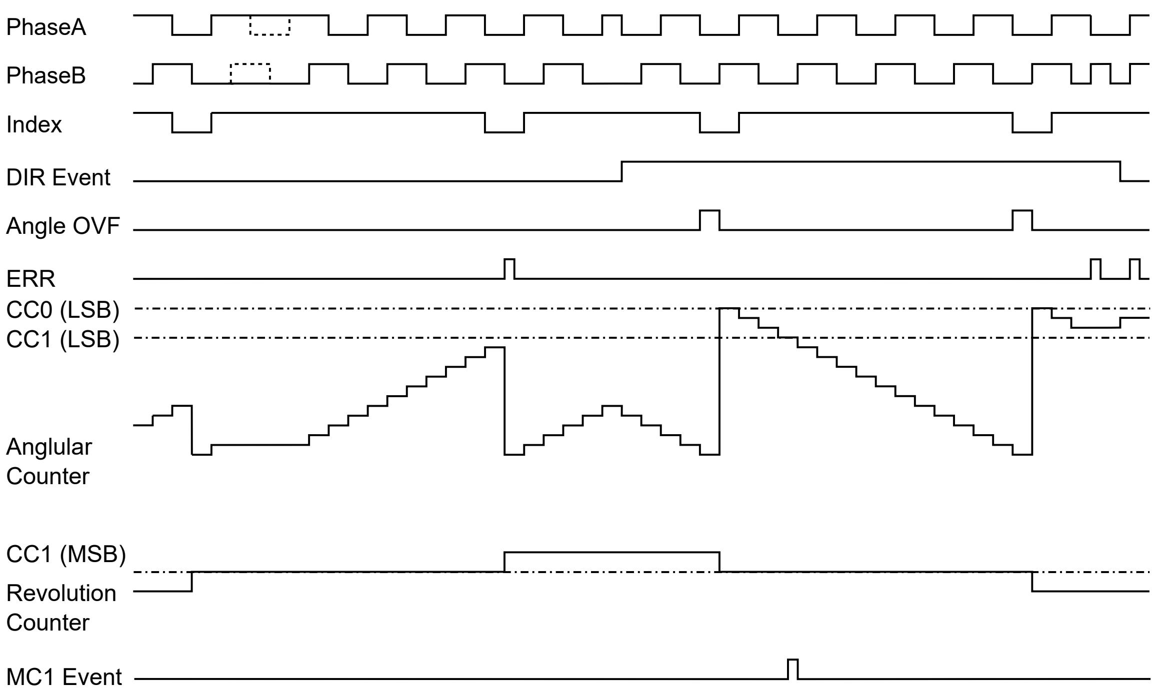


46.6.2.6.1. Position and Rotation Measurement

After filtering, the Quadrature signals are analyzed to extract the rotation direction and edges in order to be counted by the counter.

The counter is split in two parts, Angular and Revolution. The Phase A and B edge detections define the motor axis position, which is recorded by the Angular part of the counter. The motor revolution is recorded by the Revolution part of the counter. The Angular counter is updated each time a QDEC transition is detected. The Revolution counter is updated on each angular counter overflow or underflow.

Figure 46-5. Position and Rotation Measurement



In X4 and X4S configuration, a valid index is detected when the three inputs (PhaseA, PhaseB and Index) are at low level.

In X2 and X2S configuration, a valid index is detected when the two inputs (Count and Index) are at low level.

In X2 and X4 configuration, depending on current detected direction, Index will reset or reload the Angular counter and increment or decrement the Revolution counter.

In X2S and X4S configuration, the Angular counter is reset on the first Index occurrence after the PDEC decoding is enabled. When any next Index occurrence does not match an Angular counter overflow or underflow, the Index Error flag in Status register is set (STATUS.IDXERR). The Error Interrupt Flag is set (INTFLAG.ERR) and an optional interrupt can be generated.

An Index Error is also generated after the PDEC decoding is enabled and no Index has been detected after one Angular counter revolution.

46.6.2.6.2. Secure Decoder Detection

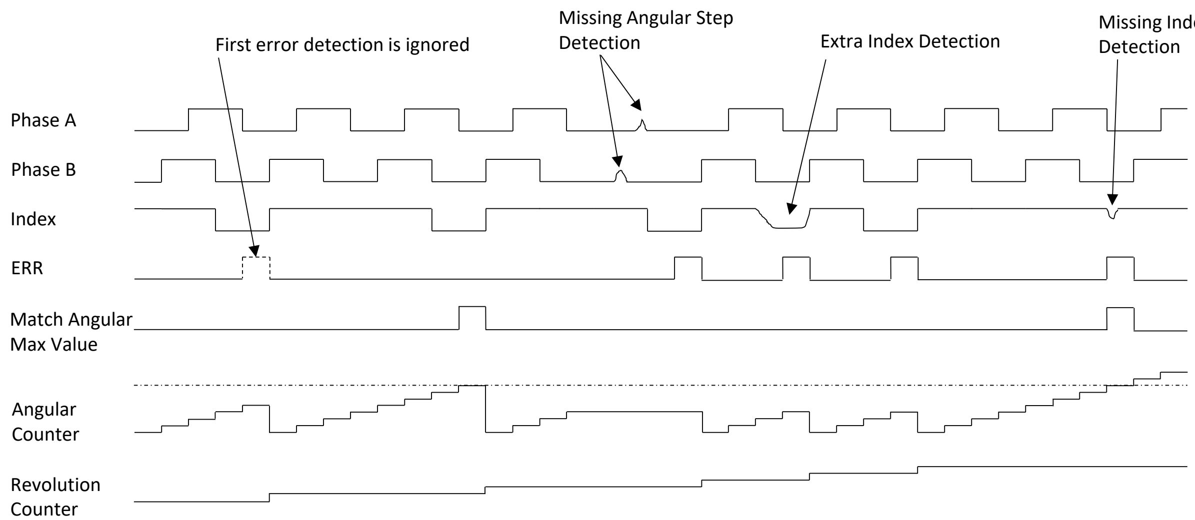
Position decoders are generally implemented using external photo-detectors, detecting a transmitted or reflected beam. Error detections or no detections may happen, and are due to:

- A transmitted beam locally stopped by dust
- A reflected beam lost by a mat dust
- An additional parasitic reflected beam, due to a gloss metallic dust

When secure detection is enabled (CTRLA.CONF = X4S or CTRLA.CONF = X2S), the Index must be enabled. The Angular counter is restarted on the detection of each overflow. When the Angular counter value reaches its maximum period value (TOP), an Index is expected to be detected. In the same way, if the Index is detected, the Angular counter value is expected to be TOP. If one of these conditions is not met, an error is generated.

Note: The first error generation is masked, as the initial position of the wheel is in unknown state. The first index is used to synchronize the Angular counter on Revolution counter increment.

Figure 46-6. Secure Decoder Detections

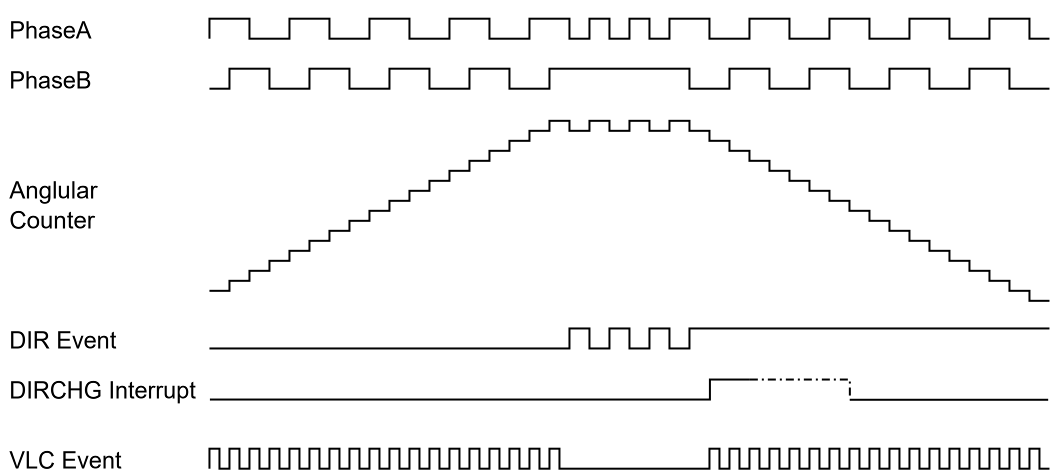


46.6.2.6.3. Direction Status and Change Detection

The direction (DIR) status can be directly read anytime in the STATUS register (STATUS.DIR). The polarity of the direction flag status depends of the input signal swap and active level configuration.

Each time a rotation direction change is detected, the Direction Change Interrupt Flag is set (INTFLAG.DIR) and an optional interrupt can be generated. The same interrupt condition is source of Direction event output.

Figure 46-7. Rotation Direction Change



To avoid spurious interrupts when coding wheel is stopped, the direction change condition is reported as an interrupt, only on the second edge confirming the direction change.

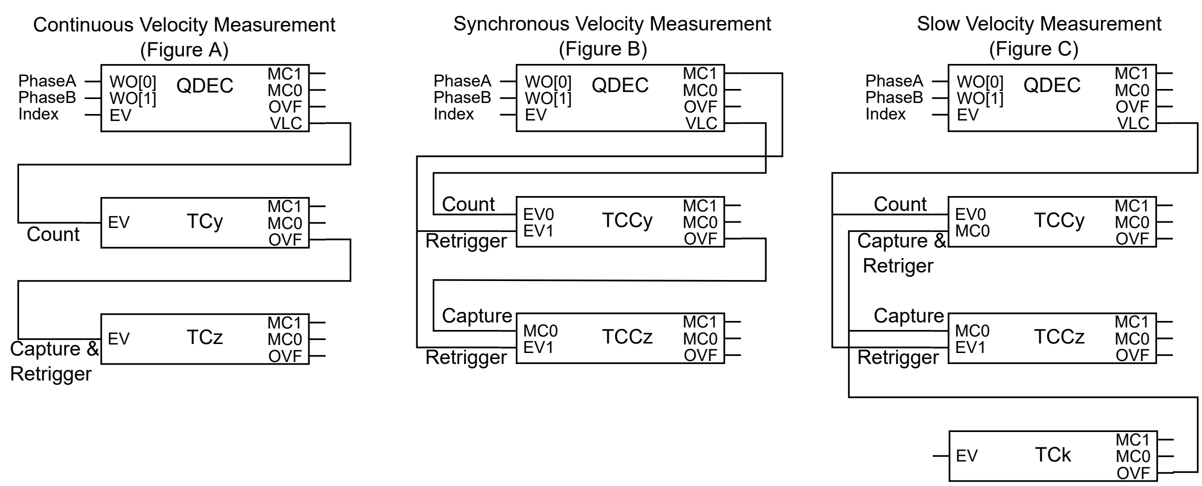
Velocity output event is generated on each QDEC transition except when the direction changes.

46.6.2.6.4. Speed Measurement

Three types of speed measurement can be done using velocity event output (VLC) and Timer/Counter (TC/TCC) device resources.

- Continuous velocity measurement: TCz measures the time on which n VLC (TCy) output events occur
- Synchronous Velocity measurement: On a specific motor position TCCz, the time is measured on which n VLC (TCCy) output events occur.
- Slow Velocity measurement: measure the number of VLC output events (TCCy) plus the delay since the last VLC output event (TCCz) within a given time slot (TCk).

Figure 46-8. Speed Measurement



46.6.2.6.5. Missing Pulse Detection and Auto-Correction

The PDEC embeds circuitry to detect and correct errors that may result from contamination on optical disks or other sources producing quadrature phase signals.

The auto-correction works in QDEC X4 mode only. A missing pulse on a phase signal is automatically detected, and the pulse count reported in the Angular part of COUNT is automatically corrected.

There is no autocorrection if both phase signals are affected at the same location on the input signals, because the autocorrection requires a valid phase signal to detect contamination on the other phase signal.

If the quadrature source is undamaged, the number of pulses counted for a predefined period of time must be the same with or without detection and auto-correction. Therefore, if the measurement results differ, a contamination exists on the source producing the quadrature signals. This does not substitute the measurements of the number of pulses between two index pulses (if available) but provides an additional method to detect damaged quadrature sources.

When the source providing quadrature signals is strongly damaged, potentially leading to a number of consecutive missing pulses greater than 1, the quadrature decoder processing may be affected.

The Maximum Consecutive Missing Pulses bits in Control A register (CTRLA.MAXCMP) define the maximum acceptable number of consecutive missing pulses. If the limit is reached, the Missing Pulse Error flag in Status register (STATUS.MPERR) is set. The Error Interrupt flag is set (INTFLAG.ERR) and an optional interrupt can be generated.

Note: When the MAXCMP value is zero, the MPERR error flag is never set.

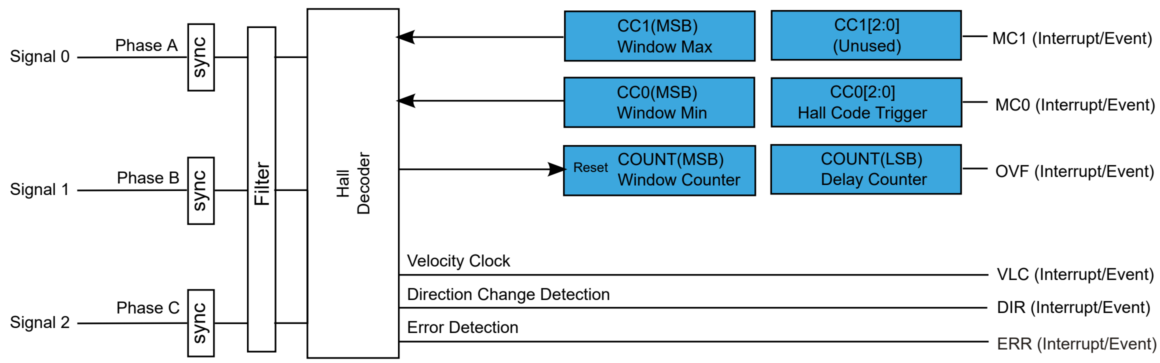
46.6.3. Additional Features

46.6.3.1. HALL Operation Mode

In HALL operation mode, control logic signal 0, 1 and 2 inputs represent the phase A, B and C of a Hall sensor, respectively.

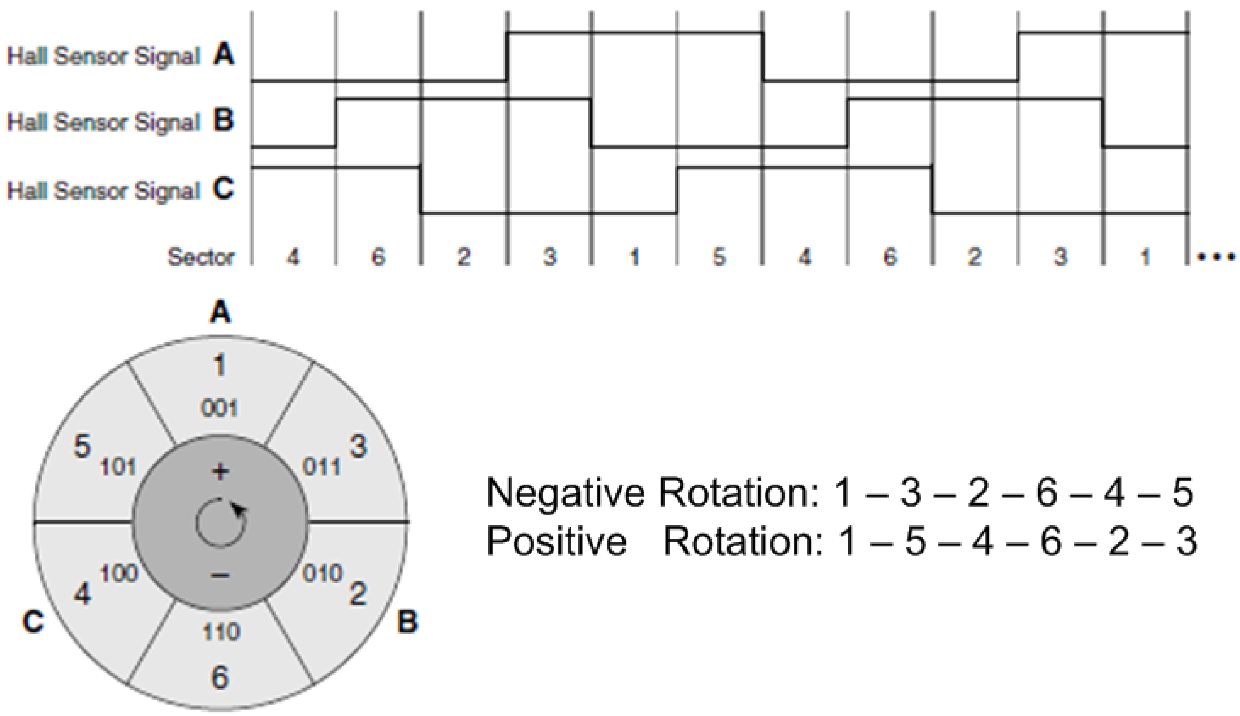
A programmable delayed event can be generated to update a TCC pattern generator.

Figure 46-9. HALL Block Diagram



When positive rotation is detected, the DIR status bit is set (STATUS.DIR = 1). When a negative rotation sequence is detected, the DIR status bit is set (STATUS.DIR = 0).

Figure 46-10. Hall States Overview



46.6.3.1.1. Hall Sensor Control

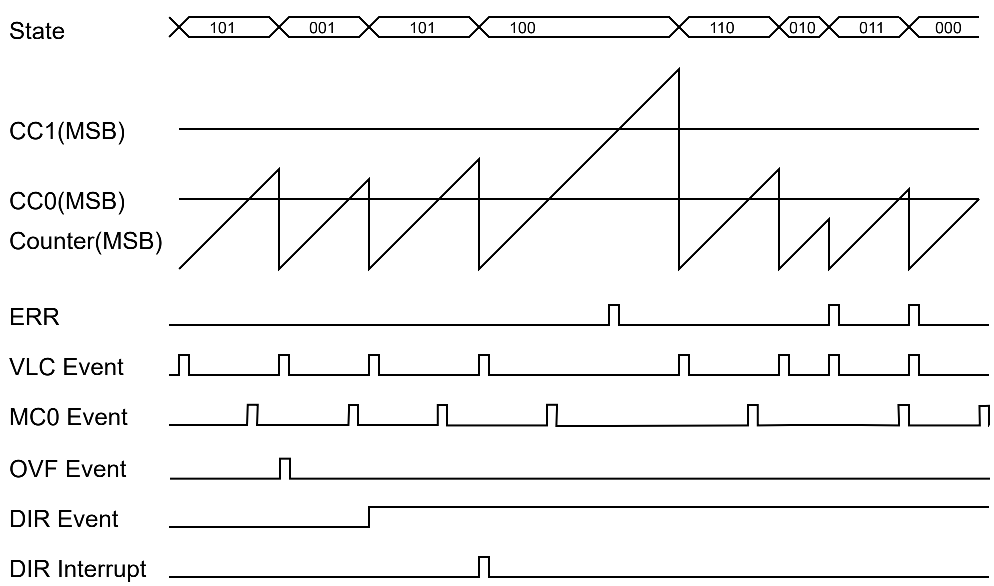
On any update of the filter output:

- The filter output value is checked to be a valid Hall value. If an invalid Hall code is reported, the Hall Error bit in Status register will be set (STATUS.HERR).

- The OVF Interrupt Flag bit is set (INTFLAG.OVF) if CC0[2:0] matches the filter output value, stored in LSB part of the COUNTER. An optional overflow interrupt or Event output is generated on the same condition detection.
- The window counter is checked to be between the value of the MSB part of CC0 and CC1, and reset to 0 value. If an error is detected, the Window Error bit in Status register (STATUS.WINERR) is set.
- The delay counter is started, and MC0 optional interrupt or event is generated when the delay counter matches the MSB part of CC0.
- Optional MC1 interrupt or event is generated when the delay counter matches the MSB part of CC1

Any error condition will set the Error Interrupt Flag (INTFLAG.ERR). An optional interrupt or event output is generated on the same condition detection.

Figure 46-11. Hall Waveforms



46.6.3.2. Counter Operation Mode

Depending on the mode of operation, the counter (Counter Value register COUNT) is cleared, reloaded, or incremented at each counter clock input.

The counter will count for each clock tick until it reaches TOP. When TOP is reached, the counter will be set to zero on the next clock input.

This comparison will set the Overflow Interrupt Flag in the Interrupt Flag Status and Clear register (INTFLAG.OVF) and can be used to trigger an interrupt or an event.

It is possible to change the counter value when the counter is running. The write access has higher priority than count, or clear. The COUNT value will always be zero when starting the PDEC, unless a different value has been written to it, or the PDEC has been disabled at a value other than zero. Due to asynchronous clock domains, the internal counter settings are written once the synchronization is complete.

46.6.3.3. Register Lock Update

Prescaler (PRESC), FILTER, and CCx registers are buffered (PRESCBUF, FILTERBUF, CCBUFx registers, respectively). When a new value is written in a buffer register, the corresponding Buffer Valid bit is set in the Buffer Status register (STATUS.FILTERBUFV, STATUS.PRESCBUFV, STATUS.CCBUFVx).

By default, a register is updated with its buffer register's value on UPDATE condition, which represents:

- The next filter transition in QDEC and HALL mode of operation
- The overflow/underflow or re-trigger event detection in COUNT mode of operation

The buffer valid flags in the STATUS register are automatically cleared by hardware when the data is copied from the buffer to the corresponding register.

It is possible to lock the updates by writing a '1' to the Lock Update bit in Control B Set register (CTRLBSET.LUPD).

The lock feature is disabled by writing a '1' to the Lock Update bit in Control B Clear register (CTRLBCLR.LUPD). When a buffer valid status flag is '1' and updating is not locked, the data from the buffer register will be copied into the corresponding register on UPDATE condition.

It is also possible to modify the LUPD bit behavior by hardware, by writing a '1' to the Auto-lock bit in Control A register (CTRLA.ALOCK). When the bit is '1', the Lock Update bit in Control B register (CTRLBSET.LUPD) is set when the UPDATE condition is detected.

46.6.3.4. Software Command and Event Actions

The PDEC peripheral supports software commands and event actions. The software commands are applied by the Software Command bit field in the Control B register (CTRLBSET.CMD, CTRLBCLR.CMD). The event actions are available in the Event Action bit-field in Event Control register (EVCTRL.EVACT).

46.6.3.4.1. Re-trigger Software Command or Event Action

A re-trigger command can be issued from software by using PDEC Command bits in Control B Set register (CTRLBSET.CMD = RETRIGGER).

When the re-trigger command is detected during counting operation, the counter will be reloaded or cleared, depending on the counting direction (DIR). If the re-trigger command is detected when the counter is stopped, the counter will resume counting operation from the value in the COUNT register.

46.6.3.4.2. Force Update Software Command

A Force Update command can be issued by writing the PDEC Command bits in Control B Set register (CTRLBSET.CMD = UPDATE). When the command is issued, the buffered registers will be updated.

46.6.3.4.3. Force Read Synchronization Software Command

A Force Read Synchronization command can be issued writing the PDEC Command bits in Control B Set register (CTRLBSET.CMD = READSYNC). When the command is issued, a COUNT register read synchronization is forced.

Note: This command should be used to read the most updated COUNT internal value.

46.6.4. Interrupts

The PDEC has the following interrupt sources:

- Overflow/Underflow: OVF
- Compare Channels: COMPx
- Error: ERR
- Velocity: VLC. This interrupt is available only in QDEC and HALL operation modes.
- Direction: DIR. This interrupt is available only in QDEC and HALL operation modes.

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). As both INTENSET and INTENCLR always reflect the same value, the status of interrupt enablement can be read from either 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 PDEC is reset. See the INTFLAG register description for details on how to clear interrupt flags.

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. For additional information, refer to the Nested Vectored Interrupt Controller.

46.6.5. Events

The PDEC can generate the following output events:

- Overflow/Underflow: OVF
- Channel x Compare Match: MCx
- Error: ERR
- Velocity: VLC. This interrupt is available only in QDEC and HALL operation modes.
- Direction: DIR. This interrupt is available only in QDEC and HALL operation modes.

Writing a '1' to an Event Output bit in the Event Control register (EVCTRL.MCEO) enables the corresponding output event. Writing a '0' to this bit disables the corresponding output event.

In counter mode the PDEC can take action on an input event. PDEC counter event input are available for each of the three PDEC channels.

- Retrigger: Restart/retrigger on event

Refer to the Event System (EVSYS) for additional information.

46.6.6. Sleep Mode Operation

The PDEC 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 written to '1'. The PDEC can wake up the device using interrupts from any sleep mode or perform actions through the Event System (EVSYS).

For additional information, refer to: these sections:

- Power Manager (PM)
- Sleep Mode Controller

46.6.7. Synchronization

Due to asynchronicity between the main clock domain and the peripheral clock domains, some registers need to be synchronized when written or read.

The following bits are synchronized when written:

- Software Reset bit in the Control A register (CTRLA.SWRST)
- Enable bit in the Control A register (CTRLA.ENABLE)

The following registers need synchronization when written:

- Control B Clear and Control B Set registers (CTRLBCLR and CTRLBSET)
- Status register (STATUS)
- Prescaler and Prescaler Buffer registers (PRESC and PRESCBUF)
- Compare Value x and Compare Value x Buffer registers (CCx and CCBUFx)

- Filter Value and Filter Buffer Value registers (FILTER and FILTERBUF)
- Counter Value register (COUNT)

Required write synchronization is denoted by the "Write-Synchronized" property in the register description.

The following registers are synchronized when read:

- Counter Value register (COUNT): the synchronization is done on demand through READSYNC software command (CTRLBSET.CMD)

Required read synchronization is denoted by the "Read-Synchronized" property in the register description.

46.7. Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRLA	31:24	MAXCMP[3:0]					ANGULAR[2:0]		
		23:16		PINVEN2	PINVEN1	PINVEN0		PINEN2	PINEN1	PINEN0
		15:8	PEREN	SWAP			ALOCK	CONF[2:0]		
		7:0		RUNSTDBY			MODE[1:0]		ENABLE	SWRST
0x04	CTRLBCLR	7:0	CMD[2:0]						LUPD	
0x05	CTRLBSET	7:0	CMD[2:0]						LUPD	
0x06	EVCTRL	15:8			MCEO1	MCEO0	VLCEO	DIREO	ERREO	OVFEO
		7:0	EVEI[2:0]			EVINV[2:0]			EVACT[1:0]	
0x08	INTENCLR	7:0			MC1	MC0	VLC	DIR	ERR	OVF
0x09	INTENSET	7:0			MC1	MC0	VLC	DIR	ERR	OVF
0x0A	INTFLAG	7:0			MC1	MC0	VLC	DIR	ERR	OVF
0x0B	Reserved									
0x0C	STATUS	15:8			CCBUFV1	CCBUFV0			FILTERBUFV	PRESCBUFV
		7:0	DIR	STOP	HERR	WINERR		MPERR	IDXERR	QERR
0x0E	Reserved									
0x0F	DBGCTRL	7:0								DBGRUN
0x10	SYNCBUSY	31:24								
		23:16								
		15:8								CC1
		7:0	CC0	COUNT	FILTER	PRESC	STATUS	CTRLB	ENABLE	SWRST
0x14	PRESC	7:0					PRESC[3:0]			
0x15	FILTER	7:0	FILTER[7:0]							
0x16 ... 0x17	Reserved									
0x18	PRESCBUF	7:0					PRESCBUF[3:0]			
0x19	FILTERBUF	7:0	FILTERBUF[7:0]							
0x1A ... 0x1B	Reserved									
0x1C	COUNT	31:24								
		23:16								
		15:8	COUNT[15:8]							
		7:0	COUNT[7:0]							
0x20	CC0	31:24								
		23:16								
		15:8	CC[15:8]							
		7:0	CC[7:0]							
0x24	CC1	31:24								
		23:16								
		15:8	CC[15:8]							
		7:0	CC[7:0]							
0x28 ... 0x2F	Reserved									
0x30	CCBUF0	31:24								
		23:16								
		15:8	CCBUF[15:8]							
		7:0	CCBUF[7:0]							
0x34	CCBUF1	31:24								
		23:16								
		15:8	CCBUF[15:8]							
		7:0	CCBUF[7:0]							

46.7.1. Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected Bits, Write-Synchronized Bits

Table 46-1. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	MAXCMP[3:0]					ANGULAR[2:0]		
Access	RW	RW	RW	RW		RW	RW	RW
Reset	0	0	0	0		0	0	0

Bit	23	22	21	20	19	18	17	16
		PINVEN2	PINVEN1	PINVEN0		PINEN2	PINEN1	PINEN0
Access		RW	RW	RW		RW	RW	RW
Reset		0	0	0		0	0	0

Bit	15	14	13	12	11	10	9	8
	PEREN	SWAP			ALOCK	CONF[2:0]		
Access	RW	RW			RW	RW	RW	RW
Reset	0	0			0	0	0	0

Bit	7	6	5	4	3	2	1	0
		RUNSTDBY			MODE[1:0]		ENABLE	SWRST
Access		RW			RW	RW	RW	W
Reset		0			0	0	0	0

Bits 31:28 – MAXCMP[3:0] Maximum Consecutive Missing Pulses

These bits define the threshold for the maximum consecutive missing pulses in AUTOC configuration of the QDEC mode.

Outside of AUTOC configuration of QDEC mode, these bits have no effect.

Note: This bit field is enable-protected. This bit field is not synchronized.

Bits 26:24 – ANGULAR[2:0] Angular Counter Length

In QDEC mode, these bits define the size of the Angular counter within COUNT. Angular counter size is equal to CTRLA.ANGULAR+9. The remaining MSB of the COUNTER register are used for counting revolutions.

For example, CTRLA.ANGULAR=0 defines the 9 LSB of COUNT as Angular counter and the residual 7 MSB of COUNT as Revolution counter. CTRLA.ANGULAR=7 will define a 16-bit Angular counter and no Revolution counter.

Outside of QDEC mode, these bits have no effect.

Note: This bit field is enable-protected. This bit field is not synchronized.

Table 46-2. Angular and Revolution Counters in COUNTER Register

ANGULAR[2:0]	Angular counter	Revolution counter
0x0	COUNTER[0:8]	COUNTER[9:15]
0x1	COUNTER[0:9]	COUNTER[10:15]

Table 46-2. Angular and Revolution Counters in COUNTER Register (continued)

ANGULAR[2:0]	Angular counter	Revolution counter
0x2	COUNTER[0:10]	COUNTER[11:15]
0x3	COUNTER[0:11]	COUNTER[12:15]
0x4	COUNTER[0:12]	COUNTER[13:15]
0x5	COUNTER[0:13]	COUNTER[14:15]
0x6	COUNTER[0:14]	COUNTER[15]
0x7	COUNTER[0:15]	no revolution counter

Bits 20, 21, 22 – PINVENx IO Pin x Invert Enable [x = 2..0]

When this bit is written to '1', the corresponding input pin active level is inverted. This bit has no effect if PINENx bit is zero.

In COUNTER mode only PINVEN[0] is significant.

Note: This bit is enable-protected. This bit is not synchronized.

Value	Description
0	Pin active level is not inverted.
1	Pin active level is inverted.

Bits 16, 17, 18 – PINENx PDEC Input From Pin x Enable [x = 2..0]

This bit enables the IO pin x as signal input.

In COUNTER mode, only PINEN[0] is significant.

Note: This bit is enable-protected. This bit is not synchronized.

Value	Description
0	Event line is the signal input.
1	I/O pin is the signal input.

Bit 15 – PEREN Period Enable

This bit is used to enable the CC0 register as counter period.

Note: This bit is enable-protected. This bit is not synchronized.

Value	Description
0	Period register function is disabled.
1	CC0 is acting as counter period register.

Bit 14 – SWAP PDEC Phase A and B Swap

This bit is used to swap input source of signal 0 and 1.

In COUNTER mode this bit has no effect.

Note: This bit is enable-protected. This bit is not synchronized.

Value	Description
0	The input sources of signal 0 and 1 are not swapped.
1	The input sources of signal 0 and 1 are swapped.

Bit 11 – ALOCK Auto Lock

When this bit is set, the Lock Update bit in Control B register (CTRLB.LUPD) is set by hardware when an UPDATE condition is detected.

Note: This bit is enable-protected. This bit is not synchronized.

Value	Description
0	Auto Lock is disabled.
1	Auto Lock is enabled.

Bits 10:8 – CONF[2:0] PDEC Configuration

These bits define the PDEC configuration.

Outside of QDEC mode, these bits have no effect.

Note: This bit field is enable-protected. This bit field is not synchronized.

Value	Name	Description
0	X4	Quadrature decoder direction
1	X4S	Secure Quadrature decoder direction
2	X2	Decoder direction
3	X2S	Secure decoder direction
4	AUTOC	Auto correction mode

Bit 6 – RUNSTDBY Run in Standby

This bit is used to keep the PDEC running in standby mode.

Note: This bit is enable-protected. This bit is not synchronized.

Value	Description
0	The PDEC is halted in standby.
1	The PDEC continues to run in standby.

Bits 3:2 – MODE[1:0] Operation Mode

These bits select one of the QDEC, HALL, COUNTER modes.

Note: This bit field is enable-protected. This bit field is not synchronized.

Value	Name	Description
0x0	QDEC	QDEC operating mode
0x1	HALL	HALL operating mode
0x2	COUNTER	COUNTER operating mode

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

Write a ‘1’ to this bit to reset the SFR registers in PDEC (except DBGCTRL) to their initial state, and PDEC will be disabled. The bit stays high until reset completes. Setting this bit also sets the SYNCBUSY.SWRST to 1. SYNCBUSY.SWRST stays 1 until reset sequence completes.

Note:

1. When the CTRLA.SWRST is written, the user should poll the SYNCB.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 the hardware.

Value	Description
0	There is no Reset operation ongoing.
1	A Reset operation is ongoing.

46.7.2. 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 (CTRLBSET) register.

Note: This register is write-synchronized: SYNCBUSY.CTRLB must be checked to ensure the CTRLBCLR register synchronization is complete.

Table 46-3. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
	CMD[2:0]						LUPD	
Access	RW	RW	RW				RW	
Reset	0	0	0				0	

Bits 7:5 – CMD[2:0] Command
These bits can be used for software control of the PDEC. When a command has been executed, the CMD bit group will read back zero. The commands are executed on the next prescaled GCLK_PDEC clock cycle.
Writing a zero to this bit group has no effect.
Writing a valid value to these bits will clear the corresponding pending command.
Writing a '0' to these bits has no effect.
Writing a '1' to an individual bit will clear the corresponding bit.

Value	Name	Description
0	NONE	No action
1	RETRIGGER	Force a counter restart or re-trigger
2	UPDATE	Force update of double buffered registers
3	READSYNC	Force a read synchronization of COUNT
4	START	Start QDEC/HALL
5	STOP	Stop QDEC/HALL

Bit 1 – LUPD Lock Update
This bit controls the update operation of the PDEC 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 will disable the lock update.

Value	Description
0	The PRESCBUF, FILTERBUF and CCBUFx buffer registers value are copied into CCx and PER registers on hardware update condition.
1	The PRESCBUF, FILTERBUF and CCBUFx buffer registers value are not copied into CCx and PER registers on hardware update condition.

46.7.3. 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 (CTRLBCLR) register.
Note: This register is write-synchronized: SYNCBUSY.CTRLB must be checked to ensure the CTRLBSET register synchronization is complete.

Table 46-4. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
	CMD[2:0]						LUPD	
Access	RW	RW	RW				RW	
Reset	0	0	0				0	

Bits 7:5 – CMD[2:0] Command
These bits can be used for software control of the PDEC. When a command has been executed, the CMD bit group will read back zero. The commands are executed on the next prescaled GCLK_PDEC clock cycle.
Writing a zero to this bit group has no effect.
Writing a valid value to these bits will set the associated command.

Value	Name	Description
0	NONE	No action
1	RETRIGGER	Force a counter restart or retrigger
2	UPDATE	Force update of double buffered registers
3	READSYNC	Force a read synchronization of COUNT
4	START	Start QDEC/HALL
5	STOP	Stop QDEC/HALL

Bit 1 – LUPD Lock Update
This bit controls the update operation of the PDEC 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 '1' to this will enable the Lock Update.

Value	Description
0	The PRESCBUF, FILTERBUF and CCBUFx buffer registers value are copied into CCx and PER registers on hardware update condition.
1	The PRESCBUF, FILTERBUF and CCBUFx buffer registers value are not copied into CCx and PER registers on hardware update condition.

46.7.4. Event Control

Name: EVCTRL
Offset: 0x06
Reset: 0x0000
Property: Enable-Protected, PAC Write-Protection

Table 46-5. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	15	14	13	12	11	10	9	8
			MCEO1	MCEO0	VLCEO	DIREO	ERREO	OVFEO
Access			RW	RW	RW	RW	RW	RW
Reset			0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	EVEI[2:0]			EVINV[2:0]			EVACT[1:0]	
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 12, 13 – MCEOx Match Channel x Event Output Enable [x = 1..0]

These bits control whether event match on channel x is enabled or not and generated for every match.

Value	Description
0	Match event on channel x is disabled and will not be generated.
1	Match event on channel x is enabled and will be generated for every compare.

Bit 11 – VLCEO Velocity Output Event Enable

This bit is used to enable the velocity event. When enabled, an event level will be generated for each change on the qualified PDEC phases.
This bit has no effect when COUNTER operation mode is selected.

Value	Description
0	VLC output event is disabled and will not be generated.
1	VLC output is enabled and will be generated for every valid velocity condition.

Bit 10 – DIREO Direction Output Event Enable

This bit is used to enable the Direction event. When enabled, an event level output is generated to report the rotation direction.

Value	Description
0	DIR output event is disabled and will not be generated.
1	DIR output is enabled and changes the level when the rotation direction changes.

Bit 9 – ERREO Error Output Event Enable

This bit enables the output of the Error event (ERR).

Value	Description
0	ERR Event output is disabled.
1	ERR Event output is enabled.

Bit 8 – OVFEO Overflow/Underflow Output Event Enable
This bit is used to enable 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.

Bits 7:5 – EVEI[2:0] Event Input Enable
This bit is used to enable asynchronous input event to the counter. The bit position in the EVEI[2:0] bitfield corresponds to the PDEC channel number.

Value	Description
0	Incoming events are disabled.
1	Incoming events are enabled.

Bits 4:2 – EVINV[2:0] Inverted Event Input Enable
This bit inverts the asynchronous input event to the counter. The bit position in the EVINV[2:0] bitfield corresponds to the PDEC channel number.

Value	Description
0	Input event source is not inverted.
1	Input event source is inverted.

Bits 1:0 – EVACT[1:0] Event Action
These bits have an effect only when COUNTER operation mode is selected, and ignored in all other operation modes.
These bits define the event action the counter will perform on an event.

Value	Name	Description
0	OFF	Event action disabled
1	Reserved	Reserved
2	COUNT	Count on event

46.7.5. Interrupt Enable Clear

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

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 Interrupt Enable Set (INTENSET) register.

Table 46-6. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
			MC1	MC0	VLC	DIR	ERR	OVF
Access			RW	RW	RW	RW	RW	RW
Reset			0	0	0	0	0	0

Bits 4, 5 – MCx Channel x Compare Match Disable [x = 1..0]
Writing a '0' to MCx has no effect.
Writing a '1' to MCx will clear the corresponding Match Channel x Interrupt Disable/Enable bit, which disables the Match Channel x interrupt.

Value	Description
0	The Match Channel x interrupt is disabled.
1	The Match Channel x interrupt is enabled.

Bit 3 – VLC Velocity Interrupt Disable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Velocity Interrupt Disable/Enable bit, which disables the Velocity interrupt.
This bit has no effect when COUNTER operation mode is selected.

Value	Description
0	The Velocity interrupt is disabled.
1	The Velocity interrupt is enabled.

Bit 2 – DIR Direction Interrupt Disable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Direction Change Interrupt Disable/Enable bit, which disables the Direction Change interrupt.
This bit has no effect when COUNTER operation mode is selected.

Value	Description
0	The Direction Change interrupt is disabled.
1	The Direction Change interrupt is enabled.

Bit 1 – ERR Error Interrupt Disable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Error Interrupt Disable/Enable bit, which disables the Error interrupt.

Value	Description
0	The Error interrupt is disabled.
1	The Error interrupt is enabled.

Bit 0 – OVF Overflow/Underflow Interrupt Disable

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

Writing a '1' to this bit will clear the Overflow Interrupt Disable/Enable bit, which disables the Overflow interrupt.

Value	Description
0	The Overflow interrupt is disabled.
1	The Overflow interrupt is enabled.

46.7.6. Interrupt Enable Set

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

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 Interrupt Enable Clear (INTENCLR) register.

Table 46-7. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
			MC1	MC0	VLC	DIR	ERR	OVF
Access			RW	RW	RW	RW	RW	RW
Reset			0	0	0	0	0	0

Bits 4, 5 – MCx Channel x Compare Match Enable [x = 1..0]
Writing a '0' to MCx has no effect.
Writing a '1' to MCx will set the corresponding Match Channel x Interrupt Disable/Enable bit, which enables the Match Channel x interrupt.

Value	Description
0	The Match Channel x interrupt is disabled.
1	The Match Channel x interrupt is enabled.

Bit 3 – VLC Velocity Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Velocity Interrupt Disable/Enable bit, which enables the Velocity interrupt.
This bit has no effect when COUNTER operation mode is selected.

Value	Description
0	The Velocity interrupt is disabled.
1	The Velocity interrupt is enabled.

Bit 2 – DIR Direction Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Direction Change Interrupt Disable/Enable bit, which enables the Direction Change interrupt.
This bit has no effect when COUNTER operation mode is selected.

Value	Description
0	The Direction Change interrupt is disabled.
1	The Direction Change 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 Disable/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/Underflow 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 enable the Overflow interrupt.

Value	Description
0	The Overflow interrupt is disabled.
1	The Overflow interrupt is enabled.

46.7.7. Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x0A
Reset: 0x00
Property: -

Table 46-8. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
			MC1	MC0	VLC	DIR	ERR	OVF
Access			RW	RW	RW	RW	RW	RW
Reset			0	0	0	0	0	0

Bits 4, 5 – MCx Channel x Compare Match [x = 1..0]
This flag is set on the next CLK_PDEC_CNT cycle after a match with the compare condition, and will generate an interrupt request if the corresponding Match 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 Channel x interrupt flag.

Bit 3 – VLC Velocity
This flag is set if a velocity transition occurs, and will generate an interrupt request if the Velocity Interrupt Enable bit in Interrupt Enable Set register (INTENSET.VLC) is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Velocity transition interrupt flag.
This flag is never set when COUNTER operation mode is selected.

Bit 2 – DIR Direction Change
This flag is set if a direction change occurs, and will generate an interrupt request if the Direction Change Interrupt Enable bit in Interrupt Enable Set register (INTENSET.DIR) is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Velocity transition interrupt flag.
This flag is never set when COUNTER operation mode is selected.

Bit 1 – ERR Error
This flag is set when an error condition is detected, and will generate an interrupt request if the Error Interrupt Enable bit in the Interrupt Enable Set register (INTENSET.ERR) is '1'. The error source can be identified by reading the Status (STATUS) register.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Error interrupt flag.

Bit 0 – OVF Overflow/Underflow
This flag is set on the next CLK_TC_CNT cycle after an overflow condition occurs, and will generate an interrupt request if the Overflow Interrupt Enable bit in the Interrupt Enable Set register (INTENSET.OVF) is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Overflow interrupt flag.

46.7.8. Status

Name: STATUS
Offset: 0x0C
Reset: 0x0040
Property: Read-Synchronized, Write-Synchronized

Note: This register is write-synchronized: SYNCBUSY.STATUS must be checked to ensure the STATUS register synchronization is complete.

Table 46-9. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	15	14	13	12	11	10	9	8
			CCBUFV1	CCBUFV0			FILTERBUFV	PRESCBUFV
Access			R	R			R	R
Reset			0	0			0	0

Bit	7	6	5	4	3	2	1	0
	DIR	STOP	HERR	WINERR		MPERR	IDXERR	QERR
Access	R	R	RW	RW		RW	RW	RW
Reset	0	1	0	0		0	0	0

Bits 12, 13 – CCBUFVx Compare Channel x Buffer Valid [x = 1..0]
The bit is set when a new value is written to the corresponding CCBUF register.
The bit is cleared by writing a '1' to the corresponding location or automatically cleared on an UPDATE condition.

Bit 9 – FILTERBUFV Filter Buffer Valid
This bit is set when a new value is written to the PRESCALERBUF register.
The bit is cleared by writing a '1' to the corresponding location or automatically cleared on an UPDATE condition.
This bit is always read '0' when COUNTER operation mode is selected.

Bit 8 – PRESCBUFV Prescaler Buffer Valid
This bit is set when a new value is written to the PRESC register.
The bit is cleared by writing a '1' to the corresponding location or automatically cleared on an UPDATE condition.

Bit 7 – DIR Direction Status Flag
This bit reflects the HALL/QDEC direction.
in COUNTER mode, this bits is always read '0'.

Value	Description
0	Clockwise direction.
1	Counter-clockwise direction.

Bit 6 – STOP Stop
This bit reflects the HALL/QDEC decoding status.
In COUNTER mode, this bits is always read '0'.

Value	Description
0	PDEC/HALL decoding is running.
1	PDEC/HALL decoding is stopped.

Bit 5 – HERR Hall Error Flag

This flag is set when an invalid HALL code is detected.
The flag is cleared by writing a '1' to this bit location. After write to the STATUS register, ensure the SYNCBUSY.STATUS is '0' before reading.
Outside of HALL mode, this bits is always read '0'.

Bit 4 – WINERR Window Error Flag

This flag is set when the counter is outside the window monitor.
The flag is cleared by writing a '1' to this bit location. After write to the STATUS register, ensure the SYNCBUSY.STATUS is '0' before reading.
Outside of HALL mode, this bits is always read '0'.

Bit 2 – MPERR Missing Pulse Error flag

This flag is set when a missing pulse error condition is detected.
The flag is cleared by writing a '1' to this bit location. After write to the STATUS register, ensure the SYNCBUSY.STATUS is '0' before reading.
Outside of QDEC mode, this bits is always read '0'.

Bit 1 – IDXERR Index Error Flag

This flag is set when an index error condition is detected.
The flag is cleared by writing a '1' to this bit location. After write to the STATUS register, ensure the SYNCBUSY.STATUS is '0' before reading.
Outside of QDEC mode, this bits is always read '0'.

Bit 0 – QERR Quadrature Error Flag

This flag is set when an invalid QDEC transition is detected.
The flag is cleared by writing a '1' to this bit location. After write to the STATUS register, ensure the SYNCBUSY.STATUS is '0' before reading.
Outside of QDEC mode, this bits is always read '0'.

46.7.9. Debug Control

Name: DBGCTRL
Offset: 0x0F
Reset: 0x00
Property: PAC Write-Protection

Table 46-10. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
								DBGRUN
Access								RW
Reset								0

Bit 0 – DBGRUN Debug Run Mode
This bit is not affected by software reset and should not be changed by software while the PDEC module is enabled.

Value	Description
0	The PDEC module is halted when the device is halted in debug mode.
1	The PDEC module continues normal operation when the device is halted in debug mode.

46.7.10. Synchronization Status

Name: SYNCBUSY
Offset: 0x10
Reset: 0x00000000
Property: Read-Only

Table 46-11. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
								CC1
Access								R
Reset								0

Bit	7	6	5	4	3	2	1	0
	CC0	COUNT	FILTER	PRESC	STATUS	CTRLB	ENABLE	SWRST
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 7, 8 – CC Compare Channel x Synchronization Busy
This bit is cleared when the synchronization of Compare Channel x (CCx) register between the clock domains is complete.
This bit is set when the synchronization of Compare Channel x (CCx) register between clock domains is started.

Bit 6 – 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 5 – FILTER Filter Synchronization Busy
This bit is cleared when the synchronization of Filter register between the clock domains is complete.
This bit is set when the synchronization of Filter register between clock domains is started.
This bit is always read '0' when COUNTER operation mode is selected.

Bit 4 – PRESC Prescaler Synchronization Busy
This bit is cleared when the synchronization of Prescaler register between the clock domains is complete.

This bit is set when the synchronization of Prescaler 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 Control B Synchronization Busy
This bit is cleared when the synchronization of Control B register between the clock domains is complete.
This bit is set when the synchronization of Control B register between clock domains is started.

Bit 1 – ENABLE Enable Synchronization Busy
This bit is cleared when the synchronization of Enable register bit between the clock domains is complete.
This bit is set when the synchronization of Enable register bit between clock domains is started.

Bit 0 – SWRST Software Reset Synchronization Busy
This bit is cleared when the synchronization of Software Reset register bit between the clock domains is complete.
This bit is set when the synchronization of Software Reset register bit between clock domains is started.

46.7.11. Prescaler Value

Name: PRESC
Offset: 0x14
Reset: 0x00
Property: Write-Synchronized

Note: This register is write-synchronized: SYNCBUSY.PRESC must be checked to ensure the PRESC register synchronization is complete.

Table 46-12. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
					PRESC[3:0]			
Access					RW	RW	RW	RW
Reset					0	0	0	0

Bits 3:0 – PRESC[3:0] Prescaler Value
These bits select the GCLK prescaler factor.

Value	Name	Description
0	DIV1	No division
1	DIV2	Divide by 2
2	DIV4	Divide by 4
3	DIV8	Divide by 8
4	DIV16	Divide by 16
5	DIV32	Divide by 32
6	DIV64	Divide by 64
7	DIV128	Divide by 128
8	DIV256	Divide by 256
9	DIV512	Divide by 512
10	DIV1024	Divide by 1024

46.7.12. Filter Value

Name: FILTER
Offset: 0x15
Reset: 0x00
Property: Write-Synchronized

Note: This register is write-synchronized: SYNCBUSY.FILTER must be checked to ensure the FILTER register synchronization is complete.

Table 46-13. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
	FILTER[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – FILTER[7:0] Filter Value
These bits select the PDEC inputs filter length. The input signal minimum duration will be (FILTER+1)*tGCLK_PDEC.
These bits have no effect when COUNTER operation mode is selected.

46.7.13. Prescaler Buffer Value

Name: PRESCBUF
Offset: 0x18
Reset: 0x00
Property: Write-Synchronized

Note: This register is write-synchronized: SYNCBUSY.PRESC must be checked to ensure the PRESC register synchronization is complete.

Table 46-14. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
					PRESCBUF[3:0]			
Access					RW	RW	RW	RW
Reset					0	0	0	0

Bits 3:0 – PRESCBUF[3:0] Prescaler Buffer Value
These bits hold the value of the prescaler buffer register. The value is copied in the corresponding PRESC register on UPDATE condition.

Value	Name	Description
0	DIV1	No division
1	DIV2	Divide by 2
2	DIV4	Divide by 4
3	DIV8	Divide by 8
4	DIV16	Divide by 16
5	DIV32	Divide by 32
6	DIV64	Divide by 64
7	DIV128	Divide by 128
8	DIV256	Divide by 256
9	DIV512	Divide by 512
10	DIV1024	Divide by 1024

46.7.14. Filter Buffer Value

Name: FILTERBUF
Offset: 0x19
Reset: 0x00
Property: Write-Synchronized

Note: This register is write-synchronized: SYNCBUSY.FILTER must be checked to ensure the FILTERBUF register synchronization is complete.

Table 46-15. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
	FILTERBUF[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – FILTERBUF[7:0] Filter Buffer Value
These bits hold the value of the filter buffer register. The value is copied in the corresponding FILTER register on UPDATE condition.
These bits have no effect when COUNTER operation mode is selected.

46.7.15. Counter Value

Name: COUNT
Offset: 0x1C
Reset: 0x00000000
Property: PAC Write-Protection, Read-Synchronized, Write-Synchronized

Note: This register is write-synchronized: SYNCBUSY.COUNT must be checked to ensure the COUNT register synchronization is complete.

Table 46-16. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
	COUNT[15:8]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	COUNT[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – COUNT[15:0] Counter Value
These bits contain the counter value. To read the most updated counter value, the READSYNC software command must be applied first (CTRLBSET.CMD = READSYNC).

46.7.16. Channel x Compare Value

Name: CCx
Offset: 0x20 + x*0x04 [x=0..1]
Reset: 0x00000000
Property: Read-Synchronized, Write-Synchronized

Note: This register is write-synchronized: SYNCBUSY.CCx must be checked to ensure the CCx register synchronization is complete.

Table 46-17. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
	CC[15:8]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CC[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – CC[15:0] Channel Compare Value
These bits hold value of the channel x compare register.

46.7.17. Channel x Compare Buffer Value

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 CCBUFx register synchronization is complete.

Table 46-18. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
	CCBUF[15:8]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CCBUF[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – CCBUF[15:0] Channel Compare Buffer Value
These bits hold the value of the channel x compare buffer register. The register is used as buffer for the associated compare register (CCx). Accessing this register using the CPU will affect the corresponding CCBVx status bit (STATUS.CCBUFVx).

47. Parallel Capture Controller (PCC)

47.1. Overview

The Parallel Capture Controller can be used to interface an external system, such as a CMOS digital image sensor, ADC, or DSP, and capture its parallel data.

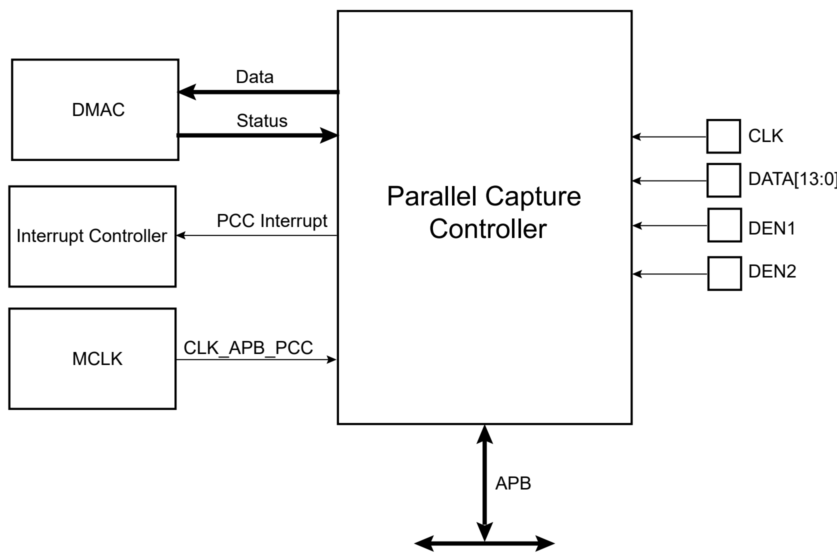
47.2. Features

The following are key features of the PCC module:

- One clock, up to 14-bit parallel data, and two Data Enable on I/O lines
- Data can be sampled every other time (for example, for chrominance sampling)
- Supports connection of the DMAC, which offers buffer reception without processor intervention
- Auto-scale feature available when 10, 12, or 14 bits data size is selected.
- Can be used to interface a CMOS Digital Image Sensor, an ADC and so on.

47.3. Block Diagram

Figure 47-1. Block Diagram



47.4. Signal Description

Signal	Description	Type
CLK	Digital input	PCC Clock
DATA[13:0]	Digital input	Data [13:0]
DEN1	Digital input	Data Enable 1
DEN2	Digital input	Data Enable 2

47.5. Peripheral Dependencies

Peripheral Name	Base Address	NVIC IRQ Index: Source	MCLK AHBx/APBx Clock Enable Mask Bit	PAC Peripheral Identifier (PAC.WRCTRL.PERIDx)	EVSYS Generator (EVSYS.CHANNELn.EVGENx)	DMA Trigger Index:Source (DMAC.CHCTRLBk.TRIGx)	Power Domain
PCC	0x4501_8000	141 : PCC	MCLK.CLKMSK3[12]	48	113 : GEN	67 : RX	VDDREG

47.5.1. Power Management

The PCC will continue to operate in any Idle Sleep mode,

47.5.2. Clocks

The PCC bus clock (CLK_APB_PCC) is provided by the Main Clock (MCLK) through the AHB-APB C bridge. The clock is enabled and disabled by writing the MSK12 bit the in the MCLK.CLKMSK3 register (MCLK.CLKMSK3[12]).

For capturing operation, the external device has to provide a PCC clock signal (PCC_CLK) synchronous to the data received (pixel clock) through a pin. Refer to the PORT and the Pinout table for additional information.

Writing any of the registers does not require the PCC_CLK to be enabled.

Important: The CLK_APB_PCC clock frequency must be at least twice the PCC_CLK frequency.

47.5.3. DMA

The DMAC can be configured to use the RX channel of the PCC as trigger source.

If configured, a trigger signal is send to the DMAC when data is received by the PCC, such that the DMAC will automatically read the received data buffer. The buffer ready signal will be automatically clear upon the read done by the DMAC.

47.5.4. Debug Operation

When the CPU is halted in debug mode, the PCC will not halt normal operation.
Note: A buffer overflow condition will occur if the received data buffer is not read by CPU or CPU DMAC.

47.5.5. Register Access Protection

To prevent any single software error from corrupting PCC behavior, certain registers in the address space can be write-protected by setting the WPEN bit in the Write Protection Mode Register (WPMR).

If a write access to a write-protected register is detected, the WPVS flag in the Write Protection Status Register (WPSR) is set and WPSR.WPVSRC indicates the register in which the write access has been attempted.

The WPVS bit is automatically cleared after reading WPSR.

The following registers can be write-protected:

- PCC Mode Register

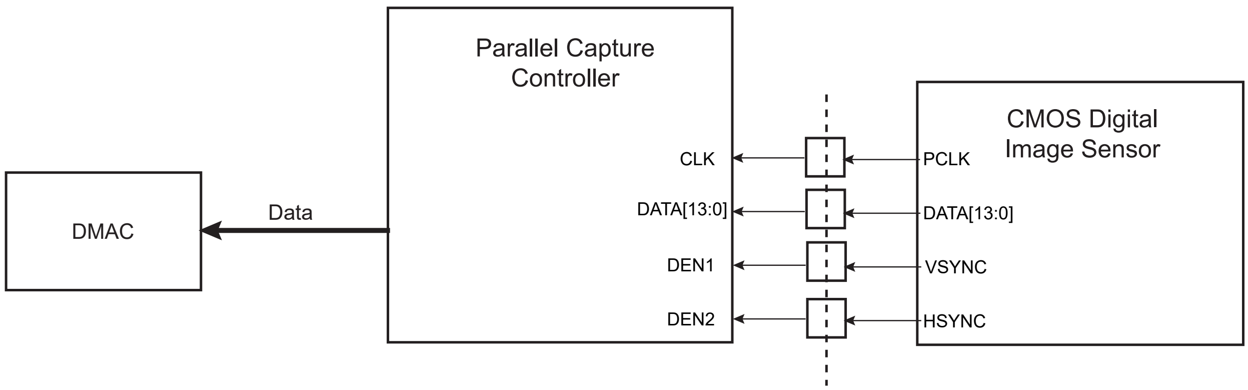
47.6. Functional Description

47.6.1. Principle of Operation

For better understanding and to ease reading, the following description uses an example with a CMOS digital image sensor.

The CMOS digital image sensor provides a sensor clock, an 10-bit data synchronous with the sensor clock and two data enables which are also synchronous with the sensor clock.

Figure 47-2. Parallel Capture Controller Connection with CMOS Digital Image Sensor



The PCC must be configured first, and is enabled by writing a '1' to the Parallel Capture Enable bit in the Mode Register (MR.PCEN).

Once enabled, the PCC samples the data at rising edge of the sensor clock, and resynchronizes it with the PCC clock domain.

The input data bus size can be programmed using the Input Data Size bit field (MR.ISIZE).

A re-initialization of the internal mechanism of the PCC can be automatically done by setting the CID register when a falling edge of the DEN1 or DEN2 is detected. This feature allows glitch filtering and prevents image de-synchronization.

The number of the data which can be read in the Reception Holding Register (RHR) can be programmed by writing the Data Size bit field (MR.DSIZE). The PCC samples one or several sensor data, according to the DSIZE value.

If the MR.SCALE bit is written to '1' and MR.ISIZE ≠ 0, the sampled data is automatically up-scaled to 16 bits. When the right number of data has be sampled, data are stored in the RHR, and the Data Ready flag in the Interrupt Status Register (ISR.DRDY) is set to '1'.

The PCC can be associated with a reception channel of the DMA Controller (DMAC). This performs reception transfer from the PCC to a memory buffer without any intervention from the CPU.

The PCC can be configured to either comply with the sensor data enable signals, or not. If the Always Sampling bit in the Mode Register (MR.ALWYS) is written to '0', the PCC samples the sensor data at the rising edge of the sensor clock only if both data enable signals are active (at '1'). If ALWYS is written to '1', the PCC samples the sensor data at the rising edge of the sensor clock, independent of the data enable signals.

The PCC can be configured to sample the sensor data only every other time. This is particularly useful when only the luminance Y from a YUV422 data stream of a CMOS digital image sensor is to be sampled. If the Half Sampling bit in the Mode Register (MR.HALFS) is written to '0', the PCC samples the sensor data as configured above. If MR.HALFS=1, the PCC samples the sensor data as configured above (i.e. respecting the MR.ALWYS setting), but only one time out of two.

The PCC can either sample the even or odd sensor data, depending on the First Sample bit (MR.FRSTS). If sensor data are numbered with an index from zero to n in the order they are received and FRSTS=0, only data with an even index are sampled. For FRSTS=1, only data with an odd index are sampled.

If data are ready in the Reception Holding Register (RHR) but it is not read before new data is stored in RHR, an overrun error occurs: The previous data is lost and the Overrun Error flag in the Interrupt Status Register (ISR.OVRE) is set. This flag is automatically cleared when ISR is read (reset after read).

The flags DRDY and OVRE can be a source of the PCC interrupt.

Figure 47-3. PCC Waveforms (DSIZE=4_DATA, ALWYS = 0, HALFS = 0)

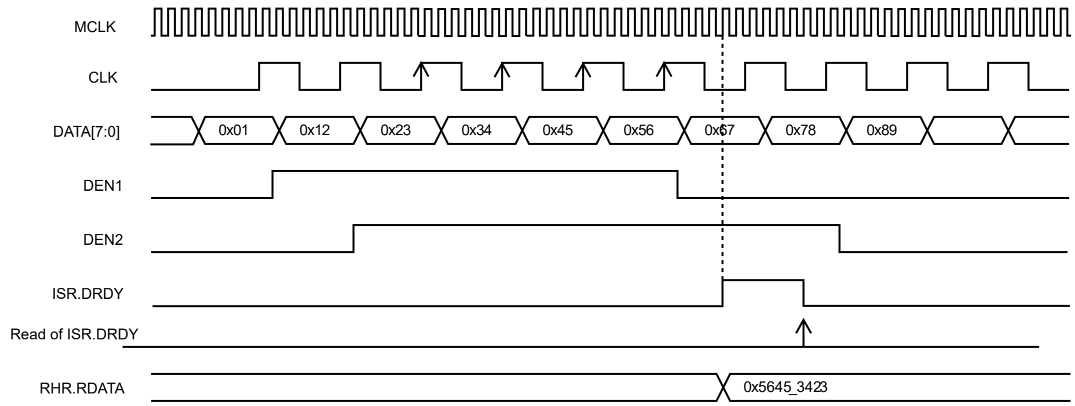


Figure 47-4. PCC Waveforms (ISIZE=10_BITS, DSIZE=2_DATA, ALWYS = 0, HALFS = 0, SCALE = 0)

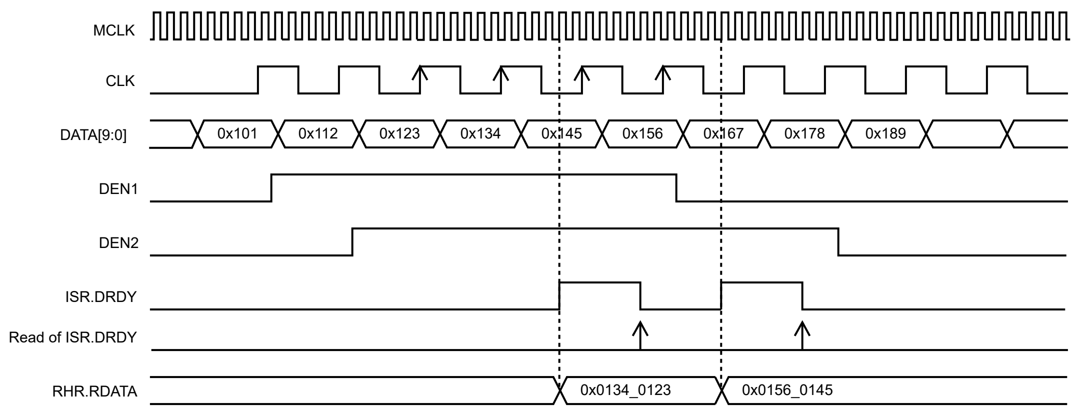


Figure 47-5. PCC Waveforms (ISIZE=10_BITS, DSIZE=2_DATA, ALWYS = 0, HALFS = 0, SCALE = 1)

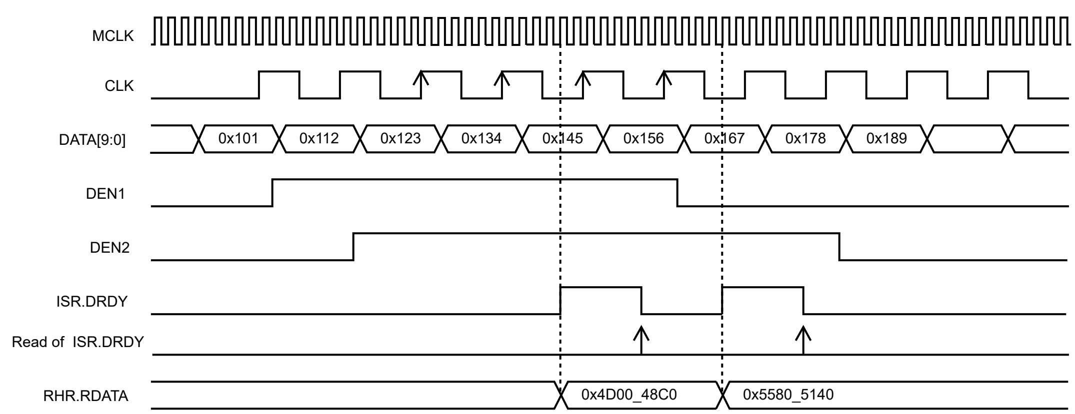


Figure 47-6. PCC Waveforms (DSIZE=4_DATA, ALWYS = 1, HALFS = 0)

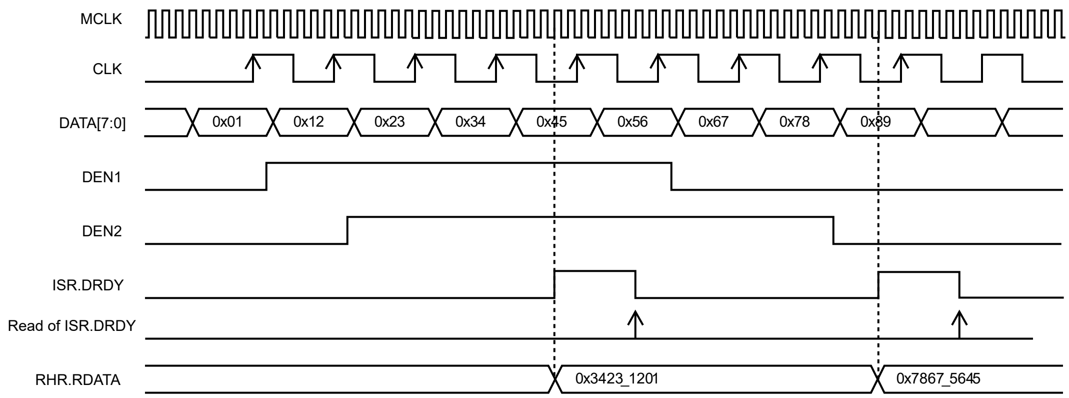


Figure 47-7. PCC Waveforms (ISIZE=10_BITS, DSIZE=2_DATA, ALWYS = 1, HALFS = 0, SCALE = 0)

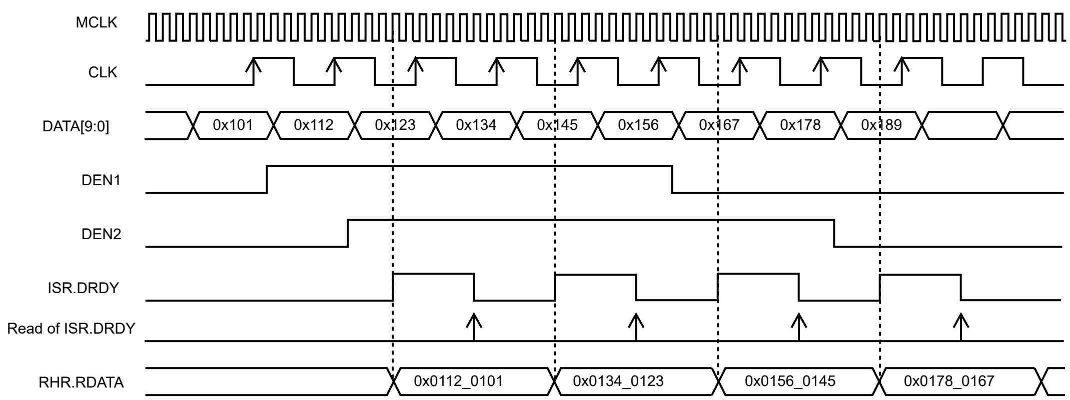


Figure 47-8. PCC Waveforms (ISIZE=10_BITS, DSIZE=2_DATA, ALWYS = 1, HALFS = 0, SCALE = 1)

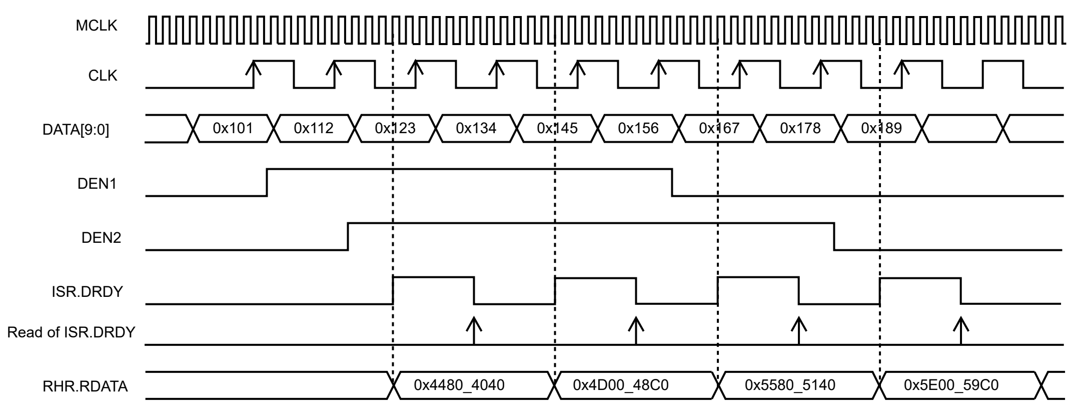


Figure 47-9. PCC Waveforms (DSIZE=4_DATA, ALWYS = 0, HALFS = 1, FRSTS = 0)

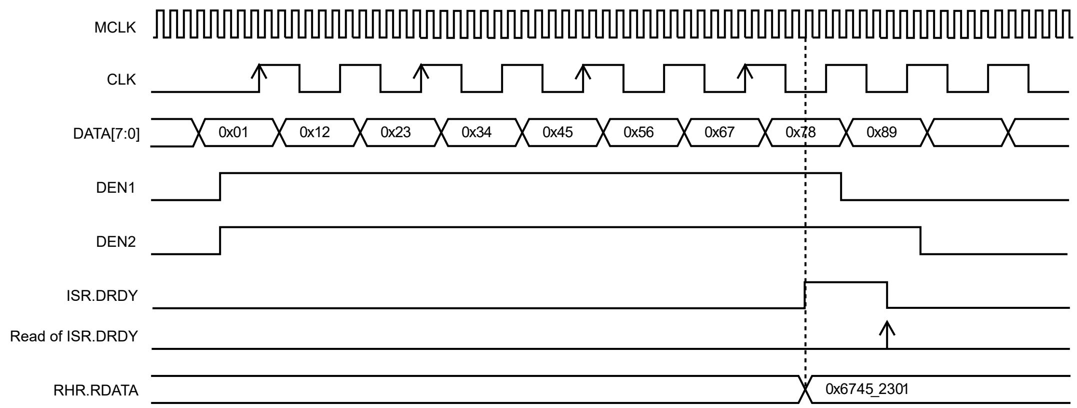


Figure 47-10. PCC Waveforms (ISIZE=10_BITS, DSIZE=2_DATA, ALWYS = 0, HALFS = 1, FRSTS = 0, SCALE = 0)

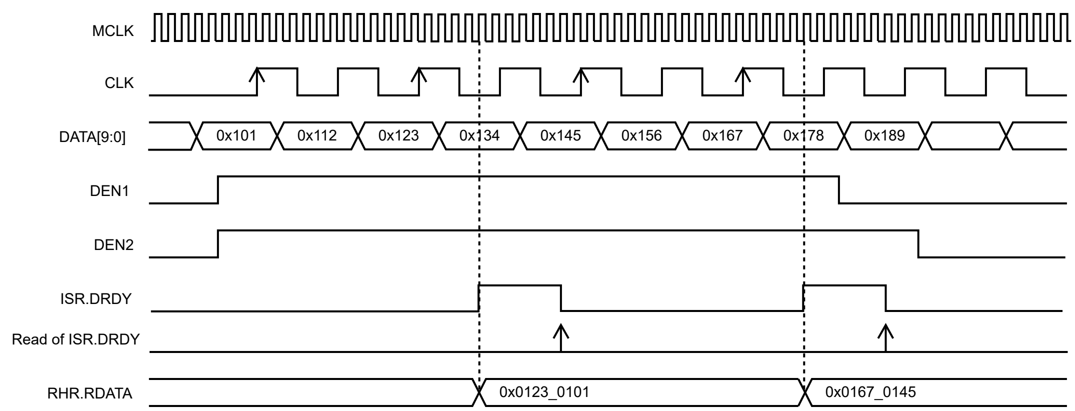


Figure 47-11. PCC Waveforms (ISIZE=10_BITS, DSIZE=2_DATA, ALWYS = 0, HALFS = 1, FRSTS = 0, SCALE = 1)

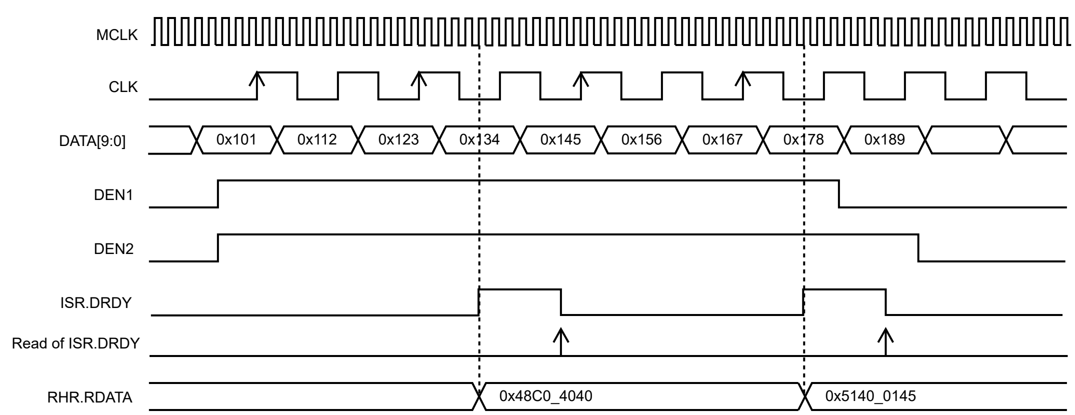


Figure 47-12. PCC Waveforms (DSIZE=4_DATA, ALWYS = 0, HALFS = 1, FRSTS = 1)

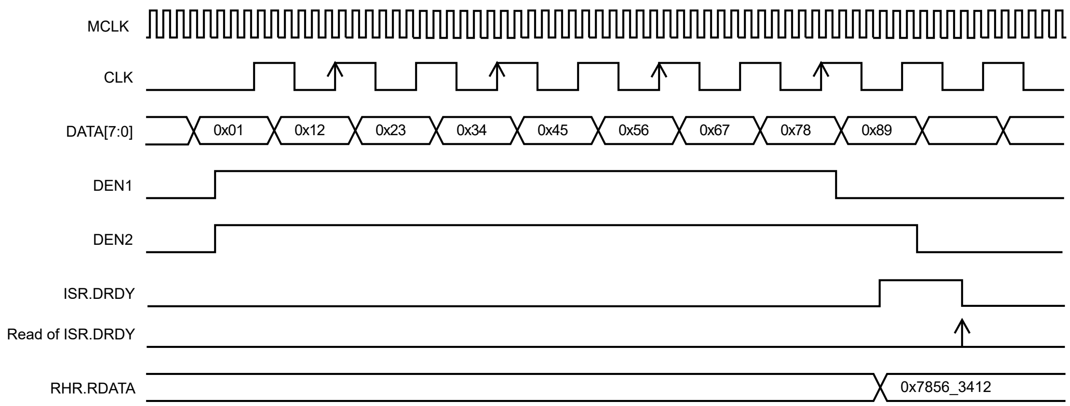


Figure 47-13. PCC Waveforms (ISIZE=10_BITS, DSIZE=2_DATA, ALWYS = 0, HALFS = 1, FRSTS = 1, SCALE = 0)

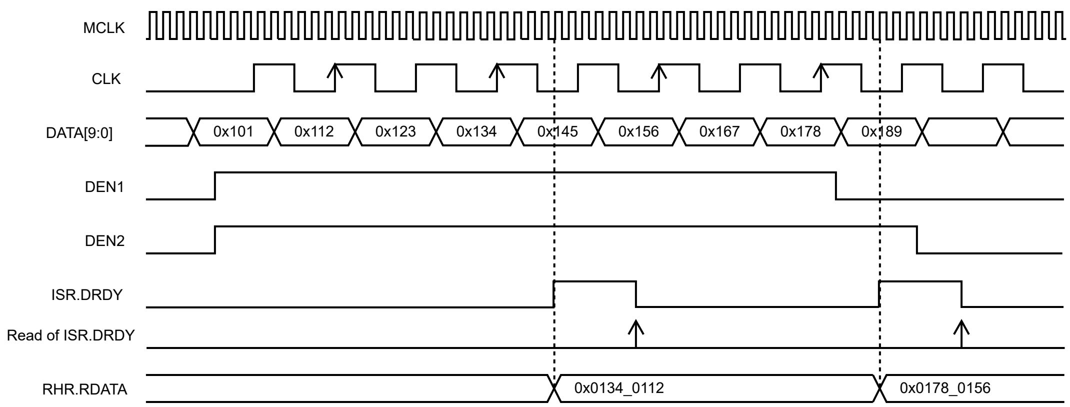
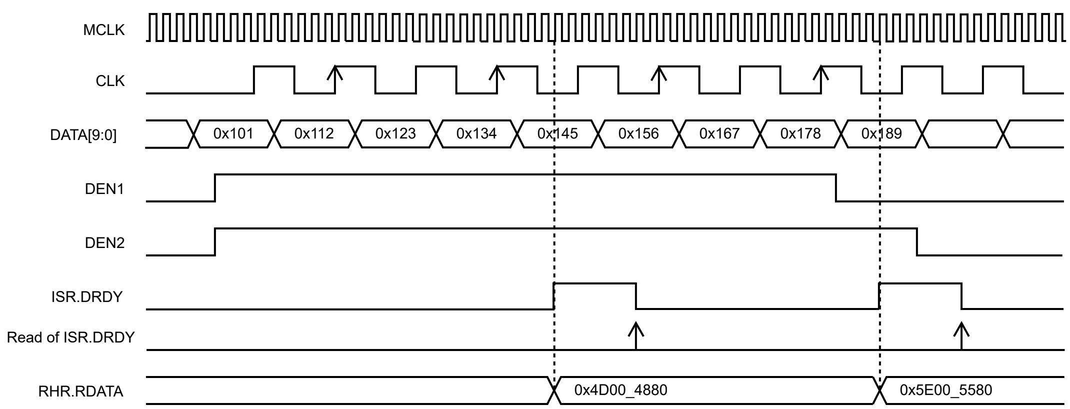


Figure 47-14. PCC Waveforms (ISIZE=10_BITS, DSIZE=2_DATA, ALWYS = 0, HALFS = 1, FRSTS = 1, SCALE = 1)



47.6.2. Register Access Protection

The configuration bit fields ISIZE, SCALE, DSIZE, ALWYS, HALFS and FRSTS in the Mode Register (MR) can be changed ONLY if the PCC is disabled at this time (MR.PCEN=0).

47.6.3. Programming Sequence

47.6.3.1. Without DMAC

1. Write the Interrupt Enable and Interrupt Disable Registers (IER and IDR) in order to configure the PCC interrupt mask.

2. Write the Mode Register (MR) fields ISIZE, SCALE, DSIZE, ALWYS, HALFS and FRSTS in order to configure the PCC. Do not enable the PCC in this write access.
3. Write the PCC Enable bit in the Mode Register (MR.PCEN) to '1' in order to enable the PCC. Do not change the configuration from the previous step.
4. Wait for a Data Ready, either by polling the Data Ready flag in the Interrupt Status Register (ISR.DRDY) or by waiting for the corresponding interrupt.
5. Check the Overrun Error flag (ISR.OVRE).
6. Read the data in the Reception Holding Register (RHR).
7. If new data are expected, go to step 4.
8. Disable the PCC by writing MR.PCEN to '0' *without* changing the configuration.

47.6.3.2. With DMAC

1. Write the Interrupt Enable and Interrupt Disable Registers (IER and IDR) in order to configure the PCC interrupt mask.
2. Configure DMAC transfer in the DMAC registers.
3. Write the Mode Register (MR) fields ISIZE, SCALE, DSIZE, ALWYS, HALFS and FRSTS in order to configure the PCC. Do not enable the PCC in this write access.
4. Write the PCC Enable bit in the Mode Register (MR.PCEN) to '1' in order to enable the PCC. Do not change the configuration from the previous step.
5. Wait for the DMA status flag to indicate that the buffer transfer is complete.
6. Check the Overrun Error flag (ISR.OVRE).
7. If a new buffer transfer is expected, go to step 5.
8. Disable the PCC by writing MR.PCEN to '0' *without* changing the configuration.

47.7. Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	MR	31:24	CID[1:0]							
		23:16						ISIZE[2:0]		
		15:8					FRSTS	HALFS	ALWYS	SCALE
		7:0			DSIZE[1:0]					PCEN
0x04	IER	31:24								
		23:16								
		15:8								
		7:0							OVRE	DRDY
0x08	IDR	31:24								
		23:16								
		15:8								
		7:0							OVRE	DRDY
0x0C	IMR	31:24								
		23:16								
		15:8								
		7:0							OVRE	DRDY
0x10	ISR	31:24								
		23:16								
		15:8								
		7:0							OVRE	DRDY
0x14	RHR	31:24	RDATA[31:24]							
		23:16	RDATA[23:16]							
		15:8	RDATA[15:8]							
		7:0	RDATA[7:0]							
0x18 ... 0xDF	Reserved									
0xE0	WPMR	31:24	WPKEY[23:16]							
		23:16	WPKEY[15:8]							
		15:8	WPKEY[7:0]							
		7:0								WPEN
0xE4	WPSR	31:24								
		23:16	WPVSRC[15:8]							
		15:8	WPVSRC[7:0]							
		7:0								WPVS

47.7.1. PCC Mode Register

Name: MR
Offset: 0x00
Reset: 0x00000000
Property: -

This register can only be written if the WPEN bit is cleared in the Write Protection Mode Register.

Table 47-1. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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

Bit	23	22	21	20	19	18	17	16
						ISIZE[2:0]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	15	14	13	12	11	10	9	8
					FRSTS	HALFS	ALWYS	SCALE
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

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

Bits 31:30 – CID[1:0] Clear If Disabled

Clears status flags if disabled. These bits are useful to re-initialize the internal mechanism of the PCC to avoid corrupted data due to glitches. Each time a falling edge of the selected DEN1 or DEN2 signal is detected, the internal mechanism of the PCC is re-initialized to avoid alignment issues.

Value	Description
`0x0`	Clear not enabled
`0x1`	Clear on falling edge on DEN1 enabled
`0x2`	Clear on falling edge on DEN2 enabled
`0x3`	Clear on falling edge on either DEN1 or DEN2 enabled

Bits 18:16 – ISIZE[2:0] Input Data Size

Value	Name	Description
`0x0`	8BITS	Input data bus size is 8 bits
`0x1`	10BITS	Input data bus size is 10 bits
`0x2`	12BITS	Input data bus size is 12 bits
`0x3`	14BITS	Input data bus size is 14 bits

Bit 11 – FRSTS First Sample

This bit is useful only if the HALFS bit is set to 1. If data are numbered in the order that they are received with an index from 0 to n.

Value	Description
0	Only data with an even index are sampled.
1	Only data with an odd index are sampled.

Bit 10 – HALFS Half Sampling

This function is independent from the ALWYS bit.

Value	Description
0	The Parallel Capture Controller samples all the data.
1	The Parallel Capture Controller samples the data only every other time.

Bit 9 – ALWYS Always Sampling

Value	Description
0	The parallel capture Controller samples the data when both data enables are active.
1	The parallel capture controller always samples the data, regardless of the state of data enable.

Bit 8 – SCALE Scale Data

Value	Description
0	No effect.
1	When input data size is not equal to 8 bits (ISIZE ≠ 0), the data stored in the PCC_RHR is automatically up-scaled to 16 bits.

Bits 5:4 – DSIZE[1:0] Data Size

Value	Name	Description
0x0	1DATA	1 data is read in the PCC_RHR
0x1	2DATA	2 data are read in the PCC_RHR
0x2	4DATA	4 data are read in the PCC_RHR (only for 8 bits data size, ISIZE = 0)
0x3	-	Reserved

Bit 0 – PCEN Parallel Capture Enable

Value	Description
0	The Parallel Capture Controller is disabled.
1	The Parallel Capture Controller is enabled.

47.7.2. Interrupt Enable Register

Name: IER
Offset: 0x04
Reset: 0x00000000
Property: -

Table 47-2. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
							OVRE	DRDY
Access							W	W
Reset							0	0

Bit 1 – OVRE Overrun Error Interrupt Enable
Writing a '1' to this register enables the Overrun Error interrupt.
Writing a '0' has no effect.

Bit 0 – DRDY Data Ready Interrupt Enable
Writing a '1' to this register enables the Data Ready Interrupt interrupt.
Writing a '0' has no effect.

47.7.3. Interrupt Disable Register

Name: IDR
Offset: 0x08
Reset: 0x00000000
Property: -

Table 47-3. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
							OVRE	DRDY
Access							W	W
Reset							0	0

Bit 1 – OVRE Overrun Error Interrupt Disable
Writing a '1' to this register disables the Overrun Error interrupt.
Writing a '0' has no effect.

Bit 0 – DRDY Data Ready Interrupt Disable
Writing a '1' to this register disables the Data Ready interrupt.
Writing a '0' has no effect.

47.7.4. Interrupt Mask Register

Name: IMR
Offset: 0x0C
Reset: 0x00000000
Property: -

Table 47-4. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

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

Bit 1 – OVRE Overrun Error Interrupt Mask

Value	Description
1	The Overrun Error interrupt is enabled.
0	The Overrun Error interrupt is not enabled.

Bit 0 – DRDY Data Ready Interrupt Mask

Value	Description
1	The Data Ready interrupt is enabled.
0	The Data Ready interrupt is not enabled.

47.7.5. Interrupt Status Register

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

Table 47-5. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

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

Bit 1 – OVRE Overrun Error Interrupt Status
The OVRE flag is automatically reset when this register is read or when the PCC is disabled.

Value	Description
0	No overrun error occurred since the last read of this register.
1	At least one overrun error occurred since the last read of this register.

Bit 0 – DRDY Data Ready Interrupt Status
The DRDY flag is automatically reset when RHR is read or when the PCC is disabled.

Value	Description
0	No new data is ready to be read since the last read of RHR.
1	New data is ready to be read since the last read of RHR.

47.7.6. Reception Holding Register

Name: RHR
Offset: 0x14
Reset: 0x00000000
Property: -

Table 47-6. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	RDATA[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
	RDATA[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
	RDATA[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
	RDATA[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – RDATA[31:0] Reception Data

ISIZE	SCALE	DSIZE	Description
8_BITS	-	1_DATA	RDATA[7:0] is useful
		2_DATA	RDATA[15:0] is useful
		4_DATA	RDATA[31:0] is useful
10_BITS	0 (OFF)	1_DATA	RDATA[9:0] is useful
		2_DATA	RDATA[9:0] and RDATA[25:16] are useful
	1 (ON)	1_DATA	RDATA[15:0] is useful
		2_DATA	RDATA[31:0] is useful
12_BITS	0 (OFF)	1_DATA	RDATA[11:0] is useful
		2_DATA	RDATA[11:0] and RDATA[27:16] are useful
	1 (ON)	1_DATA	RDATA[15:0] is useful
		2_DATA	RDATA[31:0] is useful
14_BITS	0 (OFF)	1_DATA	RDATA[13:0] is useful
		2_DATA	RDATA[13:0] and RDATA[29:16] are useful
	1 (ON)	1_DATA	RDATA[15:0] is useful
		2_DATA	RDATA[31:0] is useful

47.7.7. Write Protection Mode Register

Name: WPMR
Offset: 0xE0
Reset: 0x00000000
Property: -

Table 47-7. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	WPKEY[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	WPKEY[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WPKEY[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
								WPEN
Access								R/W
Reset								0

Bits 31:8 – WPKEY[23:0] Write Protection Key

Value	Name	Description
0x504343	PASSWD	Writing any other value in this field aborts the write operation of the WPEN bit. Always reads as 0.

Bit 0 – WPEN Write Protection Enable

Value	Description
0	Disables the write protection if WPKEY corresponds to 0x504343 (“PCC” in ASCII).
1	Enables the write protection if WPKEY corresponds to 0x504343 (“PCC” in ASCII).

47.7.8. Write Protection Status Register

Name: WPSR
Offset: 0xE4
Reset: 0x00000000
Property: -

Table 47-8. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	WPVSRC[15:8]							
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
	WPVSRC[7:0]							
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
								WPVS
Access								R
Reset								0

Bits 23:8 – WPVSRC[15:0] Write Protection Violation Source
When WPVS = 1, WPVSRC indicates the register address offset at which a write access has been attempted.

Bit 0 – WPVS Write Protection Violation Status

Value	Description
0	No write protection violation has occurred since the last read of the WPSR.
1	A write protection violation has occurred since the last read of the WPSR. If this violation is an unauthorized attempt to write a protected register, the associated violation is reported into field WPVSRC.

48. Timer/Counter for Control Applications (TCC)

48.1. Overview

The PIC32CK SG/GC family of devices contain up to eight instances of the Timer/Counter for Control applications (TCC) modules: TCC0 to TCC3 are 32-bit timers and TCC4 to TCC7 are 16-bit timers.

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 (PWM).

TCC contains waveform extension features for motor control, ballast control, LED control, H-bridge control, power converters, and other types of power control applications. These extensions 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, deactivating output signals from the microcontroller. These signals can be used to shutdown external drivers connected to the microcontroller.

Note: The number of TCC instances, TCC channel numbers and TCC extensions are device specific. The TCC configurations, such as channel numbers and features, may be reduced for some of the TCC instances.

References:

- Device Configuration Summary

48.2. Features

The following are key features of the module:

- Up to six Compare/Capture Channels (CC) with:
  - Double buffered period setting
  - Double buffered compare or capture channel
  - Circular buffer on period and compare channel registers
- Input Capture:
  - Event capture
  - Frequency capture
  - Pulse-width capture
- Waveform Generation:
  - Frequency generation
  - Single-slope pulse-width modulation (PWM)
  - Dual-slope PWM with half-cycle reload capability
- 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 (TCCx_EV_0 , TCCx_EV_1) for counter
  - One input event (TCCx_MC_y) for each channel (y=0,1,2...7)
- Output Events:
  - Three output events (Count, re-trigger and overflow) are available for counter
  - One compare match/input capture event output for each channel
- Interrupts:
  - Overflow and re-trigger interrupt
  - Compare match/input capture interrupt
  - Interrupt on fault detection
- Can be used with DMA and can Trigger DMA Transactions

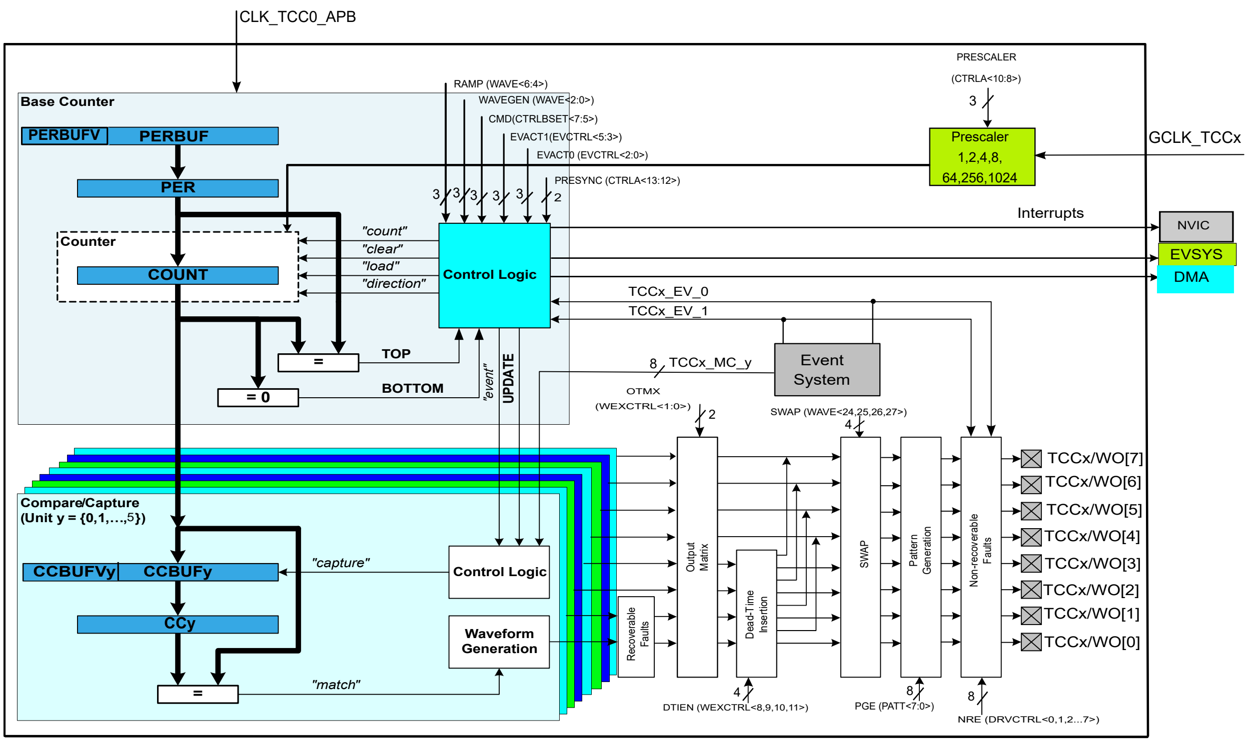
48.2.1. Feature Summary

TCC Instance	Capture-Compare **Channels**	Waveform Output **Channel Pins**	Output Channels that **Support Dithering**	(OTMX) **Output Matrix Channels Available**	(DTI) **Output Channels that Support Dead Time Insertion**	(SWAP) **Outputs Channels that Support HS / LS swapping**	(PG) **Output Channels that Support Pattern Generation**	Counter Size
0	6	8	1	1	1	1	1	32-bit
1	6	8	1	1	1	1	1	32-bit
2	6	8	1	1	1	1	1	32-bit
3	6	8	1	1	1	1	1	32-bit
4	2	2	0	0	0	0	0	16-bit
5	2	2	0	0	0	0	0	16-bit
6	2	2	0	0	0	0	0	16-bit
7	2	2	0	0	0	0	0	16-bit

48.3. Block Diagram

The following block diagram illustrates the Timer/Counter for Control Applications:

Figure 48-1. Timer/Counter for Control Applications Block Diagram



48.4. Signal Description

PIN NAME (1,2)	SIGNAL TYPE	Description
TCC0_WO0	Digital I/O	TCC0 channel 0, waveform output
TCC0_WO1	Digital I/O	TCC0 channel 1, waveform output
TCC0_WO2	Digital I/O	TCC0 channel 2, waveform output
TCC0_WO3	Digital I/O	TCC0 channel 3, waveform output
TCC0_WO4	Digital I/O	TCC0 channel 4, waveform output
TCC0_WO5	Digital I/O	TCC0 channel 5, waveform output
TCC0_WO6	Digital I/O	TCC0 channel 6, waveform output
TCC0_WO7	Digital I/O	TCC0 channel 7, waveform output
TCC1_WO0	Digital I/O	TCC1 channel 0, waveform output
TCC1_WO1	Digital I/O	TCC1 channel 1, waveform output
TCC1_WO2	Digital I/O	TCC1 channel 2, waveform output
TCC1_WO3	Digital I/O	TCC1 channel 3, waveform output
TCC1_WO4	Digital I/O	TCC1 channel 4, waveform output
TCC1_WO5	Digital I/O	TCC1 channel 5, waveform output
TCC1_WO6	Digital I/O	TCC1 channel 6, waveform output
TCC1_WO7	Digital I/O	TCC1 channel 7, waveform output
TCC2_WO0	Digital I/O	TCC2 channel 0, waveform output
TCC2_WO1	Digital I/O	TCC2 channel 1, waveform output
TCC2_WO2	Digital I/O	TCC2 channel 2, waveform output
TCC2_WO3	Digital I/O	TCC2 channel 3, waveform output
TCC2_WO4	Digital I/O	TCC2 channel 4, waveform output
TCC2_WO5	Digital I/O	TCC2 channel 5, waveform output

Signal Description (continued)

PIN NAME[1,2]	SIGNAL TYPE	Description
TCC2_WO6	Digital I/O	TCC2 channel 6, waveform output
TCC2_WO7	Digital I/O	TCC2 channel 7, waveform output
TCC3_WO0	Digital I/O	TCC3 channel 0, waveform output
TCC3_WO1	Digital I/O	TCC3 channel 1, waveform output
TCC3_WO2	Digital I/O	TCC3 channel 2, waveform output
TCC3_WO3	Digital I/O	TCC3 channel 3, waveform output
TCC3_WO4	Digital I/O	TCC3 channel 4, waveform output
TCC3_WO5	Digital I/O	TCC3 channel 5, waveform output
TCC3_WO6	Digital I/O	TCC3 channel 6, waveform output
TCC3_WO7	Digital I/O	TCC3 channel 7, waveform output
TCC4_WO0	Digital I/O	TCC4 channel 0, waveform output
TCC4_WO1	Digital I/O	TCC4 channel 1, waveform output
TCC5_WO0	Digital I/O	TCC5 channel 0, waveform output
TCC5_WO1	Digital I/O	TCC5 channel 1, waveform output
TCC6_WO0	Digital I/O	TCC6 channel 0, waveform output
TCC6_WO1	Digital I/O	TCC6 channel 1, waveform output
TCC7_WO0	Digital I/O	TCC7 channel 0, waveform output
TCC7_WO1	Digital I/O	TCC7 channel 1, waveform output

Notes:
1. Each Capture Compare (CC) channel enabled with Dead Time Insertion (DTI) extension will require 2 TCC output waveforms.
2. Refer to the *Pinout* chapter for additional information on the pin mapping for this peripheral. One signal can be mapped on several pins.

48.5. Peripheral Dependencies

Peripheral Name	Base Address	NVIC IRQ Index: Source	MCLK AHBx/APBx Clock Enable Mask Bit	GCLK Peripheral Channel Clock Name:Register	PAC Peripheral Identifier (PAC.WRCTRL.PERIDx)	EVSYS Users (EVSYS.USERm)	EVSYS Generator (EVSYS.CHANNELn.EVGENx)	DMA Trigger Index:Source (DMAC.CHCTRLBk.TRIGx)	Power Domain
TCC0	0x4481_8000	68 : CNT, DFS, ERR, FAULTA, FAULTB, FAULT0, FAULT1, OVF, TRG, UFS 69 : MC0 70 : MC1 71 : MC2 72 : MC3 73 : MC4 74 : MC5	MCLK.CLKMSK2[11]	GCLK_TCC0 : GCLK.PCHCTRL[23]	32	30 : EV0 31 : EV1 32 : MC0 33 : MC1 34 : MC2 35 : MC3 36 : MC4 37 : MC5	50 : OVF 51 : TRG 52 : CNT 53 : MC0 54 : MC1 55 : MC2 56 : MC3 57 : MC4 58 : MC5	13 : OVF 14 : MC0 15 : MC1 16 : MC2 17 : MC3 18 : MC4 19 : MC5	VDDCORE_SW

Peripheral Dependencies (continued)

Peripheral Name	Base Address	NVIC IRQ Index: Source	MCLK AHBx/APBx Clock Enable Mask Bit	GCLK Peripheral Channel Clock Name:Register	PAC Peripheral Identifier (PAC.WRCTRL.PERIDx)	EVSYS Users (EVSYS.USERm)	EVSYS Generator (EVSYS.CHANNELn.EVGENx)	DMA Trigger Index:Source (DMAC.CHCTRLBk.TRIGx)	Power Domain
TCC1	0x4481_A000	75 : CNT, DFS, ERR, FAULTA, FAULTB, FAULT0, FAULT1, OVF, TRG, UFS 76 : MC0 77 : MC1 78 : MC2 79 : MC3 80 : MC4 81 : MC5	MCLK.CLKMSK2[12]	GCLK_TCC1 : GCLK.PCHCTRL[23]	33	38 : EV0 39 : EV1 40 : MC0 41 : MC1 42 : MC2 43 : MC3 44 : MC4 45 : MC5	59 : OVF 60 : TRG 61 : CNT 62 : MC0 63 : MC1 64 : MC2 65 : MC3 66 : MC4 67 : MC5	20 : OVF 21 : MC0 22 : MC1 23 : MC2 24 : MC3 25 : MC4 26 : MC5	VDDCORE_SW
TCC2	0x4481_C000	82 : CNT, DFS, ERR, FAULTA, FAULTB, FAULT0, FAULT1, OVF, TRG, UFS 83 : MC_0 84 : MC_1 85 : MC_2 86 : MC_3 87 : MC_4 88 : MC_5	MCLK.CLKMSK2[13]	GCLK_TCC2 : GCLK.PCHCTRL[24]	34	46 : EV0 47 : EV1 48 : MC0 49 : MC1 50 : MC2 51 : MC3 52 : MC4 53 : MC5	68 : OVF 69 : TRG 70 : CNT 71 : MC0 72 : MC1 73 : MC2 74 : MC3 75 : MC4 76 : MC5	27 : OVF 28 : MC0 29 : MC1 30 : MC2 31 : MC3 32 : MC4 33 : MC5	VDDCORE_SW
TCC3	0x4481_E000	89 : CNT, DFS, ERR, FAULTA, FAULTB, FAULT0, FAULT1, OVF, TRG, UFS 90 : MC0 91 : MC1 92 : MC2 93 : MC3 94 : MC4 95 : MC5	MCLK.CLKMSK2[14]	GCLK_TCC3 : GCLK.PCHCTRL[24]	35	54 : EV0 55 : EV1 56 : MC0 57 : MC1 58 : MC2 59 : MC3 60 : MC4 61 : MC5	77 : OVF 78 : TRG 79 : CNT 80 : MC0 81 : MC1 82 : MC2 83 : MC3 84 : MC4 85 : MC5	34 : OVF 35 : MC0 36 : MC1 37 : MC2 38 : MC3 39 : MC4 40 : MC5	VDDCORE_SW
TCC4	0x4500_8000	120 : CNT, DFS, ERR, FAULTA, FAULTB, FAULT0, FAULT1, OVF, TRG, UFS 121 : MC0 122 : MC1	MCLK.CLKMSK3[4]	GCLK_TCC4 : GCLK.PCHCTRL[29]	40	62 : EV0 63 : EV1 64 : MC0 65 : MC1	86 : OVF 87 : TRG 88 : CNT 89 : MC0 90 : MC1	49 : OVF 50 : MC0 51 : MC1	VDDCORE_SW

Peripheral Dependencies (continued)

Peripheral Name	Base Address	NVIC IRQ Index: Source	MCLK AHBx/APBx Clock Enable Mask Bit	GCLK Peripheral Channel Clock Name:Register	PAC Peripheral Identifier (PAC.WRCTRL.PERIDx)	EVSYS Users (EVSYS.USERm)	EVSYS Generator (EVSYS.CHANNELn.EVGENx)	DMA Trigger Index:Source (DMAC.CHCTRLBk.TRIGx)	Power Domain
TCC5	0x4500_A000	123 : CNT, DFS, ERR, FAULTA, FAULTB, FAULT0, FAULT1, OVF, TRG, UFS 124 : MC0 125 : MC1	MCLK.CLKMSK3[5]	GCLK_TCC5 : GCLK.PCHCTRL[30]	41	66 : EV0 67 : EV1 68 : MC0 69 : MC1	91 : OVF 92 : TRG 93 : CNT 94 : MC0 95 : MC1	51 : OVF 52 : MC0 53 : MC1	VDDCORE_SW
TCC6	0x4500_C000	126 : CNT, DFS, ERR, FAULTA, FAULTB, FAULT0, FAULT1, OVF, TRG, UFS 127 : MC0 128 : MC1	MCLK.CLKMSK3[6]	GCLK_TCC6 : GCLK.PCHCTRL[31]	42	70 : EV0 71 : EV1 72 : MC0 73 : MC1	96 : OVF 97 : TRG 98 : CNT 99 : MC0 100 : MC1	54 : OVF 55 : MC0 56 : MC1	VDDCORE_SW
TCC7	0x4500_E000	129 : CNT, DFS, ERR, FAULTA, FAULTB, FAULT0, FAULT1, OVF, TRG, UFS 130 : MC0 131 : MC1	MCLK.CLKMSK3[7]	GCLK_TCC7 : GCLK.PCHCTRL[32]	43	74 : EV0 75 : EV1 76 : MC0 77 : MC1	101 : OVF 102 : TRG 103 : CNT 104 : MC0 105 : MC1	57 : OVF 58 : MC0 59 : MC1	VDDCORE_SW

48.5.1. I/O Lines

In order to use the I/O lines of this peripheral, the I/O pins must be configured using the I/O Pin Controller (PORT).

References:

- PORT - I/O Pin Controller

48.5.2. Clocks

The TCC bus clocks (CLK_TCCx_APB) where x is 0,1,2...7 is enabled by default, and can be enabled or disabled in the Main Clock (MCLK).

A generic clock (GCLK_TCCx) is required to clock the TCC. This clock must be configured and enabled in the Generic Clock Controller (GCLK) before using the TCC.

The generic clocks (GCLK_TCCx) are asynchronous to the bus clock (CLK_TCCx_APB). Due to this asynchronicity, writing certain registers will require synchronization between the clock domains. Refer to Synchronization for further details.

References:

- Peripheral Clock Masking

48.5.3. DMA

The DMA request lines are connected to the DMA Controller (DMAC). In order to use DMA requests with this peripheral the DMAC must be configured first. Refer to the *DMAC – Direct Memory Access Controller* chapter for details.

References:

- DMAC

48.5.4. Interrupts

The interrupt request line is connected to the Interrupt Controller. In order to use interrupt requests of this peripheral, the Interrupt Controller (NVIC) must be configured first. Refer to the *Nested Vector Interrupt Controller* chapter for details.

References:

- Nested Vector Interrupt Controller

48.5.5. Events

The events of this peripheral are connected to the Event System.

References:

- EVSYS

48.5.6. 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.

References:

- TCC DBGCTRL Register

48.5.7. Register Access Protection

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

- The Interrupt Flag register (INTFLAG)
- The Status register (STATUS)
- The Period and Period Buffer registers (PER, PERBUF)
- The Compare/Capture and Compare/Capture Buffer registers (CCy, CCBUFy)
- The Control Waveform register (WAVE)
- The Pattern Generation Value and Pattern Generation Value Buffer registers (PATT, PATTBUF)

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

When the CPU is halted in Debug mode, write-protection is automatically disabled.

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

48.6. Functional Description

48.6.1. Principle of Operation

The following definitions are used throughout the documentation:

Table 48-1. Timer/Counter for Control Applications - Definitions

Name	Description
TOP	The counter reaches TOP when it becomes equal to the highest value in the count sequence. The TOP value can be the same as Period (PER) or the Compare Channel 0 (CC0) register value depending on the Waveform Generator mode in *Waveform Output Generation Operations*.
ZERO	The counter reaches ZERO when it contains all zeroes. (i.e. 0x00000000)
MAX	The counter reaches maximum when it contains all ones. (i.e. 0xFFFFFFFF)
UPDATE	The timer/counter signals an update when it reaches ZERO or TOP, depending on the direction settings.
Timer	The timer/counter clock control is handled by an internal source.
Counter	The clock control is handled externally (e.g., counting external events).
CC	For compare operations, the CC are referred to as "compare channels." For capture operations, the CC are referred to as "capture channels."

There are up to six compare/capture (CC) channels starting from CC0 to CC5. The number of CC channels can be different on different instances of a TCC.

The Counter register (COUNT), Period registers with Buffer (PER and PERBUF), and Compare and Capture registers with buffers (CCy and CCBUFy) are 32-bit registers, on each TCC instance. Each Buffer register has a corresponding Buffer Valid flag in the STATUS register 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 operation, these comparisons are used to set the waveform period or pulse width.

A prescaled generic clock (GCLK_TCCx) and events from the event system can be used to control the counter. The event system is also used as a source to the input capture.

The Recoverable Fault Unit enables event controlled waveforms by acting directly on the generated waveforms of the TCC compare channels output. These events can restart, halt the timer/counter period, shorten the output pulse active time, or disable waveform output as long as the fault condition is present. This can typically be used for current sensing regulation, and zero-crossing and demagnetization re-triggering.

The Recoverable Fault event inputs are connected to the MC0 and MC1 event lines. Only asynchronous events are used internally 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 *Recoverable Faults*.

In order to support applications for different types of motors, ballasts, LEDs, H-bridges, power converters, and other types of power switching applications, the following independent units are implemented in some of the TCC instances:

- Recoverable faults and non-recoverable faults
- Output matrix
- Dead-time insertion
- Swap
- Pattern generation

See also TCC Block Diagram.

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. A non-recoverable fault can be recovered only by software.

The count event sources TCCx_EV_0 connected to TCE0 input and TCCx_EV_1 connected to TCE1 input are shared with the non-recoverable fault extension. The events can be optionally filtered. If the filter options are not used, the non-recoverable faults provide an immediate asynchronous action on waveform output, even for cases where the clock is not present. For further details on how to configure asynchronous events routing, refer to section *EVSYS – Event System*.

References:

- EVSYS

48.6.2. Basic Operation

48.6.2.1. Synchronization

Due to asynchronicity between the main clock domain and the peripheral clock domains, some registers need to be synchronized when written or read.

The following bits are synchronized when written:

- Software Reset and Enable bits in Control A register (CTRLA.SWRST and CTRLA.ENABLE)

The following registers are synchronized when written:

- Control B Clear and Control B Set registers (CTRLBCLR and CTRLBSET)
- Status register (STATUS)
- Pattern and Pattern Buffer registers (PATT and PATTBUF)
- Waveform register (WAVE)
- Count Value register (COUNT)
- Period Value and Period Buffer Value registers (PER and PERBUF)
- Compare/Capture Channel y and Compare/Capture Channel y Buffer Value registers (CCy and CCBUFy)

The following registers are synchronized when read:

- Control B Clear and Control B Set registers (CTRLBCLR and CTRLBSET)
- Count Value register (COUNT): synchronization is done on demand through READSYNC command (CTRLBSET.CMD)
- Pattern and Pattern Buffer registers (PATT and PATTBUF)
- Waveform register (WAVE)
- Period Value and Period Buffer Value registers (PER and PERBUF)
- Compare/Capture Channel y and Compare/Capture Channel y Buffer Value registers (CCy and CCBUFy)

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.

Reference:

- Register Synchronization

48.6.2.2. Initialization

The following registers are enable-protected, meaning that they can only be written when the TCC is disabled (CTRLA.ENABLE=0):

- Recoverable Fault Control registers (FCTRLA and FCTRLB)
- Waveform Extension Control register (WEXCTRL)
- Drive Control register (DRVCTRL)
- Event Control register (EVCTRL)

Register Enable-protection is denoted by the 'Enable-Protected' property in the register description.

The following register bits are enable-protected, meaning that they can only be written when the TCC is disabled (CTRLA.ENABLE=0):

- Capture Channel x Enable bits in Control A register (CTRLA.CPTEN[31:24])
- DMA One-shot Trigger Mode (CTRLA.DMAOS)
- Full Cycle (CTRLA.FCYCLE)
- Auto Lock (CTRLA.ALOCK)
- Prescaler and Counter Synchronization Selection (CTRLA.PRESCSYNC)
- Run in Standby (CTRLA.RUNSTDBY)
- Prescaler (CTRLA.PRESCALER)
- Enhanced Resolution (CTRLA.RESOLUTION)

When CTRLA.ENABLE bit (CTRLA <1>) is '0', Enable-protected bits in the CTRLA register can be written at the same time CTRLA.ENABLE bit (CTRLA <1>) is written to '1'. However, when CTRLA.ENABLE bit (CTRLA <1>) is '1' these bits can not be written at the same time CTRLA.ENABLE bit (CTRLA <1>) 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. Configure the clock source for the TCC Instance in the Main Clock Controller (MCLK) and enable the APB BUS clock for the TCC Instance by writing a ‘1’ to the TCCx_ bit in the APB Mask register of the MCLK (i.e., enable CLK_TCC0_APB for TCC0 by setting TCC0_ bit (APBCMASK<3>), CLK_TCC1_APB for TCC1 by setting TCC1_ bit (APBCMASK<4>) etc.).
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 (i.e., for Channel 0 CTRLA.CPTEN0 bit (CTRLA <24>), for Channel 1 CTRLA.CPTEN1 bit (CTRLA <25>) etc.).

Optionally, the following configurations can be set before enabling TCC:

1. Select PRESCALER setting in the Control A register (CTRLA.PRESCALER bits (CTRLA <10:8>)).
2. Select Prescaler Synchronization setting in Control A register (CTRLA.PRESYNC bits (CTRLA <13:12>)).
3. If down-counting operation is desired, write the Counter Direction bit in the Control B Set register (CTRLBSET.DIR bit (CTRLBSET <0>)) to '1'. In this case, the COUNT register must be initialized with the desired TOP value.

4. Select the Waveform Generation operation in the WAVE register (WAVEGEN bits (WAVE <2:0>)).
5. The output polarity for each individual channel can be changed by configuring the corresponding WAVE.POLy bit, where y=0,1,2,..7. For example, when the CC0 register is used for the duty cycle, the corresponding WAVE.POL0 bit (WAVE<16>) decides the polarity.
6. The waveform output for each individual channel can be inverted by configuring corresponding Waveform Output Invert Enable bit DRVCTRL.INVENy bits where y = 0,1,2...7. For example, when CC0 register is used for the duty cycle, corresponding DRVCTRL.INVEN0 bit (DRVCTRL <16>) decides inversion.

Note: See different PWM modes for detailed information.

48.6.2.3. 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 register for details.

The TCC should be disabled before the TCC is reset to avoid undefined behavior.

48.6.2.4. Prescaler Selection

The GCLK_TCCx clock is fed into the internal prescaler.

The prescaler consists of a counter that counts up to the selected prescaler value, whereupon the output of the prescaler toggles.

If the prescaler value is higher than one, the Counter Update condition can be optionally executed on the next GCLK_TCCx clock pulse or the next prescaled clock pulse. For further details, refer to the Prescaler (CTRLA.PRESCALER) and Counter Synchronization (CTRLA.PRESYNC) descriptions.

Prescaler outputs from 1 to 1/1024 are available. For a complete list of available prescaler outputs, see the register description for the Prescaler bit group in the Control A register (CTRLA.PRESCALER).

Note: When counting events, the prescaler is bypassed, the joint stream of prescaler ticks and event action ticks is called CLK_TCC_COUNT.

48.6.2.5. Counter Operation

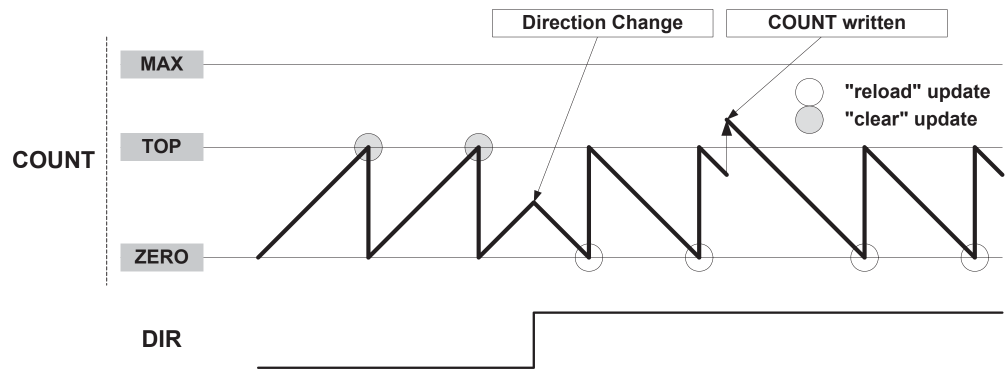
Depending on the mode of operation, the counter is cleared, reloaded, incremented, or decremented at each TCC clock input (CLK_TCC_COUNT). A counter clear or reload mark the end of current counter cycle and the start of a new one.

The counting direction is set by the Direction bit in the Control B Set/Clear registers (CTRLBSET and CTRLBCLR). If the bit reads zero, it is counting up and if the bit reads one, it is counting down.

The counter will count up or down for each tick (clock or event) until it reaches TOP or ZERO. When it is counting up and TOP is reached, the counter will be set to zero or TOP 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 bit is set. When TC is counting down, the COUNT register must be initialized to TOP value (PER or CC0 value depending on the mode).

INTFLAG.OVF bit 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 section “Additional Features”.

Figure 48-2. Counter Operation



It is possible to change the counter value (by writing directly in the COUNT register) even when the counter is running.

When starting the TCC, the COUNT value will always be ZERO or TOP, depending on direction set by CTRLBSET.DIR bit (CTRLBSET <0> or CTRLBCLR.DIR bit (CTRLBCLR <0>), unless a different value has been written to it, or the TCC has been stopped at a value other than ZERO or TOP.

The write access has higher priority than count, clear, or reload. The direction of the counter can also be changed during normal operation. See the previous figure.

Stop Command

A stop command can be issued from software by using TCC Command bits in Control B Set register (CMD bits (CTRLBSET <7:5>)=0x2, STOP).

When a stop is detected while the counter is running, and if CTRLA.FCYCLE bit (CTRLA <16>) = 0 the counter stops immediately maintaining its current value, however, if CTRLA.FCYCLE bit (CTRLA <16>)=1, it waits until end of current cycle to stop on a start value.

If the waveform generation operation (WG) is used, all waveforms are set to a state defined in Non-Recoverable State y Output Enable bit and Non- Recoverable State y Output Value bit in the Driver Control register (DRVCTRL.NREy and DRVCTRL.NRVy), and the Stop bit in the Status register is set (STATUS.STOP bit (STATUS <0>)).

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.EVACT0 bits (EVCTRL <2:0>)=0x3, STOP).

When a pause is detected, the counter will maintain its current value and all waveforms keep their current state when (CTRLA.FCYCLE (CTRLA <16>)=0), or wait to the end of the current cycle to stop on a start value and all waveforms keep their initial state when (CTRLA.FCYCLE (CTRLA <16>)= 1), until a start event action is detected: Input Event Action 0 bits in Event Control register (EVCTRL.EVACT0 bits (EVCTRL <2:0>) = 0x3, START).

Re-Trigger Command and Event Action

A re-trigger command can be issued from software by using TCC Command bits in Control B Set register (CTRLBSET.CMD (CTRLBSET <7:5>)=0x1, RETRIGGER), or from event when the re-trigger event action is configured in the Input Event 0/1 Action bits in Event Control register (EVCTRL.EVACT0 bits (EVCTRL <2:0>)=0x1, RETRIGGER and EVCTRL.EVACT1 bits (EVCTRL <5:3>) =0x1, RETRIGGER).

When the command is detected during counting operation, the counter will be reloaded or cleared, depending on the counting direction (CTRLBSET.DIR (CTRLBSET <0>) or (CTRLBCLR.DIR (CTRLBCLR <0>)). The Re-Trigger bit in the Interrupt Flag Status and Clear register will be set (TRG bit (INTFLAG <1>)). It is also possible to generate an event by writing a '1' to the Re-Trigger Event Output Enable bit in the Event Control register (EVCTRL.TRGEO bit (EVCTRL <9>)). If the re-trigger command is detected when the counter is stopped, the counter will resume counting operation from the value in COUNT.

Note:
When a re-trigger event action is configured in the Event Action bits in the Event Control register (EVCTRL.EVACT0 bits (EVCTRL <2:0>)=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 (EVCTRL <2:0>)=0x3, START) and can start the counting operation when previously stopped. The event has no effect if the counter is already counting. When the module is enabled, the counter operation starts when the event is received or when a re-trigger software command is applied.

Note:
When a start event action is configured in the Event Action bits in the Event Control register (EVCTRL.EVACT0 (EVCTRL <2:0>)=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 (CTRLBSET <0>) or CTRLBCLR.DIR (CTRLBCLR <0>)).

The count event action is selected by the Event Action 0 bit group in the Event Control register (EVCTRL.EVACT0 (EVCTRL <2:0>)=0x5, COUNT).

Direction Event Action

The direction event action can be selected in the Event Control register (EVCTRL.EVACT1 (EVCTRL <5:3>)=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 (CTRLBSET <0>) or CTRLBCLR.DIR (CTRLBCLR <0>)) 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 (EVCTRL <2:0>)=0x4, INC) and can change the Counter state when an event is received. When the TCCx_EV_0 (TCE0) event is received, the counter increments, irrespective of direction setting (CTRLBSET.DIR (CTRLBSET <0>) or CTRLBCLR.DIR (CTRLBCLR <0>)).

Decrement Event Action

The decrement event action can be selected in the Event Control register (EVCTRL.EVACT1 (EVCTRL <5:3>)=0x4, DEC) and can change the Counter state when an event is received. When the TCCx_EV_1 (TCE1) event is received, the counter decrements, irrespective of direction setting (CTRLBSET.DIR (CTRLBSET <0>) or CTRLBCLR.DIR (CTRLBCLR <0>)).

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.NREy and DRVCTRL.NRVy). TCCx_EV_0 input event on TCE0 and TCCx_EV_1 input events on 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.

48.6.2.6. 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 (CCy) for compare operations, the counter value is continuously compared to the values in the CCy registers. This can be used for timer or for waveform operation.

The Channel y Compare/Capture Buffer Value (CCBUFy) registers provide double buffer capability. The double buffering synchronizes the update of the CCy register with the buffer value at the UPDATE condition or a force update command (CTRLBSET.CMD (CTRLBSET <5:3>)=0x3, UPDATE). For further details, refer to *Double Buffering*. The synchronization prevents the occurrence of odd-length, non-symmetrical pulses and ensures glitch-free output.

48.6.2.6.1. 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 Operation in the Waveform Generation Operation bits in Waveform register (WAVE.WAVEGEN (WAVE <2:0>)).
2. Optionally invert the waveform output WO[y] by writing the corresponding Waveform Output y Inversion bit in the Driver Control register (DRVCTRL.INVENy).
3. Configure the pins with the I/O Pin Controller. Refer to PORT - I/O Pin Controller for details.
   Note: Event must not be used when the compare channel is set in waveform output operating mode.

The counter value is continuously compared with each CCy value. On a comparison match, the Match or Capture Channel y bit in the Interrupt Flag Status and Clear register (INTFLAG.MCy) will be set on the next zero-to-one transition of GCC_TCCx (see Normal Frequency Operation). An interrupt and/or event can be generated on the same condition if Match/Capture occurs, i.e. INTENSET.MCy and/or EVCTRL.MCEOy is '1'. Both interrupt and event can be generated simultaneously. 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-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 Operations, the update time occurs on counter wraparound, on overflow, on underflow, or on re-trigger.

The table below shows the update counter and overflow event/interrupt generation conditions in different Waveform Generation Operations.

Table 48-2. Counter Update and Overflow Event/interrupt Conditions

Name	Operation	TOP	Update	Output Waveform		OVF Interrupt Flag/Event	
				On Match	On Update	Up	Down
NFRQ	Normal Frequency	PER	TOP/ ZERO	Toggle	Stable	TOP	ZERO
MFRQ	Match Frequency	CC0	TOP/ ZERO	Toggle	Stable	TOP	ZERO
NPWM	Single-slope PWM	PER	TOP/ ZERO	See section 'Output Polarity' below		TOP	ZERO
DSCRITICAL	Dual-slope PWM	PER	ZERO			-	ZERO
DSBOTTOM	Dual-slope PWM	PER	ZERO			-	ZERO
DSBOTH	Dual-slope PWM	PER	TOP(1) & ZERO			TOP	ZERO
DSTOP	Dual-slope PWM	PER	ZERO			TOP	–

1. The UPDATE condition on TOP only will occur when circular buffer is enabled for the channel.

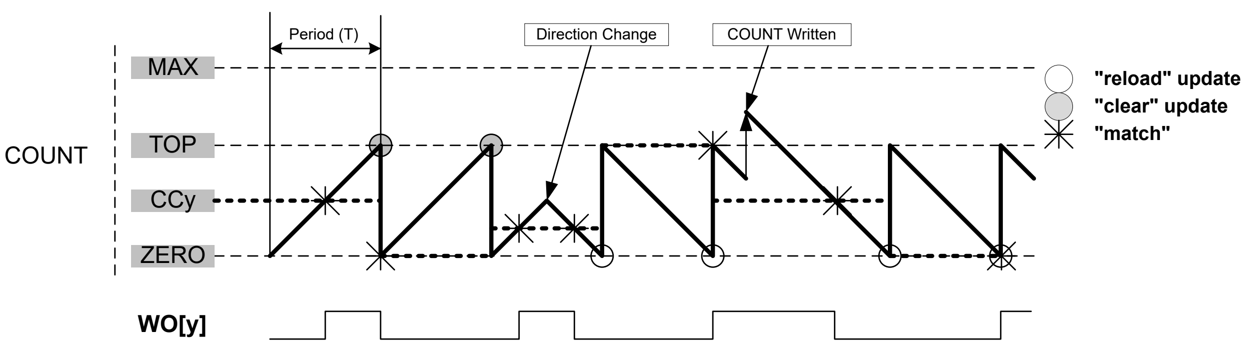
References:

- Circular Buffer
- PORT I/O

48.6.2.6.2. Normal Frequency (NFRQ)

For Normal Frequency generation, the period time (T) is controlled by the period register (PER). The waveform generation output (WO[y]) is toggled on each compare match between COUNT and CCy, and the corresponding Match or Capture Channel y Interrupt Flag (INTFLAG.MCy) will be set.

Figure 48-3. Normal Frequency Operation



The following steps must be performed to operate the TCC in Normal Frequency (NFRQ) Waveform Generation Operations.

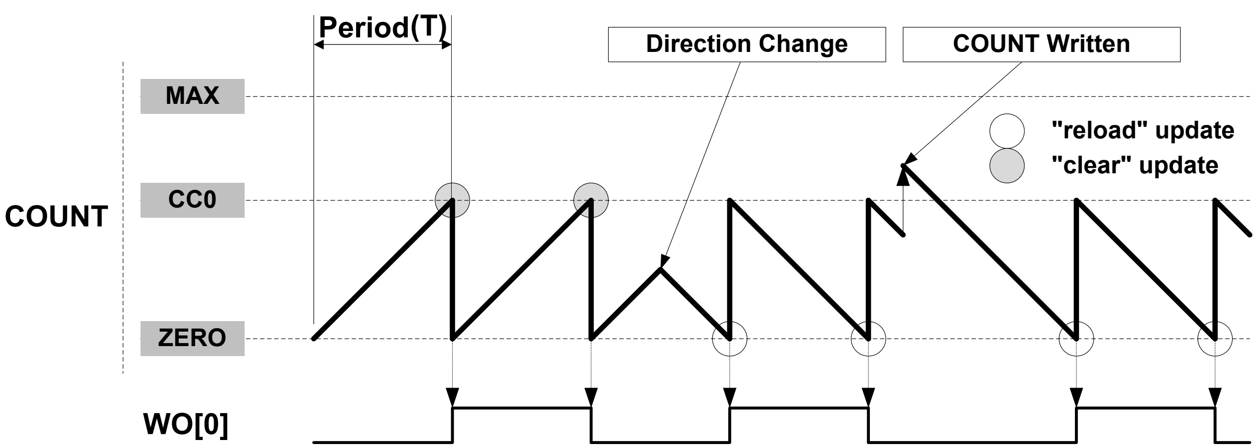
1. Configure the clock source for the TCC Instance in the Main Clock Controller (MCLK) and enable the APB BUS clock for the TCC Instance by writing a '1' to the TCCx_ bit in the APB Mask register of the MCLK (i.e. enable CLK_TCC0_APB for TCC0 by setting TCC0_ bit (APBCMASK<3>), CLK_TCC1_APB for TCC1 by setting TCC1_ bit (APBCMASK<4>) etc.).
2. Enable Generic clock for TCC Instance (e.g. enable GCLK_TCC0 for TCC0 by setting PCHCTRL31.CHEN bit PCHCTRL31<6>, GCLK_TCC1 for TCC1 by setting PCHCTRL32.CHEN bit PCHCTRL32<6> etc.).
3. Select desired Prescaler by setting CTRLA.PRESCALER bits (CTRLA<10:8>).
4. Select matrix routing to desired port pins for generated output waveform, by configuring OTMX bits (WEXCTRL.WEXCTRL<1:0>).

5. Set Waveform Generation Operations to Normal Frequency Operation (NRFQ) by clearing WAVE.WAVEGEN bits (WAVE<2:0> =0)
6. Load the selected Compare/Capture (CCy) register (e.g. CC0<31:0>) with the desired compare match value. The generated output will toggle on this match.
7. Load the period register PER<31:0> with the desired time period value.
8. Set Counter to count in up direction by clearing CTRLBCLR.DIR bit (CTRLBCLR<0> = 1). To change the counter direction down set counter direction bit CTRLBSET.DIR(CTRLBSET<0> =1) The waveform output of a channel CCy can be inverted by configuring the corresponding Waveform Output Invert Enable bit DRVCTRL.INVENy bits where y = 0,1,2...7. For example, when the CC0 register is used for the duty cycle, the corresponding INVEN0 bit (DRVCTRL<16>) decides inversion.
9. If overflow interrupt is used, set INTENSET.OVF bit (INTENSET<0>) and configure the NVIC by setting group priority, sub priority and enabling TCCx IRQ.
10. Enable TCC by setting CTRLA.ENABLE bit (CTRLA<1>).
11. To know how to clear interrupts, see Interrupts.

48.6.2.6.3. 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 48-4. Match Frequency Operation



The following steps must be performed to operate the TCC in Match Frequency (MFRQ) Operation.

1. Configure the clock source for the TCC Instance in the Main Clock Controller (MCLK) and enable the APB BUS clock for the TCC Instance by writing a ‘1’ to the TCCx_ bit in the APB Mask register of the MCLK (i.e. enable CLK_TCC0_APB for TCC0 by setting TCC0_ bit (APBCMASK<3>), CLK_TCC1_APB for TCC1 by setting TCC1_ bit (APBCMASK<4>) etc.).
2. Enable Generic clock for TCC Instance (e.g. enable GCLK_TCC0 for TCC0 by setting CHEN bit PCHCTRL31.PCHCTRL31<6>, GCLK_TCC1 for TCC1 by setting PCHCTRL32.CHEN bit PCHCTRL32<6> etc.).
3. Select desired prescaler by setting CTRLA.PRESCALER bits (CTRLA<10:8>).
4. Set Waveform Generation Operation to MATCH Frequency Operation (MFRQ) by setting WAVE.WAVEGEN bits (WAVE<2:0> = 0x1).
5. Load the selected Compare/Capture (CC) register (e.g. CC0<31:0>) with the desired compare match value. The generated output will toggle on this match, PER register is not used in this operations.

6. Set Counter to count in up direction by clearing CTRLBCLR.DIR bit (CTRLBCLR<0> = 1). To change the counter direction down set counter direction bit CTRLBSET.DIR(CTRLBSET<0> =1).
7. Select matrix routing to desired port pins for generated output waveform, by configuring WEXCTRL.OTMX bits (WEXCTRL<1:0>).
8. The waveform output can be inverted by configuring DRVCTRL.INVEN0 bit (DRVCTRL<16>).
9. If overflow interrupt is used, set INTENSET.OVF bit (INTENSET<0>) and configure the NVIC by setting group priority, sub priority and enabling corresponding TCCx IRQ.
10. Enable TCC by setting CTRLA.ENABLE bit (CTRLA<1>).
11. To know how to clear interrupts, see Interrupts.

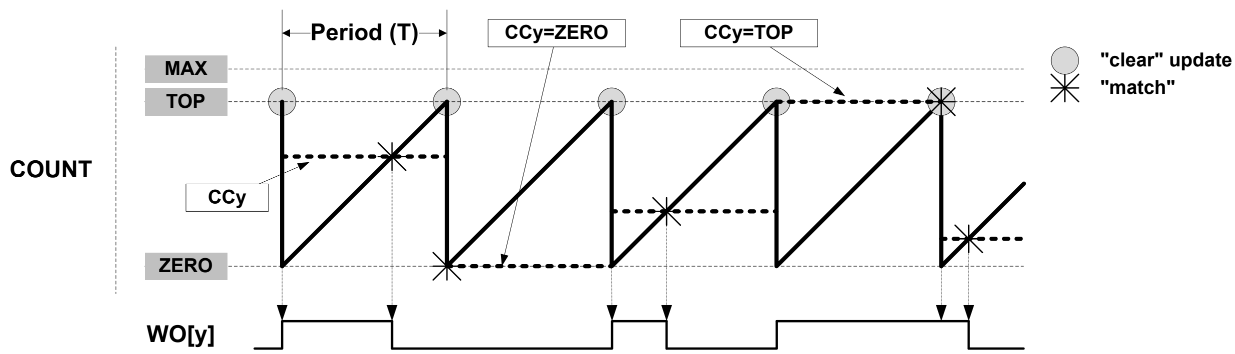
48.6.2.6.4. Normal Pulse-Width Modulation (NPWM)

NPWM uses single-slope PWM generation.

48.6.2.6.5. Normal Pulse-Width Modulation (NPWM) Single-Slope Operation

For single-slope PWM generation, the period setting (TOP) is controlled by the PER register, and CCy controls the duty cycle of the generated waveform output. When up-counting, the WO[y] is set at start or at compare match between the COUNT and TOP values, and cleared on compare match between COUNT and CCy register values. When down-counting, the WO[y] is cleared at start or at compare match between the COUNT and ZERO values, and set on compare match between COUNT and CCy register values.

Figure 48-5. Single-Slope PWM Operation



The following equation calculates the exact resolution for a single-slope PWM (R_{PWM_SS}) waveform:

$$R_{\mathrm{PWM\_SS}} = \frac{\log(\mathrm{PER}+1)}{\log(2)}$$

The PWM frequency depends on the Period register value (PER) and the peripheral clock frequency ($f_{GCLK\_TCCx = 1/GCLK\_TCCx}$), and can be calculated by the following equation:

$$f_{\mathrm{PWM\_SS}} = \frac{f_{\mathrm{GCLK\_TCCx}}}{\mathrm{N}(\mathrm{PER}+1)}$$

Where N represents the prescaler divider used CTRLA.PRESCALER (CTRLA<10:8>). The selected prescaler value can be 1, 2, 4, 8, 16, 64, 256, or 1024.

The following steps must be performed to operate the TCC in single-slope PWM generation operation.

1. Configure the clock source for the TCC Instance in the Main Clock Controller (MCLK) and enable the APB BUS clock for the TCC Instance by writing a ‘1’ to the TCCx_ bit in the APB Mask register of the MCLK (i.e. enable CLK_TCC0_APB for TCC0 by setting TCC0_ bit (APBCMASK<3>), CLK_TCC1_APB for TCC1 by setting TCC1_ bit (APBCMASK<4>) etc.).

2. Enable Generic clock for TCC Instance (e.g. enable GCLK_TCC0 for TCC0 by setting PCHCTRL31.CHEN bit PCHCTRL31<6>, GCLK_TCC1 for TCC1 by setting PCHCTRL32.CHEN bit PCHCTRL32<6> etc.).
3. Select desired prescaler by setting CTRLA.PRESCALER bits (CTRLA<10:8>).
4. Set Waveform Generation Operation to Normal Pulse Width Modulation Operation (NPWM) by setting WAVE.WAVEGEN bits (WAVE<2:0> = 2).
5. Load the selected Compare/Capture (CCy) register (e.g. CC0<31:0>) with the desired PWM duty cycle value.
6. Load the period register PER<31:0> with the desired time period value.
7. Set Counter to count in up direction by clearing CTRLBCLR.DIR bit (CTRLBCLR<0> = 1). To change the counter direction down set counter direction bit CTRLBSET.DIR(CTRLBSET<0> =1).
8. Select matrix routing to desired port pins for generated output waveform, by configuring WEXCTRL.OTMX bits (WEXCTRL<1:0>).
9. The output polarity for each individual channel can be changed by configuring the corresponding WAVE.POLy bit, where y= 0,1,2,..7. For example, when the CC0 register is used for the duty cycle, the corresponding POL0 bit (WAVE<16>) decides the polarity.
10. The waveform output for each individual channel can be inverted by configuring the corresponding Waveform Output Invert Enable bit DRVCTRL.INVENy bits where y = 0,1,2...7. For example, when the CC0 register is used for the duty cycle, the corresponding INVEN0 bit (DRVCTRL<16>) decides inversion.
11. If overflow interrupt is used, set OVF bit (INTENSET<0>) and configure the NVIC by setting group priority, sub priority and enabling corresponding TCCx IRQ.
12. Enable TCC by setting ENABLE bit (CTRLA<1>).
13. To know how to clear interrupts, see Interrupts.

48.6.2.6.6. Normal Pulse-Width Modulation (NPWM) Dual-Slope Operation

For dual-slope PWM generation, the period setting (TOP) is controlled by PER, while CCy 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 depends on DS mode (DSTOP, DSBOTTOM, DSBOTH or DSCRITICAL). Depending on Dual-Slope mode (DSTOP, DSBOTTOM, DSBOTH, or DSCRITICAL), an interrupt and/or event is generated on TOP (when counting upwards) and/or ZERO (when counting up or down).

In DSBOTH operation, the circular buffer must be enabled to enable the update condition on TOP.

References:

- Circular Buffer

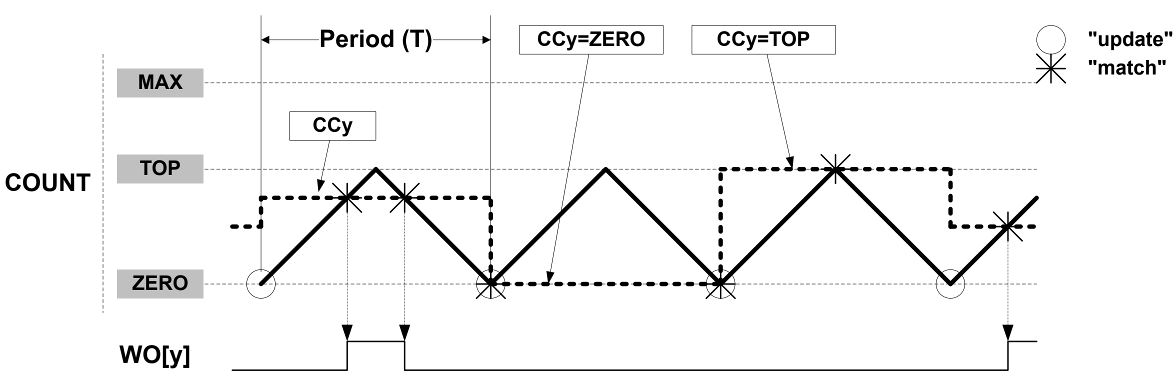
The following steps must be performed to operate the TCC in Dual-slope PWM generation operation.

1. Configure the clock source for the TCC Instance in the Main Clock Controller (MCLK) and enable the APB BUS clock for the TCC Instance by writing a ‘1’ to the TCCx_ bit in the APB Mask register of the MCLK (i.e. enable CLK_TCC0_APB for TCC0 by setting TCC0_ bit (APBCMASK<3>), CLK_TCC1_APB for TCC1 by setting TCC1_ bit (APBCMASK<4>) etc.).
2. Enable Generic clock for TCC Instance (e.g. enable GCLK_TCC0 for TCC0 by setting PCHCTRL31.CHEN bit PCHCTRL31<6>, GCLK_TCC1 for TCC1 by setting PCHCTRL32.CHEN bit PCHCTRL32<6> etc.).
3. Select desired prescaler by setting CTRLA.PRESCALER bits (CTRLA<10:8>).
4. Select prescaler Synchronization PRESCSYNC bits (CTRLA <13:12>).

5. Set waveform generation operation to Dual Slope PWM by configuring WAVE.WAVEGEN bits (WAVE<2:0>)
   a. For DSCRITICAL operation, set the value of WAVE.WAVEGEN bit (WAVE<2:0> = 4)
   b. For DSBOTTOM operation, set the value of WAVE.WAVEGEN bit (WAVE<2:0> = 5)
   c. For DSBOTH operation, set the value of WAVE.WAVEGEN bit (WAVE<2:0> = 6)
   d. For DSTOP operation, set the value of WAVE.WAVEGEN bit (WAVE<2:0> = 7)
6. Load the selected Compare/Capture (CCy) register (i.e. CC0<31:0>) with the desired PWM duty cycle value.
7. Load the period register PER<31:0> with the desired time period value.
8. Set Counter to count in up direction by clearing DIR bit (CTRLBCLR<0> = 1).
9. If DSBOTH operation (WAVEGEN bit (WAVE<2:0> = 6)) is set, enable Circular buffer by setting CIPEREN bit (WAVE<7>).
10. Select matrix routing to desired port pins for generated output waveform, by configuring WEXCTRL.OTMX bits (WEXCTRL<1:0>).
11. The dead time insertion on output matrix can be enabled by setting WEXCTRL.DTIENy bit, where y=0,1,2,3.
12. Higher side and Lower side PWM Dead-time is programmed in WEXCTRL.DTHS (WEXCTRL<30:24>) and WEXCTRL.DTLS (WEXCTRL<23:16>) respectively.
13. The output polarity for each individual channel can be changed by configuring the corresponding WAVE.POLy bit, where y= 0,1,2,..7. For example, when the CC0 register is used for the duty cycle, the corresponding POL0 bit (WAVE<16>) decides the polarity.
14. The waveform output for each individual channel can be inverted by configuring the corresponding Waveform Output Invert Enable bit DRVCTRL.INVENy bits where y = 0,1,2...7. For example, when the CC0 register is used for the duty cycle, the corresponding INVEN0 bit (DRVCTRL<16>) decides inversion.
15. If overflow interrupt is used, set OVF bit (INTENSET<0>) and configure the NVIC by setting group priority, sub priority and enabling corresponding TCCx IRQ.
16. Enable TCC by setting ENABLE bit (CTRLA<1>).
17. For instructions on how to clear interrupts, see Interrupts.

Dual-slope Critical (DSCRITICAL) PWM operation is explained below.

Figure 48-6. 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}):

$$R_{\text{PWM\_DS}} = \frac{\log(\text{PER}+1)}{\log(2)}.$$

The PWM frequency $f_{\text{PWM\_DS}}$ depends on the period setting (TOP) and the peripheral clock frequency $f_{\text{GCLK\_TCCx}}$, and can be calculated by the following equation:

$$f_{\text{PWM\_DS}} = \frac{f_{\text{GCLK\_TCCx}}}{2N \cdot \text{PER}}$$

N represents the prescaler divider used. The prescaler can be selected by CTRLA.PRESCALER bits (CTRLA<10:8>). The selected prescaler value can be 1, 2, 4, 8, 16, 64, 256, 1024.. The waveform generated will have a maximum frequency of half of the TCC clock frequency ($f_{\text{GCLK\_TCCx}}$) when TOP is set to 0x00000001 and no prescaling is used.

The pulse width ($P_{\text{PWM\_DS}}$) depends on the compare channel (CCy) register value and the peripheral clock frequency ($f_{\text{GCLK\_TCCx}}$), and can be calculated by the following equation:

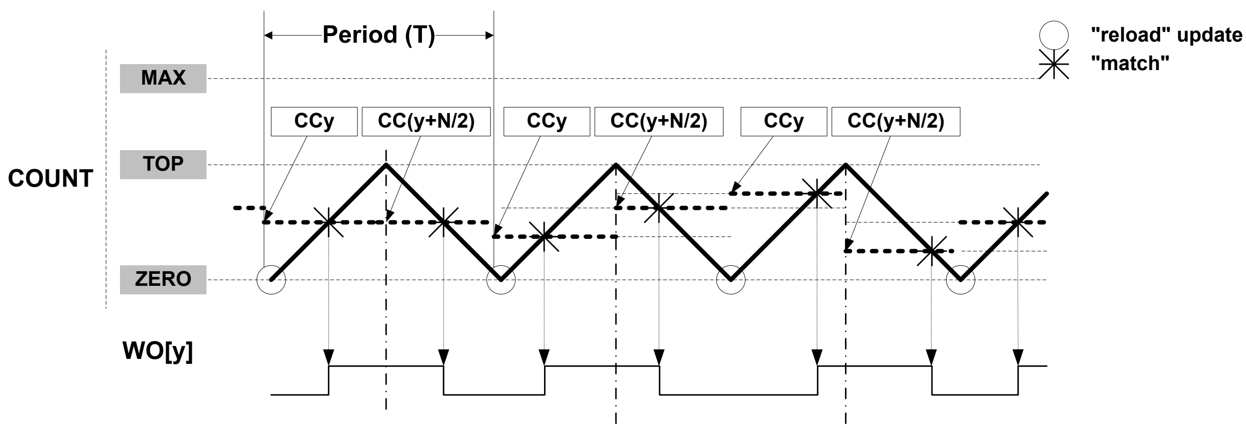
$$P_{\text{PWM\_DS}} = \frac{2N \cdot (\text{TOP} - \text{CC}y)}{f_{\text{GCLK\_TCCx}}}$$

Note: In DSTOP, DSBOTTOM and DSBOTH operation, when TOP is lower than MAX/2, the MSB bit of CCy defines the Ramp on which the CCy Match interrupt or event is generated. (Rising if CCy[MSB] = 0, falling if CCy[MSB] = 1.)

48.6.2.6.7. Dual-Slope Critical PWM Generation

Critical generation operation allows generation of non-aligned centered pulses. In this operation, the period time is controlled by PER while CCy control the generated waveform output edge during up-counting and CC(y+CC_NUM/2) control the generated waveform output edge during down-counting.

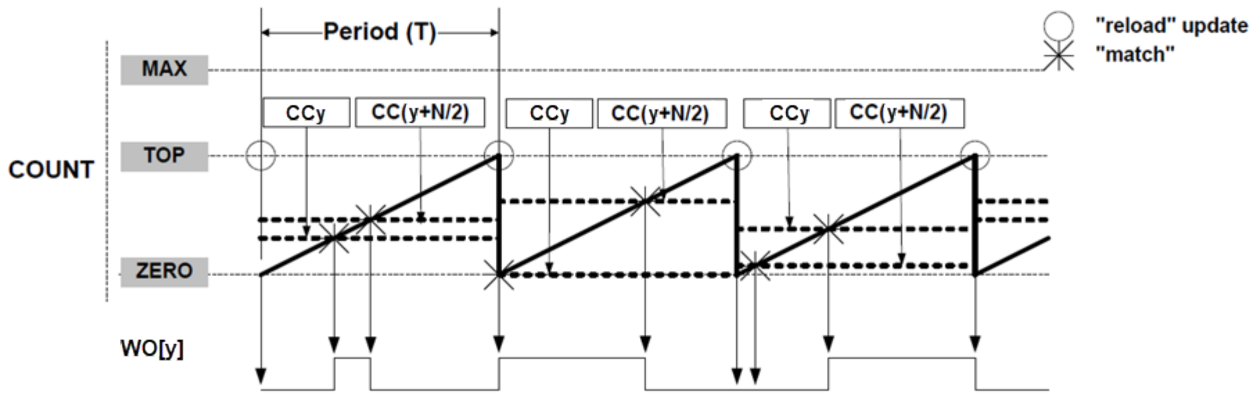
Figure 48-7. Dual-Slope Critical Pulse Width Modulation (N=CC_NUM)



48.6.2.6.8. Dual-Compare PWM Generation

Dual compare PWM generation allows generation of pulses unaligned on start or end of a period. In this operation, the period time is controlled by PER, while CCy controls the waveform output leading edge and the CC(y+CC_NUM/2) controls the waveform output trailing edge.

Figure 48-8. Dual-Compare Pulse Width Modulation (N=CC_NUM)



48.6.2.6.9. Output Polarity

The polarity (WAVE.POLy) is available in all waveform output generation. In single-slope and dual-slope PWM operation, it is possible to invert the pulse edge alignment individually on start or end of a PWM cycle for each compare channels. The table below shows the waveform output set/clear conditions, depending on the settings of timer/counter, direction, and polarity.

Table 48-3. Waveform Generation Set/Clear Conditions

Waveform Generation Operation	DIR	POL	Waveform Generation Output Update	
			Set	Clear
Single-Slope PWM	0	0	Timer/counter matches TOP	Timer/counter matches CC
		1	Timer/counter matches CC	Timer/counter matches TOP
	1	0	Timer/counter matches CC	Timer/counter matches ZERO
		1	Timer/counter matches ZERO	Timer/counter matches CC
Dual-Slope PWM	x	0	Timer/counter matches CC when counting up	Timer/counter matches CC when counting down
		1	Timer/counter matches CC when counting down	Timer/counter matches CC when counting up

In Normal and Match Frequency, the WAVE.POLy value represents the initial state of the waveform output.

48.6.2.7. Double Buffering

The Pattern (PATT), Period (PER) and Compare Channels (CCy) registers are all double buffered. Each buffer register has a buffer valid (PATTBUFV(STATUS <5>), PERBUFV(STATUS<7>) and CCBUFVy) 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 a Buffer Valid Status flag (PATTBUFV,PERBUFV or CCBUFVy) is set and the corresponding SYNCBUSY bit is set (SYNCBUSY.PATT(SYNCBUSY<5>),SYNCBUSY.PER(SYNCBUSY<7>) or SYNCBUSY.CCy), a write to the respective PATT/PATTBUF, PER/PERBUF or CCy/CCBUFy registers will generate a PAC error.

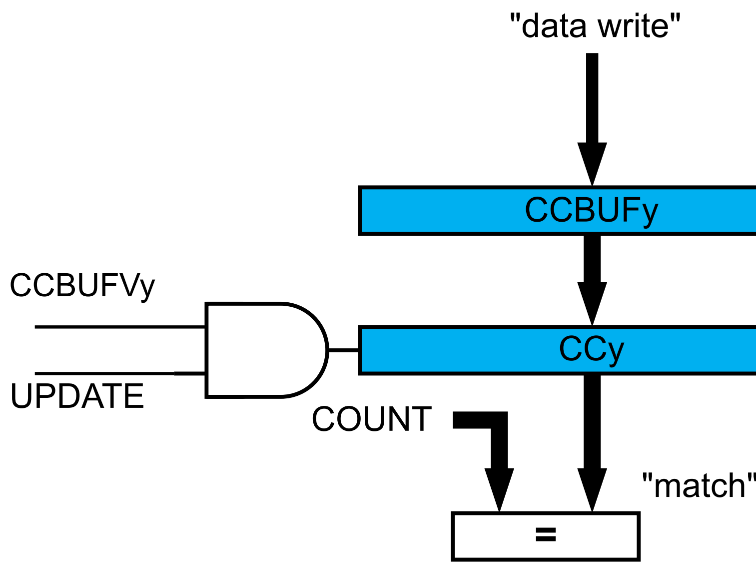
When the Buffer Valid Flag bit in the STATUS register is '1' and the Lock Update bit in the CTRLBCLR register is cleared, (writing CTRLBCLR.LUPD(CTRLBCLR<1>) to '1'), update of register by its buffer register is allowed: Data from the buffer register will be copied into the corresponding register under hardware UPDATE conditions, then the Buffer Valid flag bit in the STATUS register is automatically cleared by the hardware.

When the buffer valid flag bit in the STATUS register is '1' (i.e. the buffer register contains a valid value), and the Lock Update bit in the CTRLB register is set to '1', (writing CTRLBSET.LUPD(CTRLBSET<1>) to '1'), update of a register by its buffer register is disabled: Data from buffer register is not copied into the corresponding register on any UPDATE conditions and the buffer valid flag bit in the STATUS register stay unchanged.

Note: Software update command (CTRLBSET.CMD(CTRLBSET<7:5>)=0x3) act independently of LUPD value.

A compare register is double buffered as in the following figure.

Figure 48-9. Compare Channel Double Buffering



The registers (PATT/PER/CCy) and corresponding Buffer registers (PATTBUF/PERBUF/CCBUFy) are available in the I/O register map, and the double buffering feature is not mandatory. Double buffering feature can be bypassed by directly writing on (PATT/PER/CCy) registers.

Changing the Period

The counter period can be changed by writing a new TOP value to the register that decides period (PER or CCy, depending on the Waveform Generation Operation), however period update on registers (PER or CCy) is effective after the synchronization delay, irrespective of whether double buffer is enabled or disabled.

Figure 48-10. Unbuffered Single-Slope Up-Counting Operation

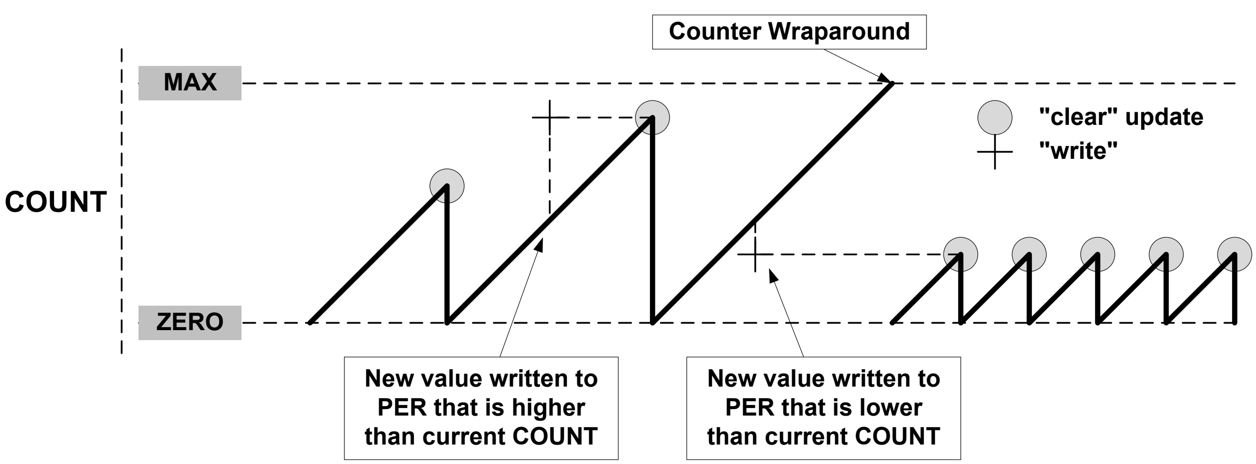
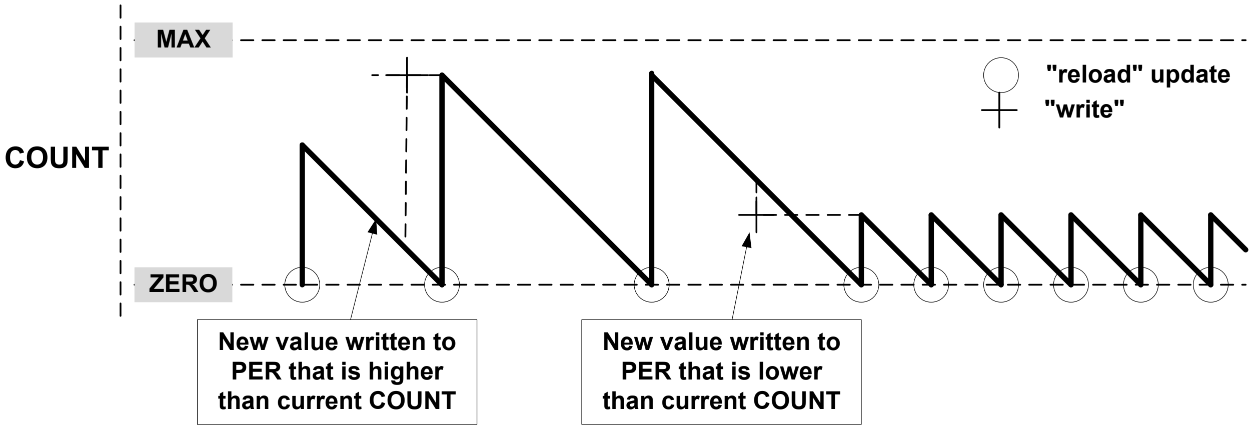
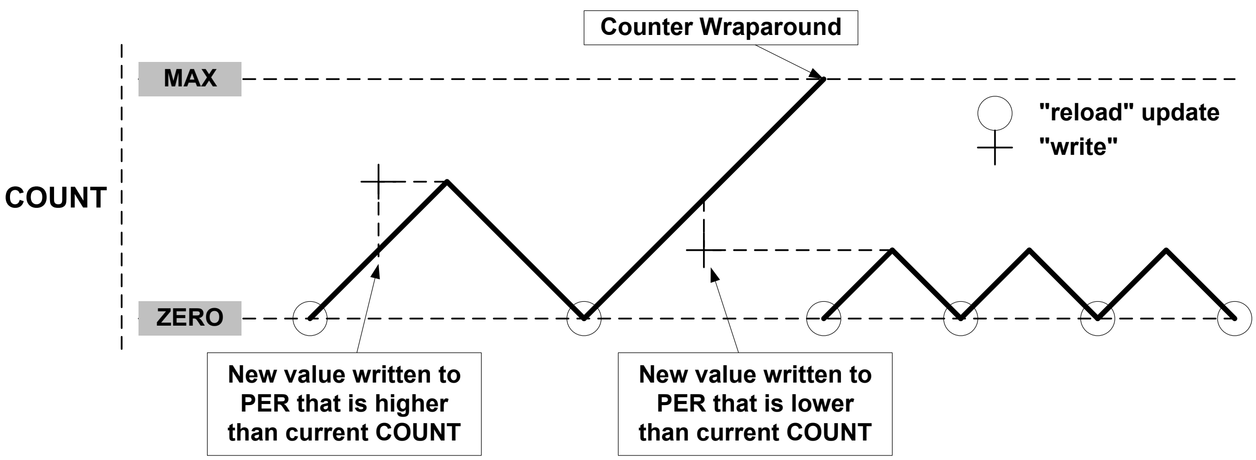


Figure 48-11. Unbuffered Single-Slope Down-Counting Operation



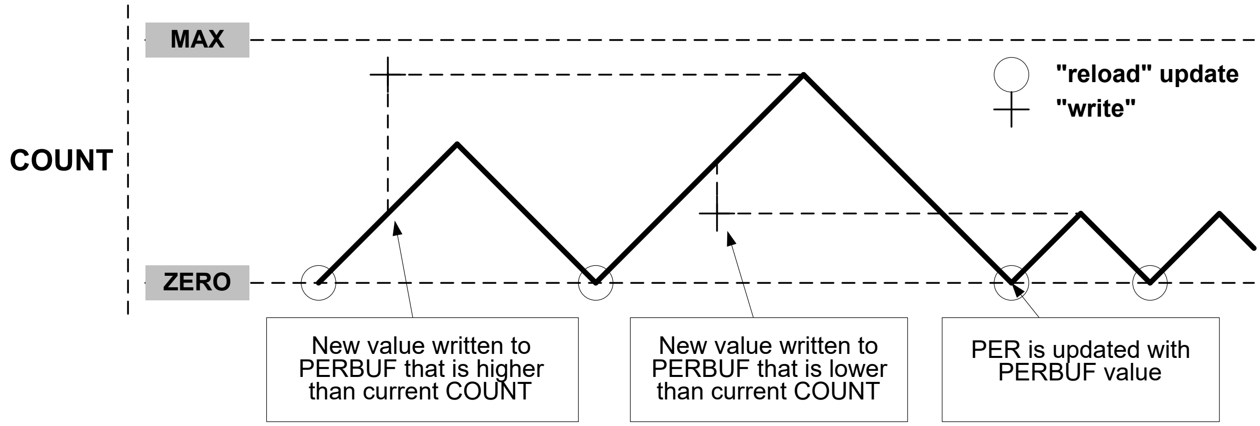
A counter wraparound can occur in any Waveform Generation Operation when up-counting without buffering, see the previous figure. COUNT and TOP are continuously compared, so when a new value that is lower than the current COUNT is written to TOP, COUNT will wrap before a compare match. Similarly, unbuffered down-counting operation is shown in the previous figure. Unbuffered operation in dual slope mode is shown in the following figure.

Figure 48-12. Unbuffered Dual-Slope Operation



When double buffering is used, the buffer can be written at any time and the counter will still maintain correct operation. The period register is always updated on the update condition, as shown in *Changing the Period Using Buffer*. This prevents wraparound and the generation of odd waveforms.

Figure 48-13. Changing the Period Using Buffering



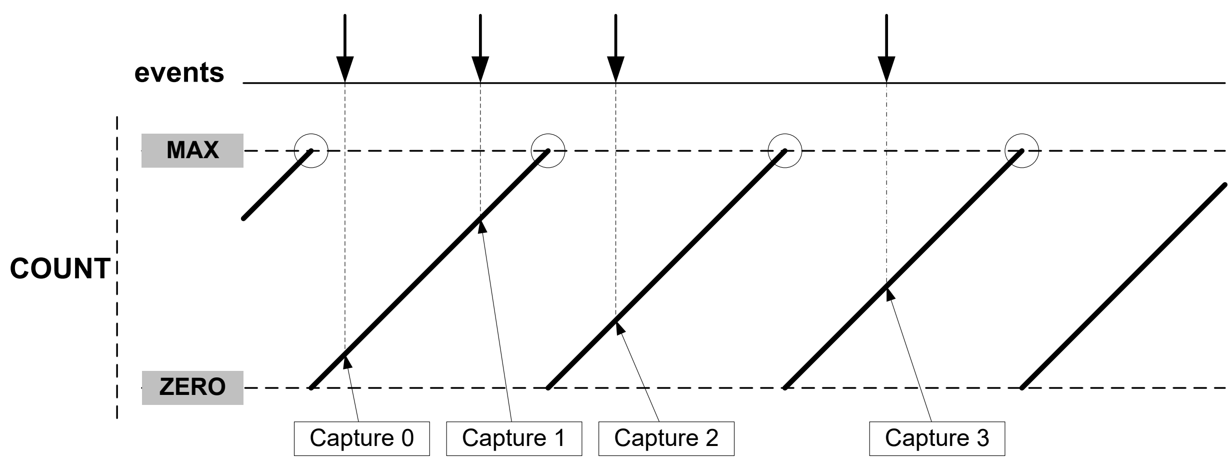
48.6.2.8. Capture Operations

To enable and use capture operations, the Match or Capture Channel y Event Input Enable bit (MCEIy) in the Event Control register must be written to '1'. The capture channels to be used must also be enabled in the Capture Channel y Enable bit (CPTENy) in the Control A register (CTRLA.CPTENy) before capturing can be performed.

Event Capture Action

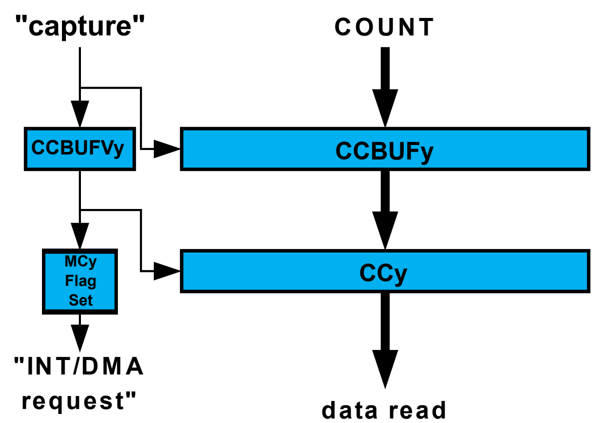
The compare/capture channels can be used as input capture channels to capture events from the Event System, and give them a timestamp. The following figure shows four capture events for one capture channel. Event system channels must be configured to operate in asynchronous mode when used for capture operations.

Figure 48-14. Input Capture Timing



For input capture, the Buffer register and the corresponding CCy act like a FIFO. When CCy is empty or read, any content in CCBUFy is transferred to CCy. The Buffer Valid flag (STATUS.CCBUFVy) is passed to set the CCy Interrupt flag (INTFLAG.MCy) and generate the optional interrupt, event, or DMA request. The CCBUFy register value cannot be read, all captured data must be read from the CCy register.

Figure 48-15. Capture Double Buffering



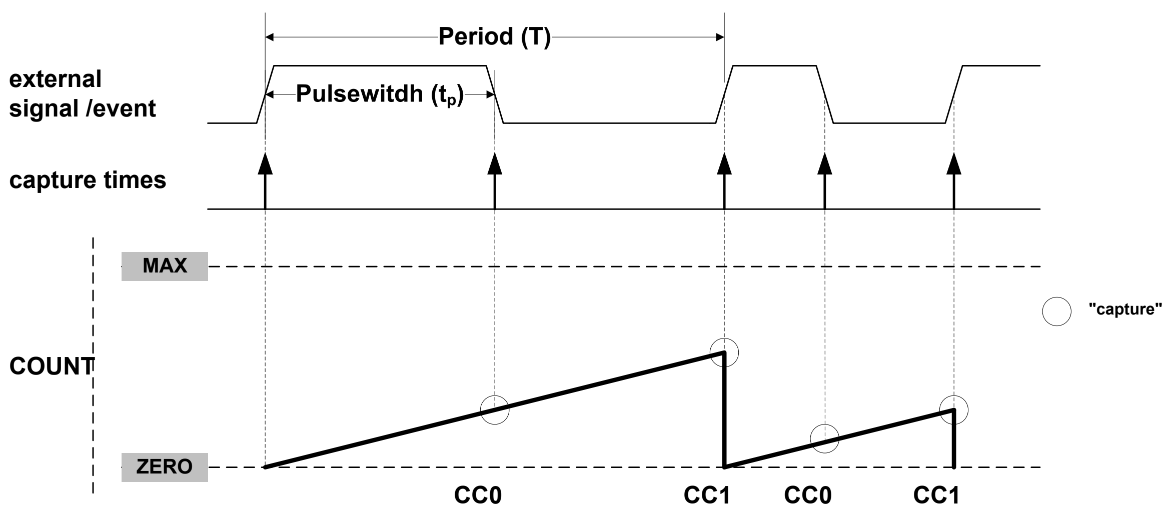
The TCC can detect capture overflow of the input capture channels. When a new capture event is detected while the Capture Buffer Valid flag, CCBUFVy (For example STATUS.CCBUFV0 for CCBUF0 buffer register) is still set, the new timestamp will not be stored and INTFLAG.ERR(INTFLAG<3>) will be set.

Pulse-Width and Period (PPW) Capture Action

The TCC can perform two input captures and restart the counter on one of the edges. This enables the TCC to measure the pulse-width and period and to characterize the frequency *f* and *dutyCycle* of an input signal, as shown below:

$$f = \frac{1}{T} \quad , \quad dutyCycle = \frac{t_p}{T}$$

Figure 48-16. PWP Capture



Selecting Pulse Width and Period (PWP) in the Timer/Counter Event Input 1 Action bit group in the Event Control register (EVCTRL.EVACT1(EVCTRL <5:3>)) enables the TCC to perform one capture action on the rising edge and the other one on the falling edge. In PWP event action offers, the *T* is captured into CC1 and *tp* into CC0.

The Timer/Counter Event (TCE) n Invert Enable bit in Event Control register (EVCTRL.TCEINVn,where n=0,1 and input events are TCCx_EV_0 or TCCx_EV_1) is used for event source n to select whether the reload should occur on the rising edge or the falling edge. If EVCTRL.TCEINVn=1, the reload will happen on the falling edge.

The corresponding capture is done only if the channel is enabled in Capture mode (CTRLA.CPTENy=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.MCy is still set, the new timestamp will not be stored and INTFLAG.ERR (INTFLAG<3>) will be set.

Note: When up-counting (CTRLBCLR.DIR(CTRLBCLR <0>)=1), counter values lower than 1 cannot be captured in Capture Minimum mode (FCTRLA.CAPTURE(FCTRLA<5:4>)=CAPTMIN or FCTRLB.CAPTURE(FCTRLA<5:4>)=CAPTMIN). To capture the full range including value 0, the TCC must be configured in Down-counting mode (CTRLBSET.DIR(CTRLBCLR <0>)=1).

48.6.3. Additional Features

48.6.3.1. One-Shot Operation

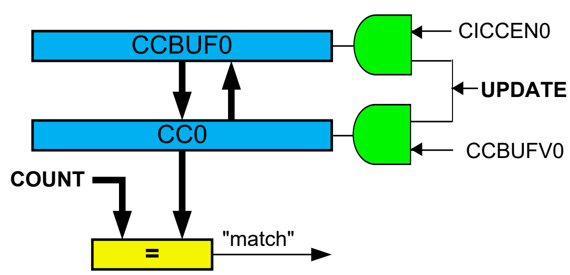
When one-shot is enabled, the counter automatically stops on the next Counter Overflow or Underflow condition. When the counter is stopped, the Stop bit in the Status register (STATUS.STOP(STATUS <0>)) is set and the waveform outputs are set to the value defined by DRVCTRL.NREy and DRVCTRL.NRVy.

One-shot operation can be enabled by writing a '1' to the One-Shot bit in the Control B Set register (CTRLBSET.ONESHOT(CTRLBSET<2>)) and disabled by writing a '1' to CTRLBCLR.ONESHOT(CTRLBCLR<2>). When enabled, the TCC will count until an overflow or underflow occurs and stop counting. The one-shot operation can be restarted by a re-trigger software command, a re-trigger event or a start event. When the counter restarts its operation, STATUS.STOP(STATUS <0>) is automatically cleared.

48.6.3.2. Circular Buffer

The Period register (PER) and the CC0 to CC5 registers for Compare Channels support circular buffer operation. When circular buffer operation is enabled, the PER or CCy values are copied into the corresponding buffer registers at each update condition. Circular buffering is dedicated to the RAMP2, RAMP2A, and DSBOTH operations.

Figure 48-17. Circular Buffer on Channel 0



48.6.3.3. Dithering Operation

The TCC supports dithering on Pulse-width or Period on a 16, 32 or 64 fractional clock cycle base.

Dithering consists in adding some extra clock cycles on some PWM cycles, to 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 Resolution bits in CTRLA register (CTRLA.RESOLUTION (CTRLA <6:5>)) and gives user three different options::

- DITH4 dithering resolution is based on the overflow of a 4 bit-counter

- DITH5 dithering resolution is based on the overflow of a 5 bit-counter
- DITH6 dithering resolution is based on the overflow of a 6 bit-counter

The least significant bits of COUNT, PER and CCy registers are used to improve accuracy of output pulse width and period. These bits are called DITHERCY bits.

The remaining bits of COUNT, PER and CCy registers define the compare value for the normal operation. The DITHERCY bits of COUNT, PER and CCy define the increment on respective dithering counter (COUNT, PER or CCy) registers to perform on each PWM cycle. If the value of dithering counter in CC register is '0' output pulse width dithering will be disabled. Similarly, if the value of dithering counter in PER register is '0' period dithering will be disabled.

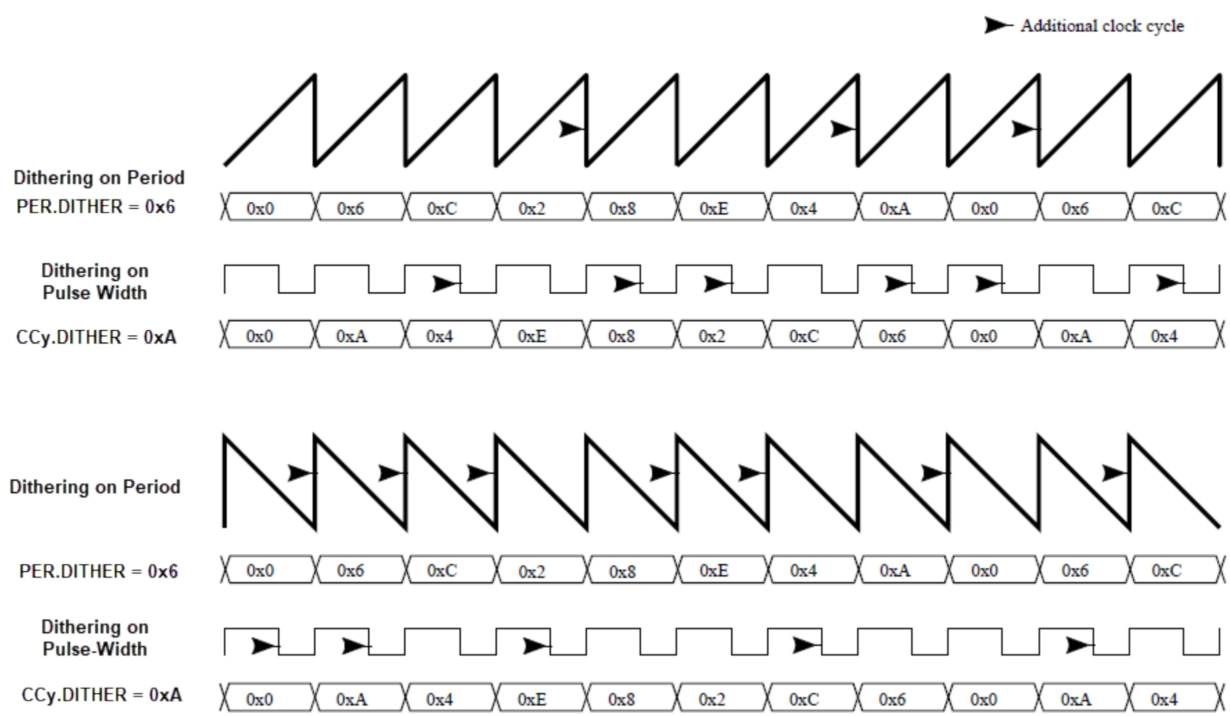
Dithering works as described below:

- In up-counting operation an extra clock cycle is inserted on each dithering counter overflow
- In down-counting operation an extra clock cycle is inserted on each PWM cycles, except when a dithering counter overflow occur

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;
}
```

Figure 48-18. Additional Clock Cycles Location in DTH4 Resolution Mode (up counting and down counting)



Dithering on Period

Writing DITHERCY in PER will lead to an average PWM period configured by the following formulas.

DITH4 mode:

If DITH4 mode is enabled for the period the 6 least significant bits from PER and COUNT registers correspond to the period DITHERCY value, rest of the bits correspond to PER or COUNT value.

$$PwmPeriod = \left(\frac{\mathrm{DITHERCY}}{16} + \mathrm{PER}\right)\left(\frac{1}{f_{\mathrm{GCLK\_TCC}x}}\right)$$

DITH5 mode:

If DITH5 mode is enabled for the period the 6 least significant bits from PER and COUNT registers correspond to the DITHERCY value, rest of the bits correspond to PER or COUNT value.

$$PwmPeriod = \left(\frac{\mathrm{DITHERCY}}{32} + \mathrm{PER}\right)\left(\frac{1}{f_{\mathrm{GCLK\_TCC}x}}\right)$$

DITH6 mode:

If DITH6 mode is enabled for the period the 6 least significant bits from PER and COUNT registers correspond to the DITHERCY value, rest of the bits correspond to PER or COUNT value.

$$PwmPeriod = \left(\frac{\mathrm{DITHERCY}}{64} + \mathrm{PER}\right)\left(\frac{1}{f_{\mathrm{GCLK\_TCC}x}}\right)$$

Dithering on Pulse-Width

Writing DITHERCY in CCy will lead to an average PWM pulse width configured by the following formulas.

DITH4 mode:

If DITH4 mode is enabled for pulse width the 6 least significant bits from CCy and COUNT registers correspond to the output pulse width DITHERCY value, rest of the bits correspond to CCy or COUNT value.

$$PwmPulseWidth = \left(\frac{\mathrm{DITHERCY}}{16} + \mathrm{CC}y\right)\left(\frac{1}{f_{\mathrm{GCLK\_TCC}x}}\right)$$

DITH5 mode:

If DITH5 mode is enabled for pulse width the 6 least significant bits from CCy and COUNT registers correspond to the output pulse width DITHERCY value, rest of the bits correspond to CCy or COUNT value.

$$PwmPulseWidth = \left(\frac{\mathrm{DITHERCY}}{32} + \mathrm{CC}y\right)\left(\frac{1}{f_{\mathrm{GCLK\_TCC}x}}\right)$$

DITH6 mode:

If DITH6 mode is enabled for pulse width the 6 least significant bits from CCy and COUNT registers correspond to the output pulse width DITHERCY value, rest of the bits correspond to CCy or COUNT value.

$$PwmPulseWidth = \left(\frac{\mathrm{DITHERCY}}{64} + \mathrm{CC}y\right)\left(\frac{1}{f_{\mathrm{GCLK\_TCC}x}}\right)$$

Note: The PWM period will remain static in this case.

48.6.3.4. Ramp Operations

This device supports Five Ramp Operations. All these RAMP Operations require the timer/counter running in single-slope PWM generation operation. The Ramp Operation is selected by writing to the RAMP[2:0] bits (WAVE <6:5>) in the Waveform Control register. The Ramp Operations are broadly divided into RAMP1 Operation and RAMP2x Operation.

RAMP1 Operation

This is the default PWM operation, described in *Single-Slope PWM Generation*.

RAMP2x Operation

These operations are dedicated for power factor correction (PFC), Half-Bridge and Push-Pull SMPS topologies, where two consecutive timer/counter cycles are interleaved, see Standard RAMP2 Operation. These cycles are called Ramp A and Ramp B. In the Ramp A cycle, the odd channel output is disabled, and in the Ramp B cycle, the 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(CTRLBSET <4:3>)).

The RAMP2x functionality interleaves the duty cycle of two output signals in a single TCC module. The RAMP2x functions require that the TCC module is used in single slope operation (counting up or down but not both). Each of the different RAMP2x operations use different resources available in the TCC module to achieve the interleaving output.

The TCC module is extremely flexible and can be used in many different types of applications.

Most commonly, the TCC is known for its usage in all forms of motor control applications. In motor control applications, where power is supplied by an AC source there is the need to condition power before the motor control in the system. The TCC module has built in features to enable Interleaved Power Factor Correction to reduce inefficiencies and cost in the system design for power conditioning where an electric motor is controlled. Interleaved Power Factor Correction uses two outputs of a TCC module in RAMP2x configuration to operate two parallel PFC converters at 180º out of phase from each other. It is important to maintain the phase of the two converters precisely to maintain equal load sharing between the two channels. This can all be done simply with the use of the TCC module in one of the three RAMP2x operations. Faults can be generated anywhere in the microcontroller and quickly relayed to the TCC module for immediate handling for protection or control methods.

There are four RAMP2x operations:

1. Standard RAMP2 Operation
2. Alternate RAMP2 (RAMP2A) Operation
3. Critical RAMP2 (RAMP2C) Operation
4. Critical Swapped RAMP2 (RAMP2CS) Operation

Standard RAMP2 (RAMP2) Operation

RAMP2 operation uses the PER register of the TCC module to determine the period of both output waveforms. CC0 and CC1 registers of the TCC module are used to define the duty cycle of W[0] and W[1] respectfully. To identify the cycle (A or B) of the operation, the Ramp index bit of the STATUS register STATUS.IDX (STATUS<1>) can be read.

In the following figure, the output of both WO[0] and WO[1] is going high (duty cycle begins) when the respective CC0 and CC1 value match COUNT, and goes low (duty cycle ends) when the TOP, or PER value matches the COUNT register value. This is a result of the polarity bits WAVE.POL0 (WAVE<16>) and WAVE.POL1 (WAVE<17>) being set to one. If the polarity bits were cleared, the duty cycle would begin at the reset of COUNT and conclude on the CC0 and CC1 match.

RAMP2 operation can also provide different periods for cycle A (Ramp A) and cycle B (Ramp B). To enable two different period values (TOP) for cycle A and cycle B, the period circular buffer will have to be enabled (WAVE.CIRPEREN (WAVE<7>)=1) as well as writing the PERBUF register. In this operation the following registers are used.

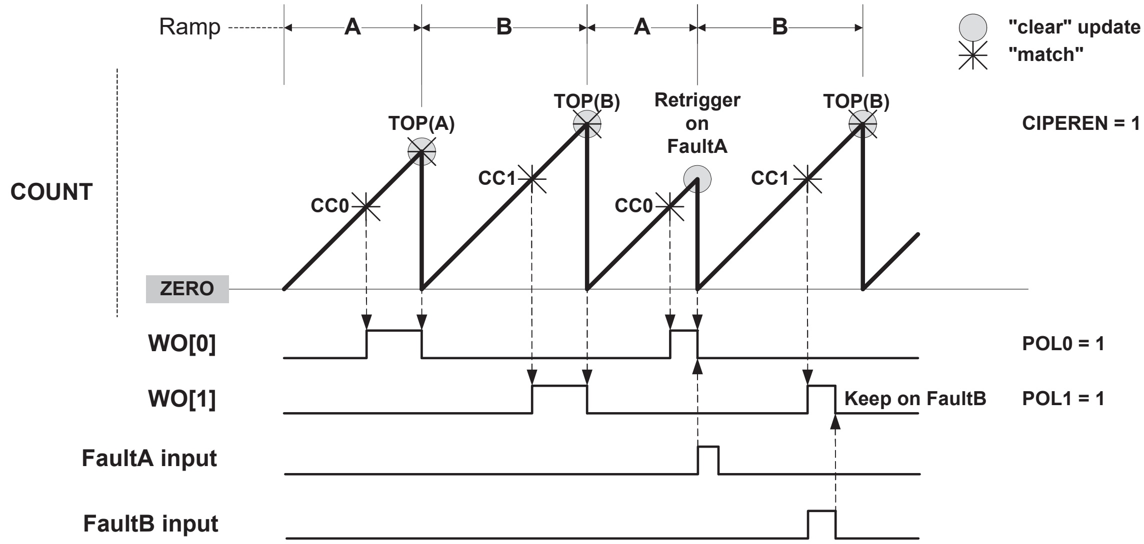
- PER – Period for both cycle A and cycle B when period circular buffer is not enabled
- PERBUF – At initialization, if the period circular buffer is enabled by setting WAVE.CIRPEREN (WAVE<7>) bit (WAVE<7>), then PERBUF defines cycle B period and PER defines cycle A period

- CC0 – Defines the duty cycle of cycle A (Ramp A)
- CC1 – Defines the duty cycle of cycle B (Ramp B)

In all cases updates to the period should be applied through the period buffer register, PERBUF. The use of the Ramp index bit STATUS.IDX (STATUS<1>) should be used to identify the cycle (A or B) of the operation to be sure the proper period is updated. The following steps are performed to operate the TCC in this operation and initialize the RAMP2 operation.

1. Configure the clock source for the TCC Instance in the Main Clock Controller (MCLK) and enable the APB BUS clock for the TCC Instance by writing a ‘1’ to the TCCx_ bit in the APB Mask register of the MCLK (i.e. enable CLK_TCC0_APB for TCC0 by setting TCC0_ bit (APBCMASK<3>), CLK_TCC1_APB for TCC1 by setting TCC1_ bit (APBCMASK<4>) etc.).
2. Enable Generic clock for TCC Instance (e.g. enable GCLK_TCC0 for TCC0 by setting PCHCTRL31.CHEN bit PCHCTRL31<6>, GCLK_TCC1 for TCC1 by setting PCHCTRL32.CHEN bit PCHCTRL32<6> etc.).
3. Select desired prescaler by setting CTRLA.PRESCALER bits (CTRLA<10:8>).
4. Set waveform generation operation to Normal Pulse Width Modulation operation (NPWM) by setting WAVE.WAVEGEN bits (WAVE<2:0> = 2).
5. Set Ramp operation to RAMP2 by setting value of WAVE.RAMP bits (WAVE<6:4>) to 0x1.
6. If cycle B requires different period value than cycle A, enable the Circular buffer by setting WAVE.CIPEREN bit (WAVE<7>).
7. Load the selected Compare/Capture (CC0) register with the desired PWM duty cycle value for Cycle A.
8. Load the another Compare/Capture (CC1) register with the desired PWM duty cycle value for Cycle B.
9. Load the period register PER<31:0> with the desired time period value (TOP) for cycle A.
10. If the circular buffer is enabled, load the period register PERBUF<31:0> with the desired time period value (TOP) for cycle B.
11. Set Counter to count in up direction by clearing CTRLBCLR.DIR bit (CTRLBCLR<0> = 1).
12. In this operation the output polarity of a selected output signal (if needed) should be set as follow
    a. Set WAVE.POL0 bit (WAVE<16>) to invert WO[0] output.
    b. Set WAVE.POL1 bit (WAVE<17>) to invert WO[1] output
13. If overflow interrupt is used, set INTENSET.OVF bit (INTENSET<0>), if compare match interrupt is used, set INTENSET.MC0 bit (INTENSET<16> and configure the NVIC by setting group priority, sub priority and enabling corresponding TCCx IRQ.
14. Enable TCC by setting CTRLA.ENABLE bit (CTRLA<1>).
15. For information on how to clear interrupts, see Interrupts.

Figure 48-19. RAMP2 Standard Operation



Alternate RAMP2 (RAMP2A) Operation

In RAMP2A operation, CC0 is required to use the circular buffer to provide two different duty cycles for cycle A (Ramp A) and cycle B (Ramp B). Updates to the duty cycle can be made by writing the CCBUF0 register at the time of an INTFLAG.OVF (INTFLAG<0>) or INTFLAG.MCy flags flag or MC flag. Writing CCBUF0 at the time of an INTFLAG.OVF (INTFLAG<0>) flag will change the duty-cycle of cycle B. Writing CCBUF0 at the time of an INTFLAG.MCy flag will change the duty cycle of cycle A. To further identify the cycle (A or B) of the operation refer to the Ramp index bit of the STATUS register, STATUS.IDX bit (STATUS<1>). To eliminate the need of monitoring the INTFLAG.OVF (INTFLAG<0>) or INTFLAG.MCy flags for the update to the duty-cycle, the DMA can be used to populate the appropriate buffer register.

RAMP2A operation uses the PER register of the TCC module to determine the period of both output waveforms. If different periods are needed for cycle A and cycle B outputs, the period circular buffer (CIPEREN) will have to be enabled. As mentioned for the duty cycle, the PER and PERBUF registers can be treated in the same manner as the CC0 and CCBUF0 to provide the period updates at the appropriate times. If the DMA is to be used, a separate DMA channel will need to be initialized for the period management.

- PER – Defines the period for both cycle A and cycle B if the period circular buffer is NOT enabled
- PERBUF – At startup PERBUF defines cycle B period and PER defines cycle A period. After the TCC module is started PERBUF holds the period value (TOP) of the next cycle
- CC0 – Defines the duty cycle for both cycle A and cycle B if the CC circular buffer is NOT enabled
- CCBUF0 – At startup CCBUF0 defines cycle B duty cycle and CC0 defines cycle A duty cycle. After the TCC module is started CCBUF holds the duty cycle value of the next cycle

In all cases for use with or without DMA in RAMP2A operation, updates to the period or duty cycle should be applied through the buffer registers (PERBUF or CCBUFy). If both the period and duty cycle have different values in cycle A and cycle B and the DMA is used, two DMA channels will be needed, one for PERBUF and one for CCBUFy. Refer to RAMP2 Alternate Operation.

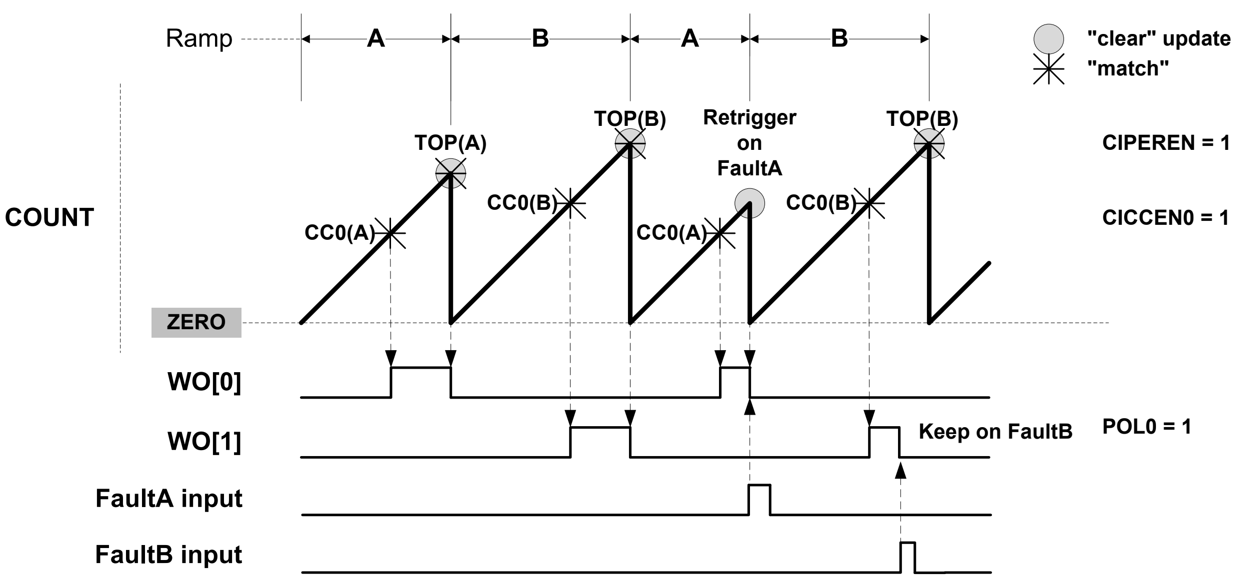
The following steps are performed to operate the TCC in this mode.

1. Configure the clock source for the TCC Instance in the Main Clock Controller (MCLK) and enable the APB BUS clock for the TCC Instance by writing a ‘1’ to the TCCx_ bit in the APB Mask register of the MCLK (i.e. enable CLK_TCC0_APB for TCC0 by setting TCC0_ bit (APBCMASK<3>), CLK_TCC1_APB for TCC1 by setting TCC1_ bit (APBCMASK<4>) etc.).

2. Enable Generic clock for TCC Instance (e.g. enable GCLK_TCC0 for TCC0 by setting PCHCTRL31.CHEN bit PCHCTRL31<6>, GCLK_TCC1 for TCC1 by setting PCHCTRL32.CHEN bit PCHCTRL32<6> etc.).
3. Select desired prescaler by setting CTRLA.PRESCALER bits (CTRLA<10:8>).
4. Set waveform generation operation to Normal Pulse Width Modulation operation (NPWM) by setting WAVE.WAVEGEN bits (WAVE<2:0> = 2).
5. Set Ramp operation to RAMP2A by setting value of WAVE.RAMP bits (WAVE<6:4>) to 0x2.
6. If cycle A requires different period value than cycle B, enable the Circular buffer for period by setting WAVE.CIPEREN bit (WAVE<7>).
7. Load Compare/Capture (CC0<31:0>) register with the desired PWM duty cycle value for Cycle A.
8. Load Compare/Capture buffer (CCBUF0<31:0>) register with the desired PWM duty cycle value for Cycle B.
9. Load the period register PER<31:0> with the desired time period value (TOP) for cycle A.
10. Load the period register PERBUF<31:0> with the desired time period value(TOP) for cycle B.
11. Set Counter to count in up direction by clearing CTRLBCLR.DIR bit (CTRLBCLR<0> = 1).
12. In this operation the output polarity of both output signals on WO[0] and WO[1], is controlled by setting WAVE.POL0 bit (WAVE<16>)
13. If overflow interrupt is used, set INTENSET.OVF bit (INTENSET<0>), if the compare match interrupt is used, set INTENSET.MC0 bit (INTENSET<16>) and configure the NVIC by setting group priority, sub priority and enabling corresponding TCCx IRQ.
14. Enable TCC by setting CTRLA.ENABLE bit (CTRLA<1>).
15. For information on how to clear interrupts, see Interrupts.

Note: Retrigger in RAMP2 operations (RAMP2, RAMP2A, RAMP2C, RAMP2CS) is not supported if a prescaler is used (CTRLA.PRESCALER != 0) and the retrigger of the counter is done on the next GCLK (CTRLA.PRESCSYNC = GCLK or CTRLA.PRESCSYNC = RESYNC). If use of the prescaler is required, configure the retrigger of the counter on the next prescaler clock (CTRLA.PRESCSYNC = PRESC).

Figure 48-20. 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 operation, 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 bit (WAVE<16>)=0. Refer to RAMP2 Critical Operation With More Than 2 Channels for more information.

The RAMP2C operation is the simplest to setup and understand as circular buffers are not used in the implementation. The period and duty cycle of output is controlled as shown below.

- PER – Defines the period for cycle B (Ramp B)
- CC0 – Defines the period for cycle A (Ramp A)
- CC1 – Defines the duty cycle for cycle B (Ramp B)
- CC2 – Defines the duty cycle for cycle A (Ramp A)

RAMP2 Critical Operation With More Than 2 Channels illustrates the RAMP2CS operation with three active channels. The following steps are performed to operate the TCC in this operation.

1. Configure the clock source for the TCC Instance in the Main Clock Controller (MCLK) and enable the APB BUS clock for the TCC Instance by writing a ‘1’ to the TCCx_ bit in the APB Mask register of the MCLK (i.e. enable CLK_TCC0_APB for TCC0 by setting TCC0_ bit (APBCMASK<3>), CLK_TCC1_APB for TCC1 by setting TCC1_ bit (APBCMASK<4>) etc.).
2. Enable Generic clock for TCC Instance (e.g. enable GCLK_TCC0 for TCC0 by setting PCHCTRL31.CHEN bit PCHCTRL31<6>, GCLK_TCC1 for TCC1 by setting PCHCTRL32.CHEN bit PCHCTRL32<6> etc.).
3. Select desired prescaler by setting CTRLA.PRESCALER bits (CTRLA<10:8>).
4. Set waveform generation operation to Normal Pulse Width Modulation Operation (NPWM) by setting WAVE.WAVEGEN bits (WAVE<2:0> = 2).
5. Set Ramp operation to RAMP2C by setting value of WAVE.RAMP bits (WAVE<6:4>) to 0x3.
6. Load Compare/Capture (CC1<31:0>) register with the desired PWM duty cycle value for Cycle B.
7. Load Compare/Capture buffer register (CC2<31:0>) register with the desired PWM duty cycle value for Cycle A.
8. Load the period register PER<31:0> with the desired time period value (TOP) for cycle B.
9. Load the Compare/Capture register (CC0<31:0>) with the desired time period value (TOP) for cycle A.
10. Set Counter to count in up direction by clearing CTRLBCLR.DIR bit (CTRLBCLR<0> = 1).
11. In this operation the output polarity of a selected output signal should be set as follow
    a. Set WAVE.POL2 bit (WAVE<18>) as this controls WO[0] output.
    b. Set WAVE.POL1 bit (WAVE<17>) as this controls WO[1] output
12. If overflow interrupt is used, set INTENSET.OVF bit (INTENSET<0>) ,if compare match interrupt is used, set INTENSET.MC2 bit (INTENSET<18>) for Ramp A match, INTENSET.MC1 bit (INTENSET<17>) for Ramp B match interrupt. Configure the NVIC by setting group priority, sub priority and enabling corresponding TCCx IRQ.
13. Enable TCC by setting CTRLA.ENABLE bit (CTRLA<1>).
14. For information on how to clear interrupts, see Interrupts.

Figure 48-21. RAMP2 Critical Operation With More Than 2 Channels

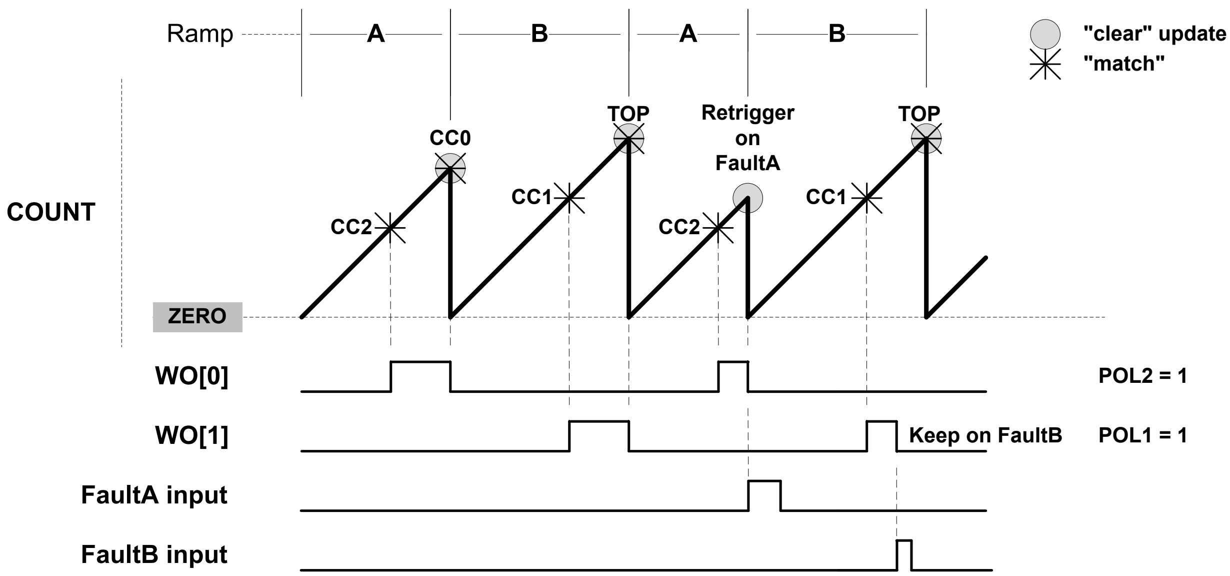
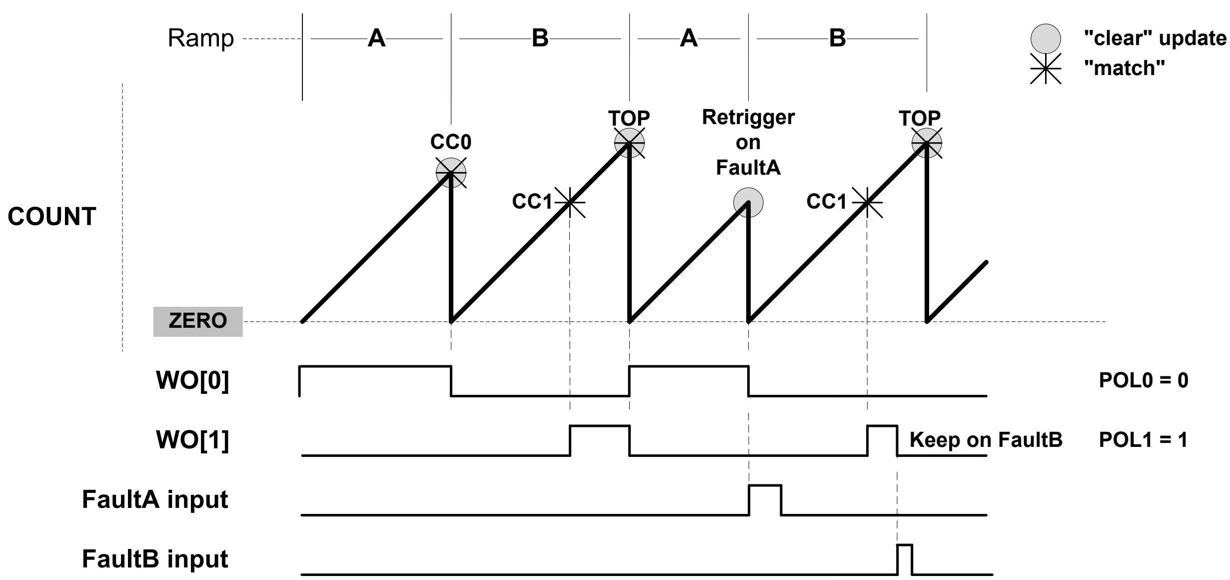


Figure 48-22. RAMP2 Critical Operation With 2 Channels



Critical Swapped RAMP2 (RAMP2C) Operation

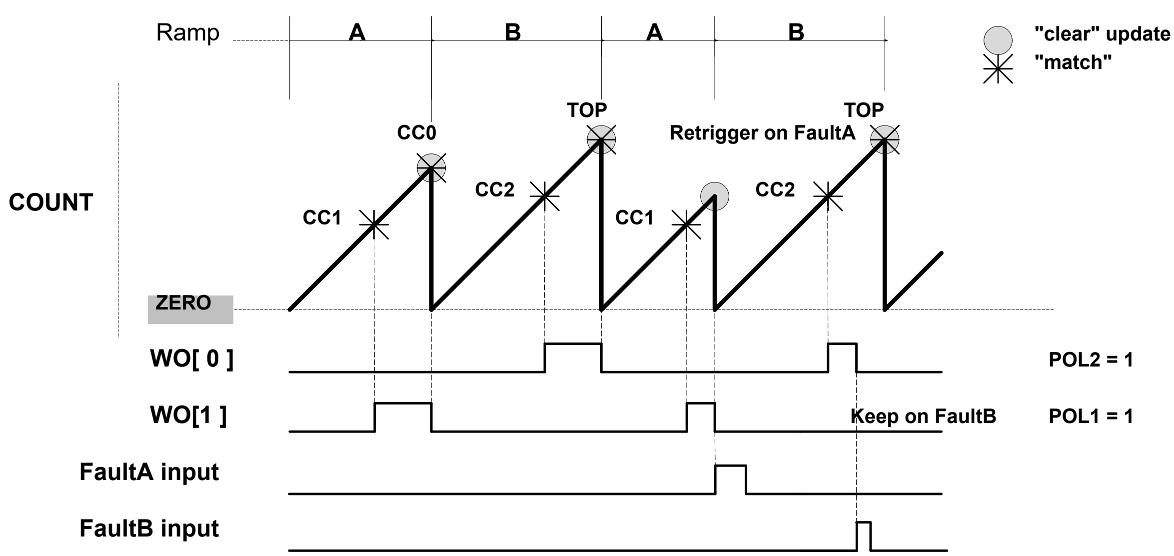
In RAMP2CS variant, WO[0] and W0[1] active ramp are inverted, WO[0] is active on RAMPB and WO[1] is active on RAMPA. The RAMP2CS operation is similar to RAMP2C operation and does not use circular buffers in the implementation. The period and duty cycle of output is controlled as shown below.

- PER – Defines the period for cycle B (Ramp B)
- CC0 – Defines the period for cycle A (Ramp A)
- CC1 – Defines the duty cycle for cycle A (Ramp A)
- CC2 – Defines the duty cycle for cycle B (Ramp B)

RAMP2 Critical Swap Operation illustrates the RAMP2CS operation. The following steps are performed to operate the TCC in this operation.

1. Configure the clock source for the TCC Instance in the Main Clock Controller (MCLK) and enable the APB BUS clock for the TCC Instance by writing a '1' to the TCCx_ bit in the APB Mask register of the MCLK (i.e. enable CLK_TCC0_APB for TCC0 by setting TCC0_ bit (APBCMASK<3>), CLK_TCC1_APB for TCC1 by setting TCC1_ bit (APBCMASK<4>) etc.).
2. Enable Generic clock for TCC Instance (e.g. enable GCLK_TCC0 for TCC0 by setting PCHCTRL31.CHEN bit PCHCTRL31<6>, GCLK_TCC1 for TCC1 by setting PCHCTRL32.CHEN bit PCHCTRL32<6> etc.).
3. Select desired prescaler by setting CTRLA.PRESCALER bits (CTRLA<10:8>).
4. Set waveform generation operation to Normal Pulse Width Modulation Operation (NPWM) by setting WAVE.WAVEGEN bits (WAVE<2:0> = 2).
5. Set Ramp operation to RAMP2CS by setting value of WAVE.RAMP bits (WAVE<6:4>) to 0x4.
6. Load Compare/Capture (CC1<31:0>) register with the desired PWM duty cycle value for Cycle A (Ramp A).
7. Load Compare/Capture buffer register (CC2<31:0>) register with the desired PWM duty cycle value for Cycle B (Ramp B).
8. Load the period register PER<31:0> with the desired time period value (TOP) for cycle B.
9. Load the Compare/Capture register (CC0<31:0>) with the desired time period value (TOP) for cycle A.
10. Set Counter to count in up direction by clearing CTRLBCLR.DIR bit (CTRLBCLR<0> = 1).
11. In this operation the output polarity of a selected output signal should be set as follows:
    a. Set WAVE.POL2 bit (WAVE<18>) as this controls WO[0] output.
    b. Set WAVE.POL1 bit (WAVE<17>) as this controls WO[1] output
12. If overflow interrupt is used, set INTENSET.OVF bit (INTENSET<0>), if compare match interrupt is used, set INTENSET.MC1 bit (INTENSET<17>) for Ramp A match, INTENSET.MC2 bit (INTENSET<18>) for Ramp B match interrupt. Configure the NVIC by setting group priority, sub priority and enabling corresponding TCCx IRQ.
13. Enable TCC by setting CTRLA.ENABLE bit (CTRLA<1>).
14. For information on how to clear interrupts, see Interrupts.

Figure 48-23. RAMP2 Critical Swap Operation



48.6.3.5. 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 (TCCx_MC_0 and TCCx_MC_1) 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 operation.

Fault Filtering

There are three filters available for both Fault A and Fault B input. They are configured by the corresponding Recoverable Fault Configuration registers (FCTRLA and FCTRLB). The three filters can either be used independently or in any combination.

Input Filtering

When disabled (FCTRLx.FILTERVAL = 0), the action of a fault on the output port is asynchronous. This helps the fault action to be performed on the output of the compare channel, even when the fault is caused by the loss of the system clock. Therefore, when a fault occurs, the system will immediately and asynchronously disable the compare channel output as long as a fault is present. To avoid TCC control block corruption by a glitch on a fault input line, the fault action on the TCC control block (retrigger, capture, etc..) is synchronized on internal clock domain. A digital filter can be enabled and configured by the Fault Filter Value bits in the Fault Configuration registers (FCTRLA.FILTERVAL (FCTRLA <28:24>) and FCTRLB.FILTERVAL (FCTRLB <28:24>)). If the fault width is less than the FILTERVAL (in TCC clock cycles), will be ignored. A valid fault event action on the TCC FSM, will be then delayed by the clock cycles represented by the FILTERVAL.

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 A or Fault B Blanking Mode bits in the Recoverable Fault Configuration registers (FCTRLA.BLANK (FCTRLA <6:5>) or FCTRLB.BLANK (FCTRLB <6:5>)). The desired duration of the blanking must be written to the Fault Blanking Time bits (FCTRLA.BLANKVAL (FCTRLA <23:16>) or FCTRLB.BLANKVAL (FCTRLB <23:16>)). The blanking time t_b is calculated by

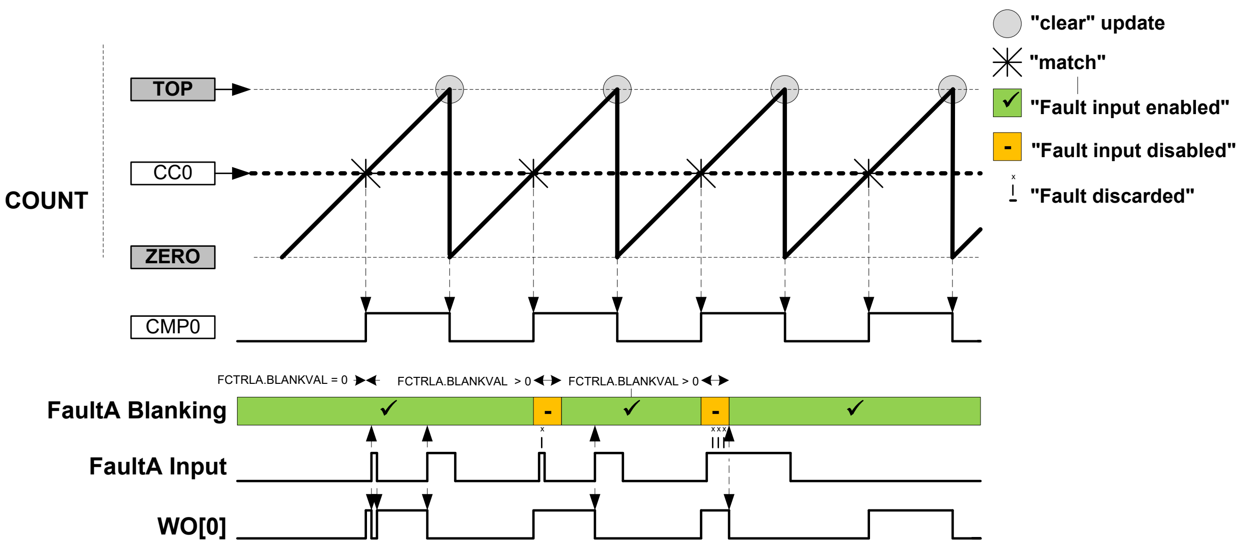
$$t_b = \frac{1 + \mathrm{BLANKVAL}}{f_{\mathrm{GCLK\_TCCx\_PRESC}}}$$

Here, $f_{\mathrm{GCLK\_TCCx\_PRESC}}$ is the frequency of the prescaled peripheral clock frequency $f_{\mathrm{GCLK\_TCCx}}$.

The prescaler is enabled by writing '1' to the Fault A or Fault B Blanking Prescaler bit (FCTRLA.BLANKPRESC(FCTRLA <15>) or FCTRLB.BLANKPRESC(FCTRLB <15>)). When disabled, $f_{\mathrm{GCLK\_TCCx\_PRESC}}=f_{\mathrm{GCLK\_TCCx}}$. When enabled, $f_{\mathrm{GCLK\_TCCx\_PRESC}}=f_{\mathrm{GCLK\_TCCx}}/64$.

The maximum blanking time (FCTRLA.BLANKVAL (FCTRLA <23:16>) or FCTRLB.BLANKVAL (FCTRLB <23:16>))=
255) at $f_{\mathrm{GCLK\_TCCx}}$=96MHz is 2.67µs (no prescaler) or 170µs (prescaling). For $f_{\mathrm{GCLK\_TCCx}}$=1MHz, the maximum blanking time is either 170µs (no prescaling) or 10.9ms (prescaling enabled).

Figure 48-24. Fault Blanking in RAMP1 Operation with Inverted Polarity



Fault Qualification This is enabled by writing a '1' to the Fault A or Fault B Qualification bit in the Recoverable Fault A or Fault B Configuration register ((FCTRLA.QUAL(FCTRLA <4>) or FCTRLB.QUAL(FCTRLB <4>)). When the recoverable fault qualification is enabled (FCTRLA.QUAL(FCTRLA <4>) =1 or FCTRLB.QUAL(FCTRLB <4>)=1, the fault input is disabled all the time and the corresponding channel output has an inactive level, as shown in the figures below.

Figure 48-25. Fault Qualification in RAMP1 Operation

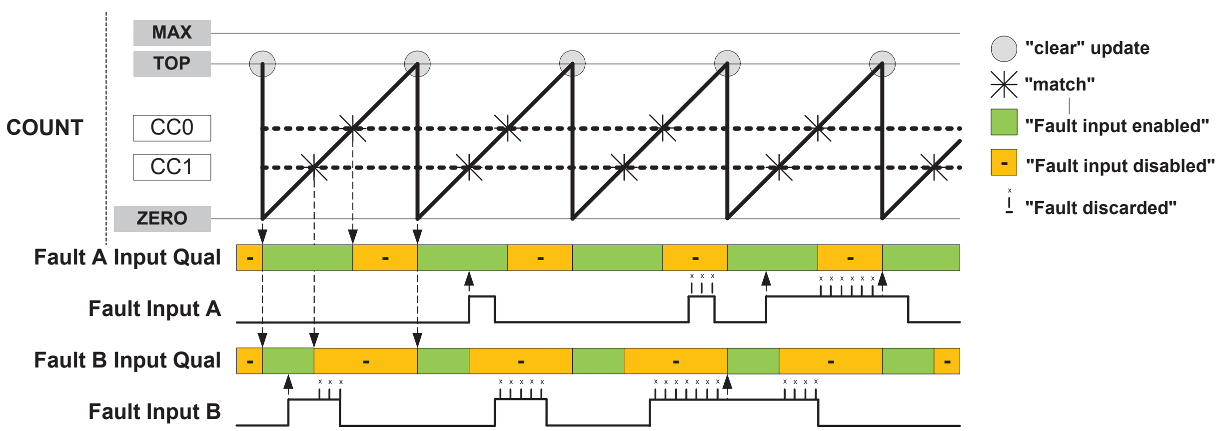
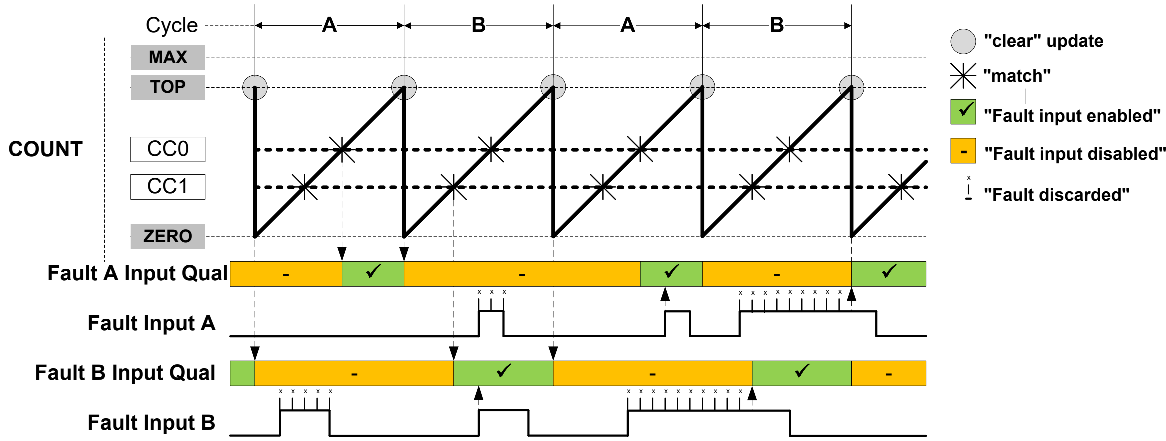


Figure 48-26. Fault Qualification in RAMP2 Operation with Inverted Polarity

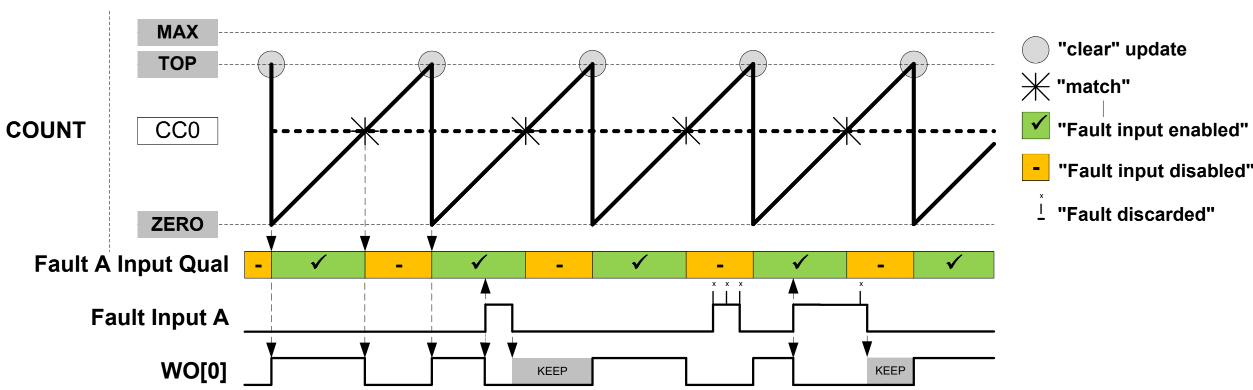


Fault Actions

Different fault actions can be configured individually for Fault A and Fault B. Most fault actions are not mutually exclusive; therefore 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 A or Fault B Keeper bit in the corresponding Recoverable Fault A or Fault B Configuration register (FCTRLA.KEEP(FCTRLA <3>) or FCTRLB.KEEP(FCTRLB <3>) 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 48-27. Waveform Generation with Fault Qualification and Keep Action



Restart Action This is enabled by writing the Fault A or Fault B Restart bit in corresponding Recoverable Fault A or Fault B Configuration register (FCTRLA.RESTART(FCTRLA <7>) =1 or FCTRLB.RESTART(FCTRLB <7>) 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 *Waveform Generation in RAMP1 Operation with Restart Action*. In RAMP1 operation, 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 *Waveform Generation in RAMP2 Operation with Restart Action*. 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 48-28. Waveform Generation in RAMP1 Operation with Restart Action

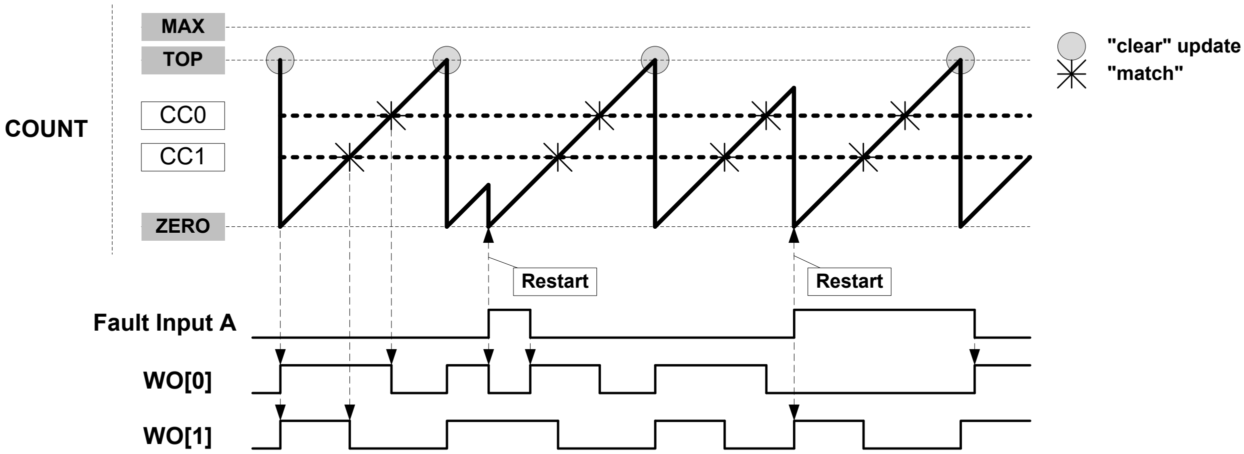
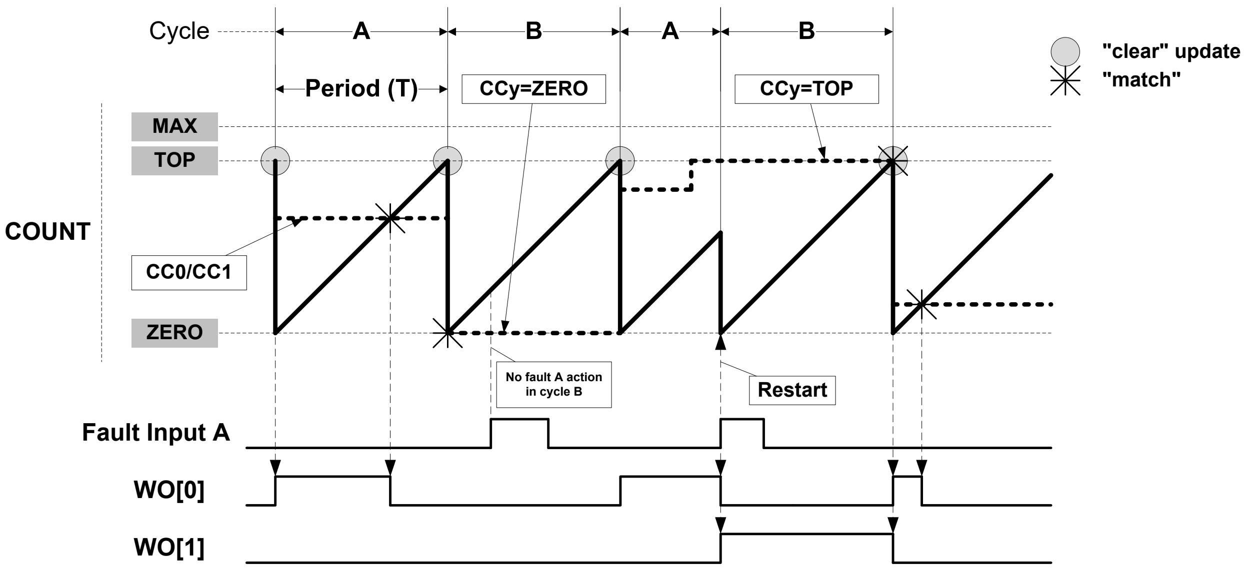


Figure 48-29. Waveform Generation in RAMP2 Operation with Restart Action



Capture Action

Several capture actions can be selected by writing the Fault A or Fault B Capture Action bits in the Fault A or Fault B Control register (FCTRLA.CAPTURE(FCTRLA <14:12>) or FCTRLB.CAPTURE(FCTRLB <14:12>). When one of the capture operations is selected, the counter value is captured when the fault occurs. The following capture operations are available:

- CAPT - the equivalent to a standard capture operation, for further details refer to *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 *Capture Action "CAPTMAX"* in the FCTRLA or FCTRLB register.
- 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 *Capture Action “DERIV0”* in the FCTRLA or FCTRLB register.

CCy Content:

In CAPTMIN and CAPTMAX operations, CCy keeps the respective extreme captured values, see *Capture Action “CAPTMAX”*. In LOCMIN, LOCMAX or DERIV0 operation, CCy follows the counter value at fault time, see *Capture Action “DERIV0”*.

Before enabling CAPTMIN or CAPTMAX mode of capture, the user must initialize the corresponding CCy register value to a value different from zero (for CAPTMIN) and TOP (for CAPTMAX). If the initial value of the CCy register is zero for CAPTMIN and TOP for CAPTMAX, no captures will be performed using the corresponding channel.

MCy Behaviour:

In LOCMIN and LOCMAX operation, capture is performed on each capture event. The MCy interrupt flag is set only when the captured value is above or equal for LOCMIN and 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 MCy 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 MCy interrupt flag is set only when on capture event time, the counter value is greater than or equal (for CAPTMIN), or less than 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 A and Fault B and each dedicated CCy channel capture counter value. In other modes, an interrupt is only generated on an extreme captured value.

Figure 48-30. Capture Action “CAPTMAX”

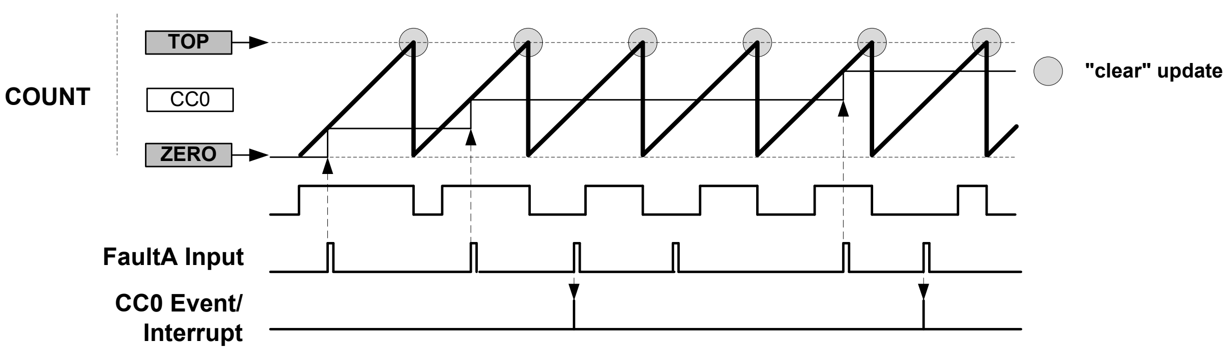
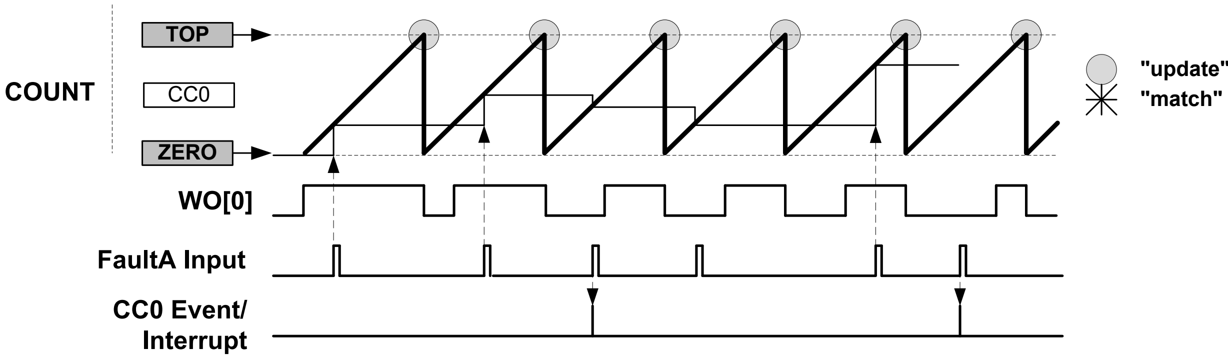


Figure 48-31. Capture Action "DERIV0"



Hardware Halt Action This is configured by writing 0x1 to the Fault A or Fault B Halt mode bits in the Recoverable Fault A or Fault B Configuration registers (FCTRLA.HALT(FCTRLA <9:8>) or FCTRLB.HALT(FCTRLB <9:8>). 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 48-32. Waveform Generation with Halt and Restart Actions

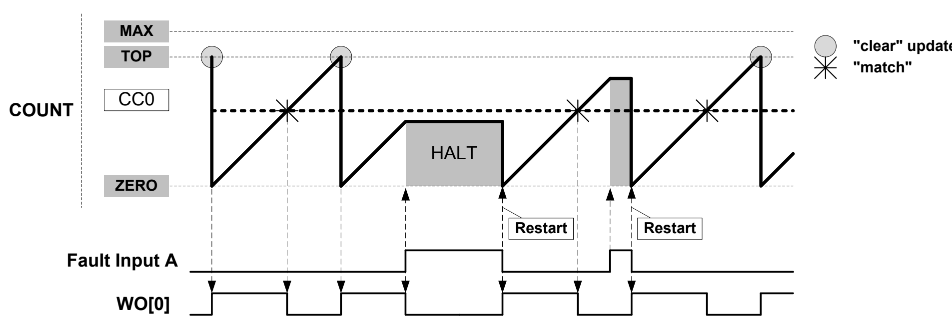
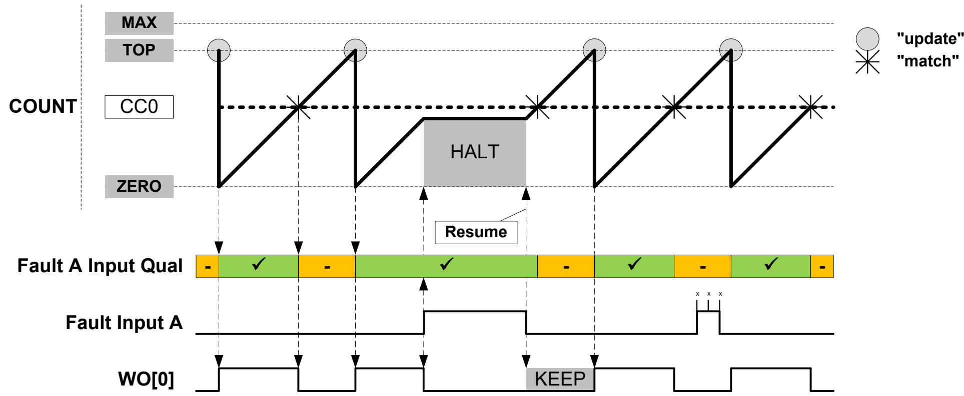
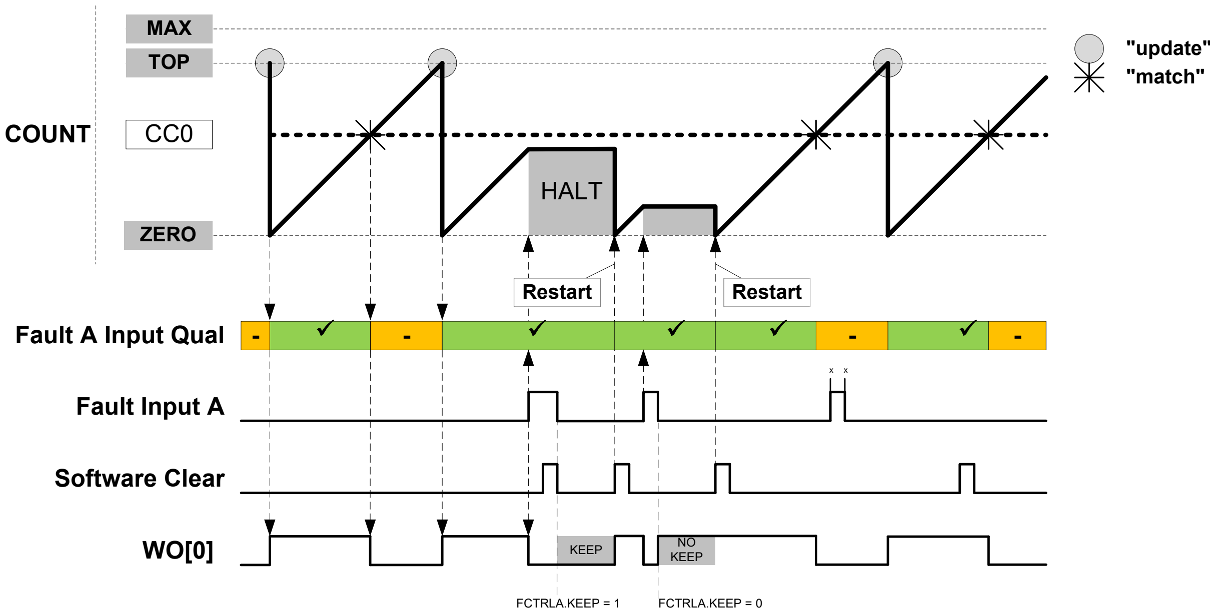


Figure 48-33. Waveform Generation with Fault Qualification, Halt, and Restart Actions



Software Halt Action This action is configured by writing 0x2 to the Fault A and Fault B Halt mode bits in the Recoverable Fault A or Fault B configuration registers (FCTRLA.HALT(FCTRLA <9:8>) or FCTRLB.HALT(FCTRLB <9:8>). 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 A or Fault B bit in the STATUS register must be cleared by software. See the following figure.

Figure 48-34. Waveform Generation with Software Halt, Fault Qualification, Keep and Restart Actions



48.6.3.6. 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.NREy and DRVCTRL.NRVy). The non-recoverable fault input (TCCx_EV_0 and TCCx_EV_1) actions are enabled in Event Control register (EVCTRL.EVACT0(EVCTRL <2:0>) and EVCTRL.EVACT1(EVCTRL <5:3>)).

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 Filter Value bits in the Driver Control register DRVCTRL.FILTERVAL0 (DRVCTRL<27:24>) and DRVCTRL.FILTERVAL1 (DRVCTRL<31:28>) . 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 (DGBCTRL <2>)) 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 (CTRLBSET <1>)=1 and the Ramp index or counter direction changes, a non-recoverable Update Fault State and the respective interrupt (UFS) are generated.

48.6.3.7. Time-Stamp Capture

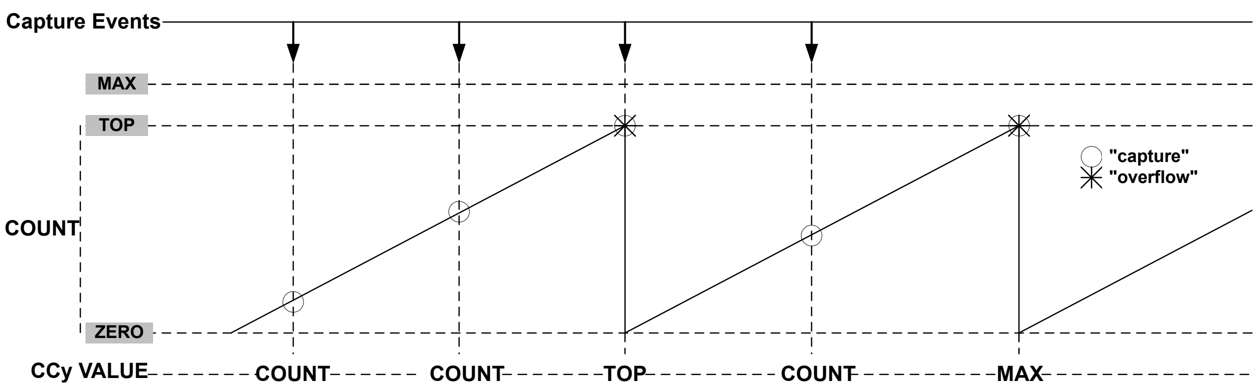
This feature is enabled when the Capture Time Stamp (STAMP) Event Action in Event Control register (EVCTRL.EVACT0(EVCTRL <2:0>)=0x6) is selected. The counter TOP value must be smaller than MAX.

When a capture event is detected, the COUNT value is copied into the corresponding Channel y Compare/Capture Value (CCy) register. In case of an overflow, the MAX value is copied into the corresponding CCy register.

When a valid captured value is present in the capture channel register, the corresponding Capture Channel y Interrupt Flag (INTFLAG.MCy) 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.MCy) is still set, the new time-stamp will not be stored and INTFLAG.ERR will be set.

Figure 48-35. Time-Stamp



48.6.3.8. Waveform Extension

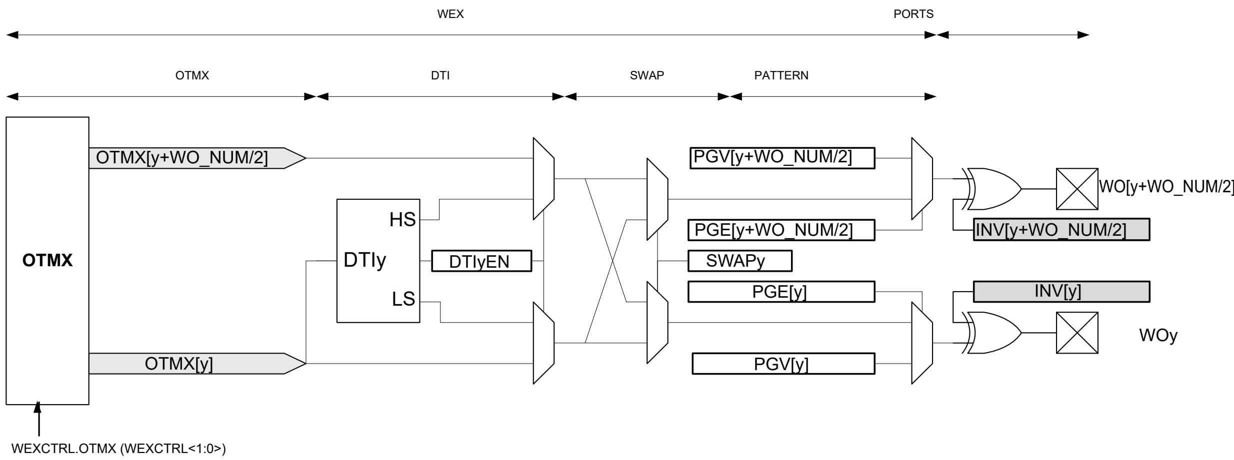
Waveform Extension Stage Details shows a schematic diagram of actions of the four optional units that follow the recoverable fault stage on a port pin pair: Output Matrix (OTMX), Dead-Time Insertion (DTI), SWAP and Pattern Generation. The DTI and SWAP units can be seen as a four port pair slices:

- 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 y DTIy / SWAPy acting on port pins (WO[y], WO[WO_NUM/2 +y])

Figure 48-36. Waveform Extension Stage Details



The **output matrix (OTMX)** unit distributes compare channels, according to the selectable configurations in the following table. This is configured by WEXCTRL.OTMX bits (WEXCTRL <1:0>) as shown below.

Table 48-4. Output Matrix Channel Pin Routing Configuration

Value	OTMX[7]	OTMX[6]	OTMX[5]	OTMX[4]	OTMX[3]	OTMX[2]	OTMX[1]	OTMX[0]
0x0	CC1	CC0	CC5	CC4	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

The following notes refer to this pin routing 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 table below is an example showing four compare channels on four outputs.

Table 48-5. Four Compare Channels on Four Outputs

Value	OTMX[3]	OTMX[2]	OTMX[1]	OTMX[0]
0x0	CC3	CC2	CC1	CC0
0x1	CC1	CC0	CC1	CC0
0x2	CC0	CC0	CC0	CC0

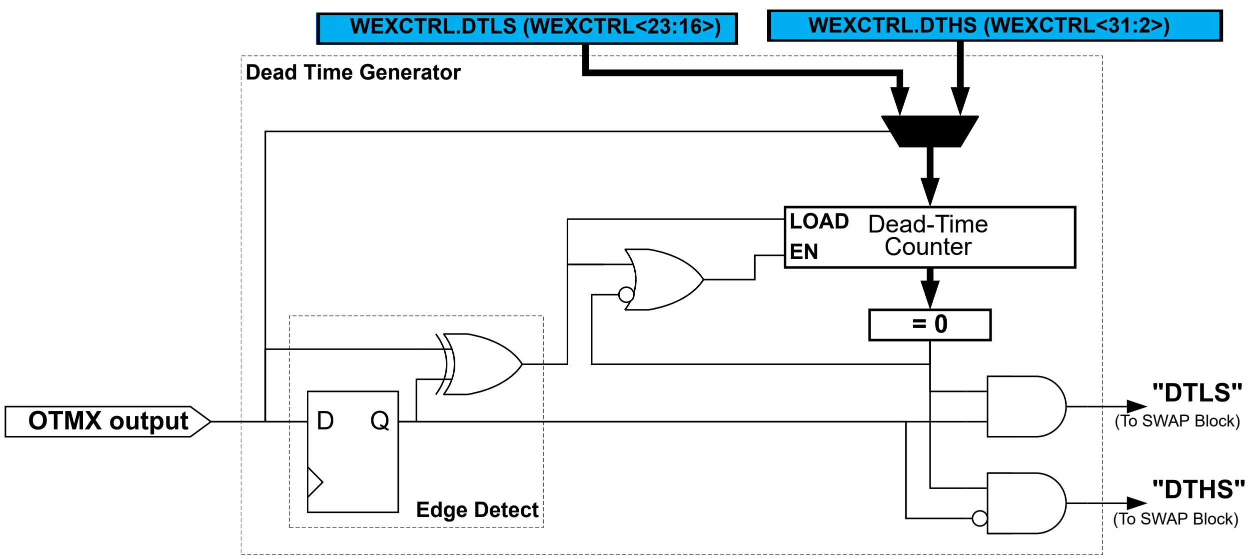
Table 48-5. Four Compare Channels on Four Outputs (continued)

Value	OTMX[3]	OTMX[2]	OTMX[1]	OTMX[0]
0x3	CC1	CC1	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 outputs (DTLS and DTHS) will never switch simultaneously.

The DTI stage consists of four equal dead-time insertion generators; one for each of the first four compare channels. *Dead-Time Generator Block Diagram* 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.

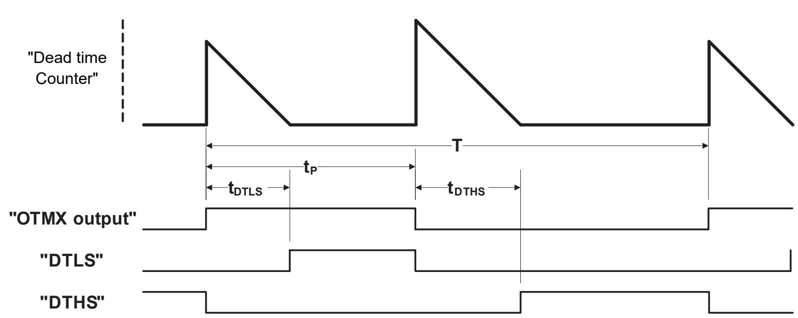
Figure 48-37. Dead-Time Generator Block Diagram



As shown in *Dead-Time Generator Timing Diagram*, 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.

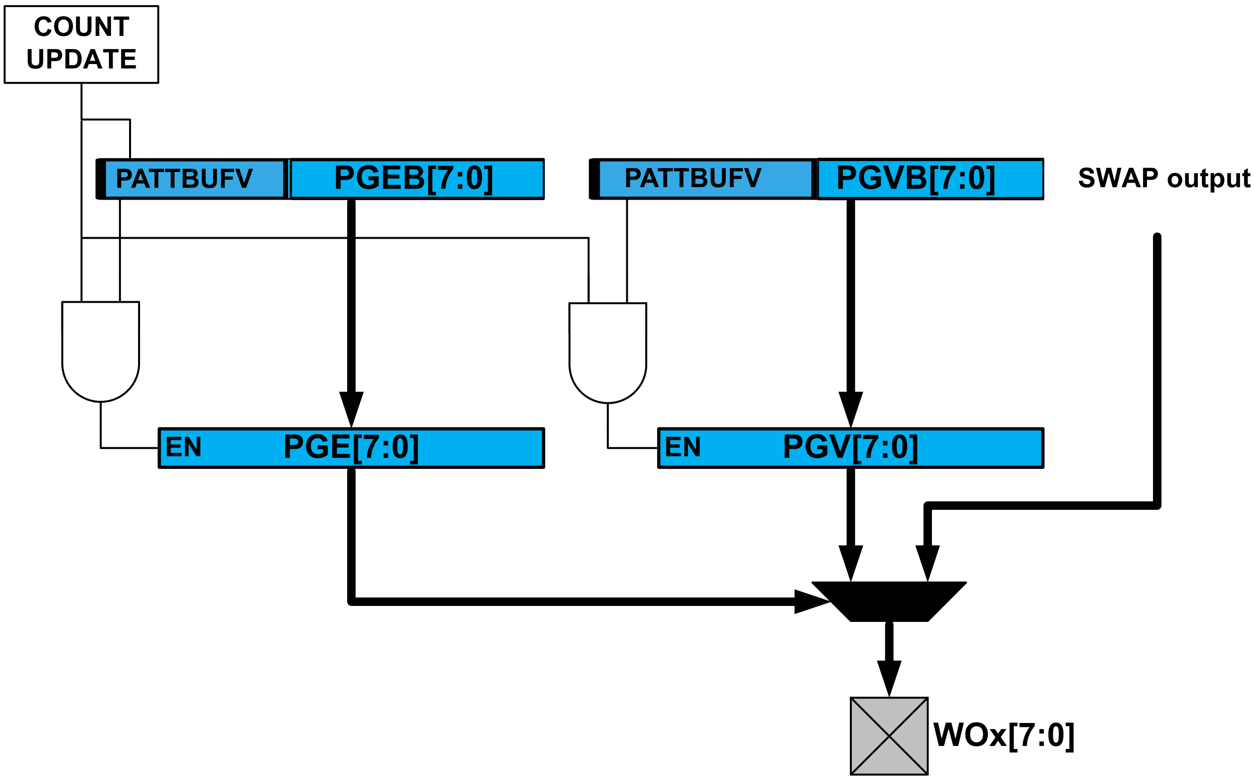
In the following figure t_P shows the lower side output from OTMX and T is the period of OTMX output waveform. Dead-time insertion for lower and upper side outputs are shown by t_{DTLS} and t_{DTHS} respectively.

Figure 48-38. 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 *Pattern Generator Block Diagram*.

Figure 48-39. 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 PGV[7:0] bits (PATT <15:8>) , PGE[7:0] bits (PATT <7:0>) registers.

48.6.4. DMA, Interrupts, and Events

The following table shows the DMA request for the TCC module.

Table 48-6. Module Requests for TCC

Condition	Interrupt request	Event output	Event input	DMA request	DMA request is cleared
Overflow / Underflow	Yes	Yes		Yes(1)	On DMA acknowledge
Channel Compare Match or Capture	Yes	Yes	Yes(2)	Yes(3)	For circular buffering: on DMA acknowledge For capture channel: when CCy register is read
Retrigger	Yes	Yes			
Count	Yes	Yes			
Capture Overflow Error	Yes				
Debug Fault State	Yes				
Recoverable Faults	Yes				
Non-Recoverable Faults	Yes				

Table 48-6. Module Requests for TCC (continued)

Condition	Interrupt request	Event output	Event input	DMA request	DMA request is cleared
TCCx Event 0 input			Yes(4)		
TCCx Event 1 input			Yes(5)		

Notes:
1. DMA request set on Overflow, Underflow or Re-trigger conditions.
2. Can perform capture or generate recoverable fault on an event input.
3. In Capture or Circular modes.
4. On event input, either action can be executed:
   – re-trigger counter
   – control counter direction
   – stop the counter
   – decrement the counter
   – perform period and pulse width capture
   – generate non-recoverable fault
5. On event input, either action can be executed:
   – re-trigger counter
   – increment or decrement counter depending on direction
   – start the counter
   – increment or decrement counter based on direction
   – increment counter regardless of direction
   – generate non-recoverable fault

48.6.4.1. 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(CTRLA<23>)) is written to '0', the TCC generates a DMA request on each cycle when an update condition (Overflow, Underflow or Re-trigger) is detected. When an update condition (Overflow, Underflow or Re-trigger) is detected while CTRLA.DMAOS(CTRLA<23>)=1, the TCC generates a DMA trigger on the cycle following the DMA One-Shot Command written to the Control B register (CTRLBSET.CMD(CTRLBSET<7:5>)=DMAOS). In both cases, the request is cleared by hardware on DMA acknowledge.
Channel Match (MCy)	A DMA request is set only on a compare match if CTRLA.DMAOS(CTRLA<23>)=0. The request is cleared by hardware on DMA acknowledge. When CTRLA.DMAOS(CTRLA<23>)=1, the DMA requests are not generated.
Channel Capture (MCy)	For a capture channel, the request is set when valid data is present in the CCy register, and cleared once the CCy register is read. In this operation mode, the CTRLA.DMAOS bit (CTRLA<23>) value is ignored.

Important: DMA One-Shot mode is not available in RAMP1/RAMP2C/RAMP2CS modes.

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.

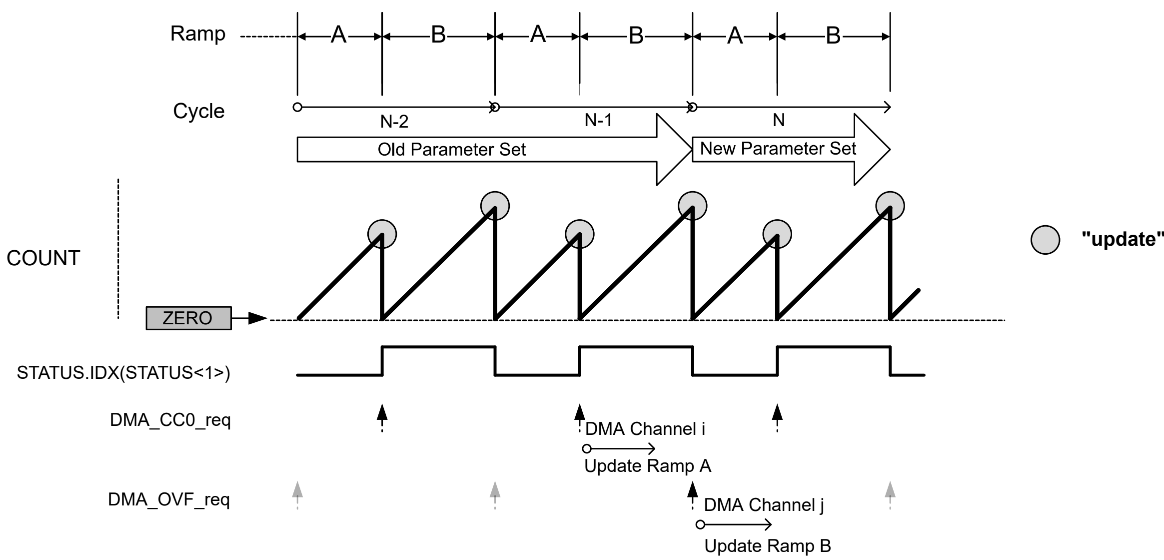
DMA Operation with Circular Buffer in RAMP2 and RAMP2A Operation

When a CC0 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 (Compare Match) 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 48-40. DMA Triggers in RAMP2 Operation and Circular Buffer Enabled



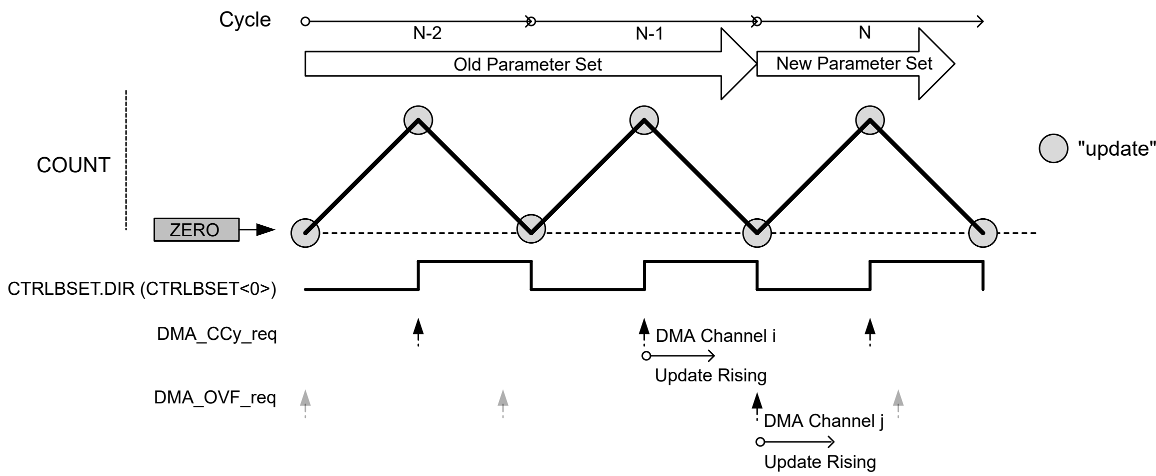
DMA Operation with Circular Buffer in DSBOTH

When CC0 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 (Compare Match) trigger. When down-counting, all circular buffer values can be updated through a second DMA channel, triggered by the OVF DMA request.

Figure 48-41. DMA Triggers in DSBOTH Operation and Circular Buffer Enabled



48.6.4.2. Interrupts

The TCC has the following interrupt sources (refer to the INTENSET register):

- Overflow/Underflow (OVF)
- Retrigger (TRG)
- Count (CNT) - refer also to description of EVCTRL.CNTSEL.
- Capture Overflow Error (ERR)
- Non-Recoverable Update Fault (UFS)
- Debug Fault State (DFS)
- Recoverable Faults (FAULTA and FAULTB)
- Non-recoverable Faults (FAULT0 and FAULT1)
- Compare Match or Capture Channels (MCy)

These interrupts are asynchronous wake-up sources. See Sleep Mode Entry and Exit Table in PM/ Sleep Mode Controller section for details.

References:

- PM_Sleep Mode Controller

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. Refer to the INTFLAG for details on how to clear Interrupt flags. The TCC has one common interrupt request line for OVF, ERR, UFS, DFS, FAULTA, FAULTB, FAULT0, and FAULT1 interrupt sources. It contains a separate line for both TRG, and CNT interrupt sources and another line for all Compare Match or Capture Channels (MCy) 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.

References:

- Nested Vector Interrupt Controller

48.6.4.3. 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: MCy

Writing a '1' or '0' to an Event Output bit in the Event Control Register (OVFEO, TRGEO, CNTEO and MCEOy bits) enables or disables the corresponding output event. Refer also to *EVSYS – Event System*.

The TCC can take the following actions on a channel input event (MCy):

- Capture event
- Generate a recoverable or non-recoverable fault

The TCC can take the following actions on counter Event 1 (TCCx_EV_1):

- Counter re-trigger
- Counter direction control
- Stop the counter
- Decrement the counter on event
- Period and pulse width capture
- Non-recoverable fault

The TCC can take the following actions on counter Event 0 (TCCx_EV_0):

- Counter re-trigger
- Count on event (increment or decrement, depending on counter direction)
- Counter start - start counting on the event rising edge. Further events will not restart the counter; the counter will keep 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.
- Non-recoverable fault

The counter Event Actions are available in the Event Control registers (EVCTRL.EVACT0(EVCTRL <2:0>) and EVCTRL.EVACT1(EVCTRL <5:3>)). For further details, refer to EVCTRL.

Writing a '1' to an Event Input bit in the Event Control register (EVCTRL.MCEIy or EVCTRL.TCEIn where n=0,1) enables the corresponding action on input event, where writing '0' disables the 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.

Reference:

- EVSYS

48.6.5. 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 TCC 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.

References:

- EVSYS

48.7. Register Summary

For descriptions and definitions of both Register and bitfield properties, refer to Register Properties.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRLA	31:24			CPTEN5	CPTEN4	CPTEN3	CPTEN2	CPTEN1	CPTEN0
		23:16	DMAOS							FCYCLE
		15:8		ALOCK	PRESCYNC[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			CC5	CC4	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	MCEO7	MCEO6	MCEO5	MCEO4	MCEO3	MCEO2	MCEO1	MCEO0
		23:16	MCEI7	MCEI6	MCEI5	MCEI4	MCEI3	MCEI2	MCEI1	MCEI0
		15:8	TCEI1	TCEI0	TCINV1	TCINV0		CNTEO	TRGEO	OVFEO
		7:0	CNTSEL[1:0]		EVACT1[2:0]			EVACT0[2:0]		
0x24	INTENCLR	31:24								
		23:16	MC7	MC6	MC5	MC4	MC3	MC2	MC1	MC0
		15:8	FAULT1	FAULT0	FAULTB	FAULTA	DFS	UFS		
		7:0					ERR	CNT	TRG	OVF
0x28	INTENSET	31:24								
		23:16	MC7	MC6	MC5	MC4	MC3	MC2	MC1	MC0
		15:8	FAULT1	FAULT0	FAULTB	FAULTA	DFS	UFS		
		7:0					ERR	CNT	TRG	OVF
0x2C	INTFLAG	31:24								
		23:16	MC7	MC6	MC5	MC4	MC3	MC2	MC1	MC0
		15:8	FAULT1	FAULT0	FAULTB	FAULTA	DFS	UFS		
		7:0					ERR	CNT	TRG	OVF
0x30	STATUS	31:24			CMP5	CMP4	CMP3	CMP2	CMP1	CMP0
		23:16			CCBUFV5	CCBUFV4	CCBUFV3	CCBUFV2	CCBUFV1	CCBUFV0
		15:8	FAULT1	FAULT0	FAULTB	FAULTA	FAULT1IN	FAULT0IN	FAULTBIN	FAULTAIN
		7:0	PERBUFV		PATTBUFV		DFS	UFS	IDX	STOP
0x34	COUNT	31:24	COUNT[31:24]							
		23:16	COUNT[23:16]							
		15:8	COUNT[15:8]							
		7:0	COUNT[7:0]							

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x38	PATT	15:8	PGV[7:0]							
		7:0	PGE[7:0]							
0x3A ... 0x3B	Reserved									
0x3C	WAVE	31:24					SWAP3	SWAP2	SWAP1	SWAP0
		23:16	POL7	POL6	POL5	POL4	POL3	POL2	POL1	POL0
		15:8					CICCEN3	CICCEN2	CICCEN1	CICCEN0
		7:0	CIPEREN	RAMP[2:0]				WAVEGEN[2:0]		
0x40	PER	31:24	PER[25:18]							
		23:16	PER[17:10]							
		15:8	PER[9:2]							
		7:0	PER[1:0]		DITHER[5:0]					
0x44	CC0	31:24	CC[25:18]							
		23:16	CC[17:10]							
		15:8	CC[9:2]							
		7:0	CC[1:0]		DITHER[5:0]					
0x48	CC1	31:24	CC[25:18]							
		23:16	CC[17:10]							
		15:8	CC[9:2]							
		7:0	CC[1:0]		DITHER[5:0]					
0x4C	CC2	31:24	CC[25:18]							
		23:16	CC[17:10]							
		15:8	CC[9:2]							
		7:0	CC[1:0]		DITHER[5:0]					
0x50	CC3	31:24	CC[25:18]							
		23:16	CC[17:10]							
		15:8	CC[9:2]							
		7:0	CC[1:0]		DITHER[5:0]					
0x54	CC4	31:24	CC[25:18]							
		23:16	CC[17:10]							
		15:8	CC[9:2]							
		7:0	CC[1:0]		DITHER[5:0]					
0x58	CC5	31:24	CC[25:18]							
		23:16	CC[17:10]							
		15:8	CC[9:2]							
		7:0	CC[1:0]		DITHER[5:0]					
0x5C ... 0x63	Reserved									
0x64	PATTBUF	15:8	PGVB[7:0]							
		7:0	PGEB[7:0]							
0x66 ... 0x6B	Reserved									
0x6C	PERBUF	31:24	PERBUF[25:18]							
		23:16	PERBUF[17:10]							
		15:8	PERBUF[9:2]							
		7:0	PERBUF[1:0]		DITHERBUF[5:0]					
0x70	CCBUF0	31:24	CCBUF[25:18]							
		23:16	CCBUF[17:10]							
		15:8	CCBUF[9:2]							
		7:0	CCBUF[1:0]		DITHERBUF[5:0]					
0x74	CCBUF1	31:24	CCBUF[25:18]							
		23:16	CCBUF[17:10]							
		15:8	CCBUF[9:2]							
		7:0	CCBUF[1:0]		DITHERBUF[5:0]					

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x78	CCBUF2	31:24	CCBUF[25:18]							
		23:16	CCBUF[17:10]							
		15:8	CCBUF[9:2]							
		7:0	CCBUF[1:0]		DITHERBUF[5:0]					
0x7C	CCBUF3	31:24	CCBUF[25:18]							
		23:16	CCBUF[17:10]							
		15:8	CCBUF[9:2]							
		7:0	CCBUF[1:0]		DITHERBUF[5:0]					
0x80	CCBUF4	31:24	CCBUF[25:18]							
		23:16	CCBUF[17:10]							
		15:8	CCBUF[9:2]							
		7:0	CCBUF[1:0]		DITHERBUF[5:0]					
0x84	CCBUF5	31:24	CCBUF[25:18]							
		23:16	CCBUF[17:10]							
		15:8	CCBUF[9:2]							
		7:0	CCBUF[1:0]		DITHERBUF[5:0]					

48.7.1. Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00000000
Property: PAC Write-Protection, Write-Synchronized (ENABLE, SWRST)

Table 48-7. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
			CPTEN5	CPTEN4	CPTEN3	CPTEN2	CPTEN1	CPTEN0
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	DMAOS							FCYCLE
Access	R/W							R/W
Reset	0							0

Bit	15	14	13	12	11	10	9	8
		ALOCK	PRESCYNC[1:0]		RUNSTDBY	PRESCALER[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
		RESOLUTION[1:0]					ENABLE	SWRST
Access		R/W	R/W				R/W	R/W
Reset		0	0				0	0

Bits 24, 25, 26, 27, 28, 29 – CPTENy Capture Channel y Enable
These bits are used to select the capture or compare operation on channel y (where y=0,1,2...5).
Writing a '1' to CPTENy enables capture on channel y.
Writing a '0' to CPTENy disables capture on channel y.
All these bits are enable-protected.

Bit 23 – DMAOS DMA One-Shot Trigger Mode
This bit enables the DMA One-shot Trigger Mode.
Note: This bit is enable-protected.

Value	Description
0	Generate DMA triggers on each TCC cycle
1	Generate a DMA trigger on TCC cycle following a CTRLBSET.CMD(CTRLBSET<7:5>) = DMAOS command.

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.
This bit is enable-protected.

Value	Description
0	The stop condition is evaluated immediately.
1	The stop condition is evaluated at the end of the cycle.

Bit 14 – ALOCK Auto Lock

This bit is enable-protected.

Value	Description
0	The Lock Update bit in the Control B register (CTRLBSET.LUPD (CTRLBSET<1>)) is not affected by overflow/ underflow, and re-trigger events
1	CTRLBSET.LUPD (CTRLBSET<1>) is set to '1' on each overflow/underflow or re-trigger event.

Bits 13:12 – PRESCYNC[1:0] Prescaler and Counter Synchronization

These bits select if on re-trigger event, the Counter is cleared or reloaded on either the next GCLK_TCCx clock, or on the next prescaled GCLK_TCCx clock. It is also possible to reset the prescaler on re-trigger event.
These bits are enable-protected.

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.
These bits are enable-protected.

Value	Description
0	The TCC is halted in standby mode.
1	The TCC continues to run in standby mode.

Bits 10:8 – PRESCALER[2:0] Prescaler

These bits select the Counter prescaler factor.
These bits are enable-protected.

Value	Name	Description
0x0	DIV1	Prescaler: GCLK_TCC
0x1	DIV2	Prescaler: GCLK_TCC/2
0x2	DIV4	Prescaler: GCLK_TCC/4
0x3	DIV8	Prescaler: GCLK_TCC/8
0x4	DIV16	Prescaler: GCLK_TCC/16
0x5	DIV64	Prescaler: GCLK_TCC/64
0x6	DIV256	Prescaler: GCLK_TCC/256
0x7	DIV1024	Prescaler: GCLK_TCC/1024

Bits 6:5 – RESOLUTION[1:0] Dithering Resolution

These bits increase the TCC resolution by enabling the dithering options.
These bits are enable-protected.

Table 48-8. Dithering

Value	Name	Description
0x0	NONE	The dithering is disabled.
0x1	DITH4	Dithering is based on overflow of a 4 bit-counter. PER[3:0] and CCy[3:0] contain dithering pattern selection.
0x2	DITH5	Dithering is based on overflow of a 5 bit-counter. PER[4:0] and CCy[4:0] contain dithering pattern selection.
0x3	DITH6	Dithering is based on overflow of a 6 bit-counter. PER[5:0] and CCy[5:0] contain dithering pattern selection.

Bit 1 – ENABLE Enable

Due to synchronization there is delay between when the CTRLA.ENABLE (CTRLA<1>) is written and the peripheral is enabled/disabled. The value written to CTRLA.ENABLE(CTRLA<1>) will be read back immediately and the ENABLE bit in the SYNCBUSY register (SYNCBUSY.ENABLE(SYNCBUSY<1>)) will be set. SYNCBUSY.ENABLE (SYNCBUSY<1>) will be cleared when the operation is complete.

Value	Description
0	The peripheral is disabled.
1	The peripheral is enabled.

Bit 0 – SWRST Software Reset

Writing a '0' to this bit has no effect.
Writing a '1' to this bit resets all registers in the TCC (except DBGCTRL) to their initial state, and the TCC will be disabled.
Writing a '1' to CTRLA.SWRST(CTRLA<0>) will always take precedence; all other writes in the same write-operation will be discarded.
Due to synchronization there is a delay from writing CTRLA.SWRST(CTRLA<0>) until the reset is complete. CTRLA.SWRST(CTRLA<0>) and SYNCBUSY.SWRST(SYNCBUSY<0>) will both be cleared when the reset is complete.

Notes:

1. When the CTRLA.SWRST is written, the user should poll the SYNCB.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	The Reset operation is ongoing.

48.7.2. 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 (CTRLBSET) register.

Table 48-9. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
	CMD[2:0]			IDXCMD[1:0]		ONESHOT	LUPD	DIR
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:5 – CMD[2:0] TCC Command
These bits can be used for software control of re-triggering and stop commands of the TCC. When a command has been executed, the CMD bit field will read back zero. The commands are executed on the next prescaled GCLK_TCCx clock cycle.
Writing zero to this bit group has no effect.
Writing a '1' to any of these bits will clear the pending command.

Value	Name	Description
0x0	NONE	No action
0x1	RETRIGGER	Clear start, restart or retrigger
0x2	STOP	Force stop
0x3	UPDATE	Force update of double buffered registers
0x4	READSYNC	Force COUNT read synchronization
0x5	DMAOS	One-shot DMA trigger

Bits 4:3 – IDXCMD[1:0] Ramp Index Command
These bits can be used to force cycle A (Ramp A) and cycle B (Ramp B) changes in all RAMP2x 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	No Command: 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.
Reading this bit gives the following status:

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 CTRLBCLR.LUPD (CTRLBCLR<1>) 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.
Reading this bit gives the following status:

Value	Description
0	The CCBUFy, PERBUF, PGVB bits (PATTBUF <15:8>) and PGEV bits (PATTBUF <7:0>) buffer register bitfields *are* copied into the corresponding CCy, PER, PGV bits (PATT <15:8>), and PGE bits (PATT <7:0>) registers and register bitfields on hardware update condition.
1	The CCBUFy, PERBUF, PGVB bits (PATTBUF <15:8>) and PGEV bits (PATTBUF <7:0>) buffer registers bitfields are *not* copied into the corresponding CCy, PER, PGV bits (PATT <15:8>), and PGE bits (PATT <7:0>) registers and register bitfields 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. Reading this bit gives the following status:

Value	Description
0	The timer/counter is counting up (incrementing).
1	The timer/counter is counting down (decrementing).

48.7.3. 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 Set (CTRLBCLR) register.

Table 48-10. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
	CMD[2:0]			IDXCMD[1:0]		ONESHOT	LUPD	DIR
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:5 – CMD[2:0] TCC Command

These bits can be used for software control of re-triggering and stop commands of the TCC. When a command 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 group has no effect
Writing a valid value to this bit group will set the associated command.

Value	Name	Description
0x0	NONE	No action
0x1	RETRIGGER	Force start, restart or retrigger
0x2	STOP	Force stop
0x3	UPDATE	Force update of double buffered registers
0x4	READSYNC	Force a read synchronization of COUNT
0x5	DMAOS	One-shot DMA trigger

Bits 4:3 – IDXCMD[1:0] Ramp Index Command

These bits can be used to force cycle A and cycle B changes in all RAMP2x 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 a zero to these bits has no effect.
Writing a valid value to these bits will set a command.

Value	Name	Description
0x0	DISABLE	No Command: 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 CTRLBSET.LUPD (CTRLBSET<1>) 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 CCBUFy, PERBUF, PGVB bits (PATTBUF <15:8>) and PGEV bits (PATTBUF <7:0>) buffer registers values *are* copied into the corresponding CCy, PER, PGV bits (PATT <15:8>), and PGE bits (PATT <7:0>) registers on hardware update condition.
1	The CCBUFy, PERBUF, PGVB bits (PATTBUF <15:8>) and PGEV bits (PATTBUF <7:0>) buffer registers values are *not* copied into CCy, PER, PGV bits (PATT <15:8>), and PGE bits (PATT <7:0>) registers on hardware update condition.

Bit 0 – DIR Counter Direction

This bit is used to change the direction of the counter.
Note: When TC is counting down, the COUNT register must be initialized to TOP value (PER or CC0 value depending on the mode).

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. Reading this bit gives following status:

Value	Description
0	The timer/counter is counting up (incrementing).
1	The timer/counter is counting down (decrementing).

48.7.4. Synchronization Busy

Name: SYNCBUSY
Offset: 0x08
Reset: 0x00000000
Property: -

Table 48-11. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
			CC5	CC4	CC3	CC2	CC1	CC0
Access			R	R	R	R	R	R
Reset			0	0	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, 12, 13 – CCy Compare/Capture Channel y Synchronization Busy

This bit is cleared when the synchronization of Compare/Capture Channel y register between the clock domains is complete.
This bit is set when the synchronization of Compare/Capture Channel y register between clock domains is started.
CCy bit is available only for existing Compare/Capture Channels. For details on CC channels number, refer to each TCC feature list.
This bit is set when the synchronization of CCy register between clock domains is started.

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/CTRLBCLR register between the clock domains is complete.
This bit is set when the synchronization of CTRLBSET/CTRLBCLR register between clock domains is started.

Bit 1 – ENABLE ENABLE Synchronization Busy
This bit is cleared when the synchronization of ENABLE bit between the clock domains is complete.
This bit is set when the synchronization of ENABLE bit between clock domains is started.

Bit 0 – SWRST SWRST Synchronization Busy
This bit is cleared when the synchronization of SWRST bit between the clock domains is complete.
This bit is set when the synchronization of SWRST bit between clock domains is started.

48.7.5. Fault Control A

Name: FCTRLA
Offset: 0x0C
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Table 48-12. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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 the Fault A (MCE0) event input line. Input signal with pulse width shorter than (FILTERVAL) * GCLK_TCCx is filtered. An Input signal with pulse width larger than (FILTERVAL+1) * GCLK_TCCx is passed to the TCC. The value must be set to zero when this 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(FCTRLA<6:5>)).
When enabled, the fault input source is internally disabled for BLANKVAL* prescaled GCLK_TCCx periods after the detection of the waveform edge.

Bit 15 – BLANKPRESC Recoverable Fault 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_TCCx.
1	Blank time is BLANKVAL* 64 * prescaled GCLK_TCCx.

Bits 14:12 – CAPTURE[2:0] Recoverable Fault A Capture Action
These bits select the capture and Fault A interrupt/event conditions.

Table 48-13. 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 FCTRLA.CHSEL bits (FCTRLA <11:10>) . INTFLAG.FAULTA (INTFLAG<12>) 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 FCTRLA.CHSEL bits (FCTRLA <11:10>) , if COUNT value is lower than the last stored capture value (CC). INTFLAG.FAULTA (INTFLAG<12>) flag rises on each local minimum detection.
0x3	CAPTMAX	On rising edge of a valid recoverable Fault A, capture counter value on channel selected by FCTRLA.CHSEL bits (FCTRLA <11:10>) , if COUNT value is higher than the last stored capture value (CC). INTFLAG.FAULTA (INTFLAG<12>)flag rises on each local maximum detection.
0x4	LOCMIN	On rising edge of a valid recoverable Fault A, capture counter value on channel selected by FCTRLA.CHSEL bits (FCTRLA <11:10>) . INTFLAG.FAULTA (INTFLAG<12>) 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 FCTRLA.CHSEL bits (FCTRLA <11:10>) . INTFLAG.FAULTA (INTFLAG<12>) flag rises on each local maximum detection.
0x6	DERIV0	On rising edge of a valid recoverable Fault A, capture counter value on channel selected by FCTRLA.CHSEL bits (FCTRLA <11:10>) . INTFLAG.FAULTA (INTFLAG<12>) flag rises on each local maximum 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 CMPy value condition.
1	The recoverable Fault A input is disabled when channel output is at inactive level (CMPy = 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 Fault A mode.
Event system channel connected to Match or Capture Channel 0 (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	Fault A (MCE0) event input
0x2	INVERT	Enable Fault A (inverted) (MCE0) event input
0x3	ALTFAULT	Alternate fault (A or B) state at the end of the previous period.

48.7.6. Fault Control B

Name: FCTRLB
Offset: 0x10
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Table 48-14. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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 FAULT B (MCE1) event input line. An Input signal with pulse width shorter than (FILTERVAL) * GCLK_TCCx is filtered. An Input signal with pulse width larger than (FILTERVAL+1) * GCLK_TCCx is passed. The value must be set to zero when this 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(FCTRLB<6:5>)).
When enabled, the fault input source is internally disabled for BLANKVAL* prescaled GCLK_TCCx periods after the detection of the waveform edge.

Bit 15 – BLANKPRESC Recoverable Fault 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_TCCx.
1	Blank time is BLANKVAL* 64 * prescaled GCLK_TCCx.

Bits 14:12 – CAPTURE[2:0] Recoverable Fault B Capture Action
These bits select the capture and Fault B interrupt/event conditions.

Table 48-15. 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 FCTRLB.CHSEL bits (FCTRLB <11:10>). INTFLAG.FAULTB (INTFLAG<13>) flag rises on each new captured value.
0x2	CAPTMIN	On rising edge of a valid recoverable Fault B, capture counter value on channel selected byFCTRLB.CHSEL bits (FCTRLB <11:10>), if COUNT value is lower than the last stored capture value (CC). INTFLAG.FAULTB (INTFLAG<13>) flag rises on each local minimum detection.
0x3	CAPTMAX	On rising edge of a valid recoverable Fault B, capture counter value on channel selected by FCTRLB.CHSEL bits (FCTRLB <11:10>), if COUNT value is higher than the last stored capture value (CC). INTFLAG.FAULTB (INTFLAG<13>) 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 FCTRLB.CHSEL bits (FCTRLB <11:10>). INTFLAG.FAULTB (INTFLAG<13>) 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 FCTRLB.CHSEL bits (FCTRLB <11:10>). INTFLAG.FAULTB (INTFLAG<13>) 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 FCTRLB.CHSEL bits (FCTRLB <11:10>). INTFLAG.FAULTB (INTFLAG<13>) 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 CMPy value condition.
1	The recoverable Fault B input is disabled when output signal is at inactive level (CMPy = 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 Fault B mode.
Event system channel connected to Match or Capture Channel 1 (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	Enable Fault B (MCE1) event input
0x2	INVERT	Enable Fault B (inverted) (MCE1) event input
0x3	ALTFAULT	Alternate fault (A or B) state at the end of the previous period.

48.7.7. Waveform Extension Control

Name: WEXCTRL
Offset: 0x14
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Table 48-16. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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_TCCx clock cycles for the dead-time high side.

Bits 23:16 – DTLS[7:0] Dead-time Low Side Outputs Value
This register holds the number of GCLK_TCCx clock cycles for the dead-time low side.

Bits 8, 9, 10, 11 – DTIEN Dead-time Insertion Generator y Enable
Setting any of these bits enables the dead-time insertion generator for the corresponding output matrix. This will override the output matrix [y] and [y+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[y] and [y+WO_NUM/2], from matrix outputs[y] signal.

Bits 1:0 – OTMX[1:0] Output Matrix
These bits define the matrix routing of the TCC waveform generation outputs to the port pins, according to Waveform Extension.

48.7.8. Driver Control

Name: DRVCTRL
Offset: 0x18
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Table 48-17. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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 Fault 1 (TCE1) event input line. An Input signal with pulse width shorter than (FILTERVAL) * GCLK_TCCx is filtered. An Input signal with pulse width larger than (FILTERVAL+1) * GCLK_TCCx is passed. When value is 0, Fault1 acts asynchronously on PWM outputs.

Bits 27:24 – FILTERVAL0[3:0] Non-Recoverable Fault Input 0 Filter Value
These bits define the filter value applied on Fault 0 (TCE0) event input line. An Input signal with pulse width shorter than (FILTERVAL) * GCLK_TCCx is filtered. An Input signal with pulse width larger than (FILTERVAL+1) * GCLK_TCCx is passed. When value is 0, Fault 0 acts asynchronously on PWM outputs.

Bits 16, 17, 18, 19, 20, 21, 22, 23 – INVENy Waveform Output y Inversion
These bits are used to select inversion on the output of channel y.
Writing a '1' to INVENy inverts output from WO[y].
Writing a '0' to INVENy disables inversion of output from WO[y].

Bits 8, 9, 10, 11, 12, 13, 14, 15 – NRVy Non-Recoverable State y 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 – NREy Non-Recoverable State y Output Enable

These bits enable the override of individual outputs by NRVy value, under non-recoverable fault condition.

Value	Description
0	Non-recoverable fault tri-state the output.
1	Non-recoverable faults set the output to NRVy level.

48.7.9. Debug control

Name: DBGCTRL
Offset: 0x1E
Reset: 0x00
Property: PAC Write-Protection

Table 48-18. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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 written to '1', OCD break request from the OCD system will trigger 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 INTFLAG.DFS (INTFLAG<11>) 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.

48.7.10. Event Control

Name: EVCTRL
Offset: 0x20
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Table 48-19. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	MCEO7	MCEO6	MCEO5	MCEO4	MCEO3	MCEO2	MCEO1	MCEO0
Access	R/W	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
	MCEI7	MCEI6	MCEI5	MCEI4	MCEI3	MCEI2	MCEI1	MCEI0
Access	R/W	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
	TCEI1	TCEI0	TCINV1	TCINV0		CNTEO	TRGEO	OVFEO
Access	R/W	R/W	R/W	R/W		R/W	R/W	R/W
Reset	0	0	0	0		0	0	0

Bit	7	6	5	4	3	2	1	0
	CNTSEL[1:0]		EVACT1[2:0]			EVACT0[2:0]		
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 24, 25, 26, 27, 28, 29, 30, 31 – MCEOy Match or Capture Channel y Event Output Enable
These bits control if the Match/capture event on channel y is enabled and will be generated for every match or capture.

Value	Description
0	Match/capture y event is disabled and will not be generated.
1	Match/capture y event is enabled and will be generated for every compare/capture on channel y.

Bits 16, 17, 18, 19, 20, 21, 22, 23 – MCEIy Fault A/B or Capture Channel y Event Input Enable
These bits indicate if the Match/capture y incoming event is enabled
These bits are used to enable Fault A/B or capture input events to the CCy channel of TCCC.

Value	Description
0	Incoming events are disabled.
1	Incoming events are enabled.

Bits 14, 15 – TCEIn Timer/Counter Event Input n Enable
This bit is used to enable input event n to the TCC.

Value	Description
0	Incoming event n is disabled.
1	Incoming event n is enabled.

Bits 12, 13 – TCINVn Timer/Counter Event n Invert Enable
This bit inverts the event n input.

Value	Description
0	Input event source n is not inverted.
1	Input event source n 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 on EVCTRL.CNTSEL(EVCTRL<7:6>) settings.

Value	Description
0	Counter cycle output event is disabled and will not be generated.
1	Counter cycle output event is enabled and will be generated depending on EVCTRL.CNTSEL(EVCTRL<7:6>) 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 at begin of each counter cycle
0x1	END	An interrupt/event is generated at end of each counter cycle
0x2	-	Reserved
0x3	BOUNDARY	An interrupt/event is generated at begin of first counter cycle, and end of last counter cycle.

Bits 5:3 – EVACT1[2:0] Timer/Counter Event Input 1 Action
These bits define the action the TCC will perform on TCE1 event input.

Value	Name	Description
0x0	OFF	Event action disabled.
0x1	RETRIGGER	Start, restart or re-trigger TCC on event
0x2	DIR (asynch)	Direction control
0x3	STOP	Stop TCC on event
0x4	DEC	Decrement TCC on event
0x5	-	Reserved
0x6	PWP	Period captured into CC1 Pulse Width on CC0
0x7	FAULT	Non-recoverable Fault

Bits 2:0 – EVACT0[2:0] Timer/Counter Event Input 0 Action
These bits define the action the TCC will perform on TCE0 event input 0.

Value	Name	Description
0x0	OFF	Event action disabled.
0x1	RETRIGGER	Start, restart or re-trigger TCC on event
0x2	COUNTEV	Count on event.
0x3	START	Start TCC on event
0x4	INC	Increment TCC on EVENT
0x5	COUNT (async)	Count on active state of asynchronous event
0x6	STAMP	Capture overflow times (Max value)
0x7	FAULT	Non-recoverable Fault

48.7.11. 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.

Table 48-20. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	MC7	MC6	MC5	MC4	MC3	MC2	MC1	MC0
Access	R/W	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
	FAULT1	FAULT0	FAULTB	FAULTA	DFS	UFS		
Access	R/W	R/W	R/W	R/W	R/W	R/W		
Reset	0	0	0	0	0	0		

Bit	7	6	5	4	3	2	1	0
					ERR	CNT	TRG	OVF
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bits 16, 17, 18, 19, 20, 21, 22, 23 – MCy Match or Capture Channel y 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 y Interrupt Disable/Enable bit, which disables the Match or Capture Channel y interrupt.

Value	Description
0	The Match or Capture Channel y interrupt is disabled.
1	The Match or Capture Channel y 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.
Note: This bit is only available on variant L devices. Refer to the *Configuration Summary* for more information.

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.
1	The Overflow interrupt is enabled.

48.7.12. 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.

Table 48-21. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	MC7	MC6	MC5	MC4	MC3	MC2	MC1	MC0
Access	R/W	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
	FAULT1	FAULT0	FAULTB	FAULTA	DFS	UFS		
Access	R/W	R/W	R/W	R/W	R/W	R/W		
Reset	0	0	0	0	0	0		

Bit	7	6	5	4	3	2	1	0
					ERR	CNT	TRG	OVF
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bits 16, 17, 18, 19, 20, 21, 22, 23 – MCy Match or Capture Channel y 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 y Interrupt Disable/Enable bit, which enables the Match or Capture Channel y interrupt.

Value	Description
0	The Match or Capture Channel y interrupt is disabled.
1	The Match or Capture Channel y 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 enables the Non-Recoverable Fault 0 interrupt.

Value	Description
0	The Non-Recoverable Fault 0 interrupt is disabled.
1	The Non-Recoverable Fault 0 interrupt is enabled.

Bit 13 – FAULTB Recoverable Fault B Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Recoverable Fault B Interrupt Disable/Enable bit, which enables the Recoverable Fault B interrupt.

Value	Description
0	The Recoverable Fault B interrupt is disabled.
1	The Recoverable Fault B interrupt is enabled.

Bit 12 – FAULTA Recoverable Fault A Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Recoverable Fault A Interrupt Disable/Enable bit, which enables the Recoverable Fault A interrupt.

Value	Description
0	The Recoverable Fault A interrupt is disabled.
1	The Recoverable Fault A interrupt is enabled.

Bit 11 – DFS Non-Recoverable Debug Fault 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.
Note: This bit is only available on variant L devices. Refer to the *Configuration Summary* for more information.

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.

48.7.13. Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x2C
Reset: 0x00000000
Property: -

Note: Interrupt flags must be cleared and then read back to confirm they are cleared before exiting the ISR to avoid double interrupts.

Table 48-22. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	MC7	MC6	MC5	MC4	MC3	MC2	MC1	MC0
Access	R/W	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
	FAULT1	FAULT0	FAULTB	FAULTA	DFS	UFS		
Access	R/W	R/W	R/W	R/W	R/W	R/W		
Reset	0	0	0	0	0	0		

Bit	7	6	5	4	3	2	1	0
					ERR	CNT	TRG	OVF
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bits 16, 17, 18, 19, 20, 21, 22, 23 – MCy Match or Capture Channel y Interrupt Flag

This flag is set and resynchronized on the APB clock after a match with the compare condition or once CCy 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 y interrupt flag
In Capture operation, this flag is automatically cleared when CCy register is read.

Bit 15 – FAULT1 Non-Recoverable Fault 1 Interrupt Flag

This flag is set and resynchronized on the APB clock 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 and resynchronized on the APB clock after a Non-Recoverable Fault 0 occurs.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Non-Recoverable Fault 0 interrupt flag.

Value	Description
0	The Non-Recoverable Fault y interrupt is disabled.
1	The Non-Recoverable Fault y interrupt is enabled.

Bit 13 – FAULTB Recoverable Fault B Interrupt Flag
This flag is set and resynchronized on the APB clock 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 and resynchronized on the APB clock 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 11 – DFS Non-Recoverable Debug Fault State Interrupt Flag
This flag is set and resynchronized on the APB clock 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(CTRLBSET<1>)).
Writing a zero to this bit has no effect.
Writing a one to this bit clears the Non-Recoverable Update Fault interrupt flag.
Note: This bit is only available on variant L devices. Refer to the *Configuration Summary* for more information.

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 y 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 and resynchronized on the APB clock 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 and resynchronized on the APB clock after a counter retrigger occurs.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the re-trigger interrupt flag.

Bit 0 – OVF Overflow Interrupt Flag
This flag is set and resynchronized on the APB clock 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.

48.7.14. Status

Name: STATUS
Offset: 0x30
Reset: 0x00000001
Property: -

Table 48-23. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
			CMP5	CMP4	CMP3	CMP2	CMP1	CMP0
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
			CCBUFV5	CCBUFV4	CCBUFV3	CCBUFV2	CCBUFV1	CCBUFV0
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	FAULT1	FAULT0	FAULTB	FAULTA	FAULT1IN	FAULT0IN	FAULTBIN	FAULTAIN
Access	R/W	R/W	R/W	R/W	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	PERBUFV		PATTBUFV		DFS	UFS	IDX	STOP
Access	R/W		R/W		R/W	R/W	R	R
Reset	0		0		0	0	0	1

Bits 24, 25, 26, 27, 28, 29 – CMPy Channel y Compare Value
This bit reflects the channel y output compare value.

Value	Description
0	Channel compare output value is 0.
1	Channel compare output value is 1.

Bits 16, 17, 18, 19, 20, 21 – CCBUFVy Channel y Compare or Capture Buffer Valid
For a compare channel, this bit is set when a new value is written to the corresponding CCBUFy register. The bit is cleared either by writing a '1' to the corresponding location when CTRLBSET.LUPD (CTRLBSET<1>) 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 CCBUFy register. The bit is automatically cleared when the CCy register is read.

Bit 15 – FAULT1 Non-recoverable Fault 1 State
This bit is set by hardware as soon as non-recoverable Fault 1 condition occurs.
This bit is cleared by writing a one to this bit and when the corresponding STATUS.FAULT1IN(STATUS<11>) status bit is low.
Once this bit is cleared, the TCC will restart from the last COUNT value. To restart the TCC from BOTTOM, the TCC restart (RETRIGGER) command (CTRLBSET.CMD (CTRLBSET<7:5>)=1) must be

executed before clearing the FAULT1 STATE bit. For further details on TCC commands, refer to the available commands description in the CTRLBSET register.

Bit 14 – FAULT0 Non-recoverable Fault 0 State
This bit is set by hardware as soon as non-recoverable Fault 0 condition occurs.
This bit is cleared by writing a one to this bit and when the corresponding STATUS.FAULT0IN(STATUS<10>) status bit is low.
Once this bit is cleared, the TCC will restart from the last COUNT value. To restart the TCC from BOTTOM, the TCC restart (RETRIGGER) command (CTRLBSET.CMD (CTRLBSET<7:5>)=1) must be executed before clearing the FAULT0 STATE bit. For further details on TCC commands, refer to the available commands description in the CTRLBSET register.

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 the 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 (FCTRLB.HALT (FCTRLB<9:8>) =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 the 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 (FCTRLA.HALT (FCTRLB<9:8>) =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. The bit is cleared either by writing a '1' to the corresponding location when CTRLBSET.LUPD (CTRLBSET<1>) is set, or automatically on an UPDATE condition.

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 CTRLBSET.LUPD (CTRLBSET<1>) is set, or by writing a '1' to this bit.

Bit 3 – DFS Debug Fault State
This bit is set by hardware in Debug mode when DDBGCTRL.FDDBD(DDBGCTRL<2>) 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.NREy and DRVCTRL.NRVy 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 (CTRLBSET<1>)). The bit is cleared by writing a one to this bit.

When the bit is set, the waveforms state depend on DRVCTRL.NREy and DRVCTRL.NRVy registers.

Bit 1 – IDX Ramp Index

In RAMP2 and RAMP2A operation, the bit is cleared during the cycle A and set during the cycle B. In RAMP1 operation, the bit always reads zero. For details on Ramp operations, refer to 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(CTRLBSET <2>)=1).
This bit is clear on the next incoming counter increment or decrement.

Value	Description
0	Counter is running.
1	Counter is stopped.

48.7.15. Counter Value

Name: COUNT
Offset: 0x34
Reset: 0x00000000
Property: PAC Write-Protection, Write-Synchronized, Read-Synchronized

Note: Prior to any read access, this register must be synchronized by user by writing the according TCC Command value to the Control B Set register (CTRLBSET.CMD(CTRLBSET<7:5>)=READSYNC).

Table 48-24. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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 hold the value of the Counter register. When dithering is used, some LSBs of the counter cannot be used for counting and are read to 0. These bits are used for dithering. The number of LSBs used for dithering depends on the dithering resolution set by the CTRLA.RESOLUTION bits (CTRLA <6:5>).

Note: This bit field occupies the MSB of the register, [31:m]. m is dependent on the Resolution bit in the Control A register (CTRLA.RESOLUTION(CTRLA <6:5>)):

CTRLA.RESOLUTION	Bits [31:m]
0x0 - NONE	31:0 (default)
0x1 - DITH4	31:4
0x2 - DITH5	31:5
0x3 - DITH6	31:6

48.7.16. Pattern

Name: PATT
Offset: 0x38
Reset: 0x0000
Property: Write-Synchronized

Table 48-25. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	15	14	13	12	11	10	9	8
	PGV[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
	PGE[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:8 – PGV[7:0] Pattern Generation Output Value
This register holds the values of pattern for each waveform output.

Bits 7:0 – PGE[7:0] 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 SWAP output with the corresponding value bit from PGV[7:0] bit array.

48.7.17. Waveform

Name: WAVE
Offset: 0x3C
Reset: 0x00000000
Property: Write-Synchronized

Table 48-26. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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
	POL7	POL6	POL5	POL4	POL3	POL2	POL1	POL0
Access	R/W	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
					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 – SWAPy Swap DTI Output Pair y
Setting these bits enables output swap of DTI outputs [y] and [y+WO_NUM/2]. Note the DTIyEN settings will not affect the swap operation.

Bits 16, 17, 18, 19, 20, 21, 22, 23 – POLy Channel Polarity y
Setting these bits enables the output polarity in single-slope and dual-slope PWM operations.

Value	Name	Description
0	(single-slope PWM waveform generation)	Compare output is initialized to ~DIR and set to DIR when TCC counter matches CCy value
1	(single-slope PWM waveform generation)	Compare output is initialized to DIR and set to ~DIR when TCC counter matches CCy value.
0	(dual-slope PWM waveform generation)	Compare output is set to ~DIR when TCC counter matches CCy value
1	(dual-slope PWM waveform generation)	Compare output is set to DIR when TCC counter matches CCy value.

Bits 8, 9, 10, 11 – CICCENy Circular CC Enable y
Setting this bits enables the compare circular buffer option on the first four Compare/Capture channels. When the bit is set, CCy register value is copied-back into the CCy register on UPDATE condition.

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 PERBUF register on UPDATE condition.

Bits 6:4 – RAMP[2:0] Ramp Operation

These bits select Ramp operation (RAMP). These bits are not synchronized.

Value	Name	Description
0x0	RAMP1	RAMP1 operation
0x1	RAMP2A	Alternative RAMP2 operation
0x2	RAMP2	RAMP2 operation
0x3	RAMP2C	Critical RAMP2 operation
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. These bits are not synchronized.

Value	Name	Description						
		Operation	Top	Update	Waveform Output On Match	Waveform Output On Update	OVF Interrupt Flag/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	-	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	–

48.7.18. Period Value

Name: PER
Offset: 0x40
Reset: 0xFFFFFFFF
Property: Write-Synchronized

Table 48-27. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	PER[25:18]							
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[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 31:6 – PER[25:0] Period Value

These bits hold the value of the Period Buffer register. The number of bits in this field corresponds to the size of the counter.

Note: When the TCC instance is a 16-bit timer/counter, the excess bits are read zero.

Note: This bit field occupies the MSB of the register, [31:m]. m is dependent on the Resolution bit in the Control A register (CTRLA.RESOLUTION(CTRLA <6:5>)):

CTRLA.RESOLUTION	Bits [31:m]
0x0 - NONE	31:0
0x1 - DITH4	31:4
0x2 - DITH5	31:5
0x3 - DITH6	31:6 (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 <6:5>)):

CTRLA.RESOLUTION	Bits [n:0]
0x0 - NONE	-
0x1 - DITH4	3:0
0x2 - DITH5	4:0
0x3 - DITH6	5:0 (depicted)

48.7.19. Compare/Capture Channel y

Name: CCy
Offset: 0x44 + y*0x04 [y=0..5]
Reset: 0x00000000
Property: Write-Synchronized, Read-Synchronized

The CCy register represents the 32-bit value. 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.

The CCy register is updated with the buffer value from their corresponding CCBUFy register when an UPDATE condition occurs.

In addition, in match frequency operation, the CC0 register controls the counter period.

Table 48-28. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	CC[25:18]							
Access	R/W	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[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 31:6 – CC[25:0] Channel y Compare/Capture Value
These bits hold the value of the Channel y compare/capture register.

Notes:

1. When the TCC instance is a 16-bit timer/counter, the excess bits are read zero.
2. This bit field occupies the MSB of the register, [31:m]. m is dependent on the Resolution bit in the Control A register (CTRLA.RESOLUTION(CTRLA <6:5>)):

CTRLA.RESOLUTION	Bits [31:m]
0x0 - NONE	31:0
0x1 - DITH4	31:4
0x2 - DITH5	31:5
0x3 - DITH6	31:6 (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 <6:5>)):

CTRLA.RESOLUTION	Bits [n:0]
0x0 - NONE	-
0x1 - DITH4	3:0
0x2 - DITH5	4:0
0x3 - DITH6	5:0 (depicted)

48.7.20. Pattern Buffer

Name: PATTBUF
Offset: 0x64
Reset: 0x0000
Property: Write-Synchronized, Read-Synchronized

Note: This register must be written with 16 bit accesses only (no 8 bit writes).

Table 48-29. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	15	14	13	12	11	10	9	8
	PGVB[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
	PGEB[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:8 – PGVB[7:0] 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.

Bits 7:0 – PGEB[7:0] 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.

48.7.21. Period Buffer Value

Note: This register must be written with 32-bit accesses only (no 8-bit or 16 bit writes).

Name: PERBUF
Offset: 0x6C
Reset: 0xFFFFFFFF
Property: Write-Synchronized, Read-Synchronized

Table 48-30. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	PERBUF[25:18]							
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[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 31:6 – PERBUF[25:0] Period Buffer Value

These bits hold the value of the Period Buffer register. The value is copied to the PER register on UPDATE condition.

Note: When the TCC instance is a 16-bit timer/counter, the excess bits are read zero.

Note: This bit field occupies the MSB of the register, [31:m]. m is dependent on the Resolution bit in the Control A register (CTRLA.RESOLUTION(CTRLA <6:5>)):

CTRLA.RESOLUTION	Bits [31:m]
0x0 - NONE	31:0
0x1 - DITH4	31:4
0x2 - DITH5	31:5
0x3 - DITH6	31:6 (depicted)

Bits 5:0 – DITHERBUF[5:0] Dithering Buffer Cycle Number

These bits represent the buffer for the PER.DITHER bits. When the double buffering is enabled, the value of this bit field is copied to the PER.DITHER bits on an UPDATE condition.

Note: This bit field consists of the n LSB of the register. n is dependent on the value of the Resolution bits in the Control A register (CTRLA.RESOLUTION(CTRLA <6:5>)):

CTRLA.RESOLUTION(CTRLA <6:5>)	Bits [n:0]
0x0 - NONE	-
0x1 - DITH4	3:0
0x2 - DITH5	4:0
0x3 - DITH6	5:0 (depicted)

48.7.22. Channel y Compare/Capture Buffer Value

Name: CCBUFy
Offset: 0x70 + y*0x04 [y=0..5]
Reset: 0x00000000
Property: Write-Synchronized, Read-Synchronized

Note: This register must be written with 32-bit accesses only (no 8-bit or 16 bit writes).

CCBUFy is copied into CCy at TCC update time.

Table 48-31. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	CCBUF[25:18]							
Access	R/W	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[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 31:6 – CCBUF[25:0] Channel y Compare/Capture Buffer Value
These bits hold the value of the Channel y Compare/Capture Buffer Value register. The register serves as the buffer for the associated compare or capture registers (CCy). Accessing this register using the CPU or DMA will affect the corresponding STATUS.CCBUFVy status bit.
Note: When the TCC instance is a 16-bit timer/counter, the excess bits are read zero.

Note:

1. This bit field occupies the MSB of the register, [31:m]. m is dependent on the Resolution bit in the Control A register (CTRLA.RESOLUTION(CTRLA <6:5>)):

CTRLA.RESOLUTION(CTRLA <6:5>)	Bits [31:m]
0x0 - NONE	31:0
0x1 - DITH4	31:4
0x2 - DITH5	31:5
0x3 - DITH6	31:6 (depicted)

Bits 5:0 – DITHERBUF[5:0] Dithering Buffer Cycle Number

These bits represent the CCy.DITHER bits buffer. When the double buffering is enable, CCBUFy.DITHERBUF bits value is copied to the CCy.DITHER bits on an UPDATE condition.

Note: This bit field consists of the n LSB of the register. n is dependent on the value of the Resolution bits in the Control A register (CTRLA.RESOLUTION(CTRLA <6:5>)):

CTRLA.RESOLUTION(CTRLA <6:5>)	Bits [n:0]
0x0 - NONE	-
0x1 - DITH4	3:0
0x2 - DITH5	4:0
0x3 - DITH6	5:0 (depicted)

49. TrustRAM (TRAM)

49.1. Overview

The TrustRAM (TRAM) module is the controller interface for the 1 KB security RAM, which is intended for volatile secret data. The TRAM module is capable of performing address map scrambling for both write and read access to the security RAM. It can also perform data scrambling on write access, and data descrambling on read access. To improve side-channel attack resistance, the TRAM module can perform silent access of the data stream from the security RAM.

The TRAM module 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 security RAM is inverted to prevent physical 'burn-in' signatures. When a tamper event occurs, the TRAM module executes a full erasure of scramble key 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.

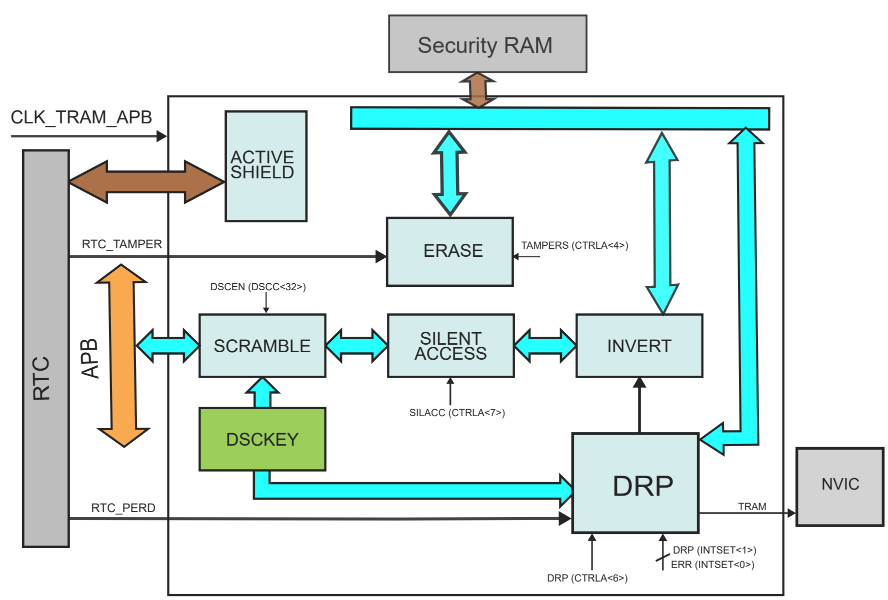
49.2. Features

The following are key features of the TrustRAM module:

- Address scrambling to the security RAM
- Data scrambling to the security RAM and data descrambling from the security RAM
- Silent access of data for improved side-channel resistance
- Data remanence prevention
- Active shielding to prevent physical tamper on security RAM
- Full erasure of scramble key and security RAM data on tamper detection

49.3. Block Diagram

Figure 49-1. Block Diagram



49.4. Peripheral Dependencies

Peripheral Name	Base Address	NVIC IRQ Index:Source	MCLK Clock Enable Mask Bit	PAC Peripheral ID (PAC.WRCTRL.PERIDx)	Power Domain
TRAM	0x4401_E000	31 : ERR, DRP	MCLK.CLKMSK1[15]	15	AVDD

49.5. Functional Description

49.5.1. Principle of Operation

System bus transactions from the CPU to the security RAM undergo a scrambling routine. The TRAM module modifies both address and data bus information through an algorithm determined by a scrambling key (DSCC.DSCKEY (DSCC<29:0>)). This is performed on both write and read transactions.
When the TRAM module receives a Tamper Event (RTC_TAMPER) from the RTC module, it erases the full security RAM and the scrambling key. When it receives a RTC Interval Periodic Event (RTC_PERD) from the RTC module, the TRAM module runs a data remanence routine on the security RAM.
The TRAM module can be configured to generate interrupts. The following sections describe each operation in detail.

49.5.2. Basic Operation

49.5.2.1. Clocks

The TRAM bus clock (CLK_TRAM_APB) can be enabled and disabled in the Main Clock module, and the default state of CLK_TRAM_APB can be found in the *Peripheral Clock Masking* section.

49.5.2.2. Register Access Protection

All registers with write-access are optionally write-protected in the Peripheral Access Controller (PAC), except for the following registers:

- The Interrupt Flag (INTFLAG) register
- All RAM addresses

Write-protection is denoted by the Write-Protected property in the register description. Write-protection does not apply to accesses through an external debugger. Refer to the *Peripheral Access Controller* chapter for additional information.

49.5.2.3. Enable Protection

The following registers and bits are enable-protected, meaning that they can only be written when the TRAM module is disabled (CTRLA.ENABLE bit (CTRLA <1>) = ‘0’):

- The Tamper Erase bit in the Control A register (CTRLA.TAMPERS bit (CTRLA <4>))
- The Data Remanence Protection bit in the Control A register (CTRLA.DRP bit (CTRLA <6>))
- The Silent Access bit in the Control A register (CTRLA.SILACC bit (CTRLA <7>))
- The Data Scramble Control register (DSCC)

When the CTRLA.ENABLE bit (CTRLA <1>) is '0', the enable-protected bits in the CTRLA register can be written simultaneously the CTRLA.ENABLE bit (CTRLA <1>) is written to '1'. However, when the CTRLA.ENABLE bit (CTRLA <1>) is '1' these bits can not be written simultaneously the CTRLA.ENABLE bit (CTRLA <1>) is written. Enable-protection is denoted by the Enable-Protected property in the register description.

49.5.2.4. Enabling, Disabling and Resetting

The TRAM is enabled by writing a ‘1’ to the Enable bit in the Control A register CTRLA.ENABLE bit (CTRLA<1>). The TRAM is disabled by writing a ‘0’ to CTRLA.ENABLE bit (CTRLA<1>).

The TRAM is reset by writing a ‘1’ to the Software Reset bit in the Control A register (CTRLA.SWRST bit (CTRLA <0>)). 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’.

49.5.2.5. Synchronization

In the TRAM module some bits must be synchronized when accessed. A register or register bits may require:

- Synchronization when written
- No synchronization

When executing an operation that requires synchronization, the corresponding status bit in the Synchronization Busy (SYNCBUSY) register sets immediately, and is cleared when synchronization is complete.

If an operation that requires the execution of synchronization while the corresponding bit in the SYNCBUSY register is set to ‘1’, the operation is discarded and an error is generated. In the TRAM module, the following bits need synchronization when written:

- The Software Reset bit in the Control A register (CTRLA.SWRST bit (CTRLA <0>))
- The Enable bit in the Control A register (CTRLA.ENABLE bit (CTRLA <1>))

49.5.2.6. Initialization

Follow these steps to operate the TRAM module in basic mode:

1. Configure the clock source for CLK_TRAM_APB in the Main Clock Controller (MCLK) and enable the clock by writing a ‘1’ to the TRAM bit in the APB Mask register of the MCLK.
2. Clear all registers by writing ‘1’ to the CTRLA.SWRST bit (CTRLA<0>) and wait for the SYNCBUSY.SWRST bit (SYNCBUSY<0>) to set.
3. Program data scrambling key in the DSCC.DSCKEY bits (DSCC <29:0) and enable scrambling by programming the DSCEN bit (DSCC <31>).
4. Enable the TRAM module by setting the CTRLA.ENABLE bit (CTRLA<1>) and wait for the SYNCBUSY.CTRLA bit (SYNCBUSY<1>) to set.

49.5.2.7. Scrambling

The Data Scramble Control (DSCC) must be configured before the CTRLA.ENABLE bit (CTRLA <1>) is set. These settings cannot be changed while the module is enabled.

The scrambling logic is enabled by writing ‘1’ to the ENABLE bit in the Data Scramble Control register (DSCC.DSCEN bit (DSCC <31>)). Scrambling is disabled by writing a ‘0’ to the DSCC.DSCEN bit (DSCC <31>).

49.5.2.8. Silent Access

The Silent Access bit (CTRLA.SILACC (CTRLA <7>)) must be configured before the CTRLA.ENABLE bit (CTRLA <1>) is set. This setting cannot be changed while the module is enabled. When this mode is enabled, only half of the security RAM (4 KB) is available for data storage because the other half is reserved to store the 1's complement (bitwise invert) values.

The TRAM module executes the following protocols:

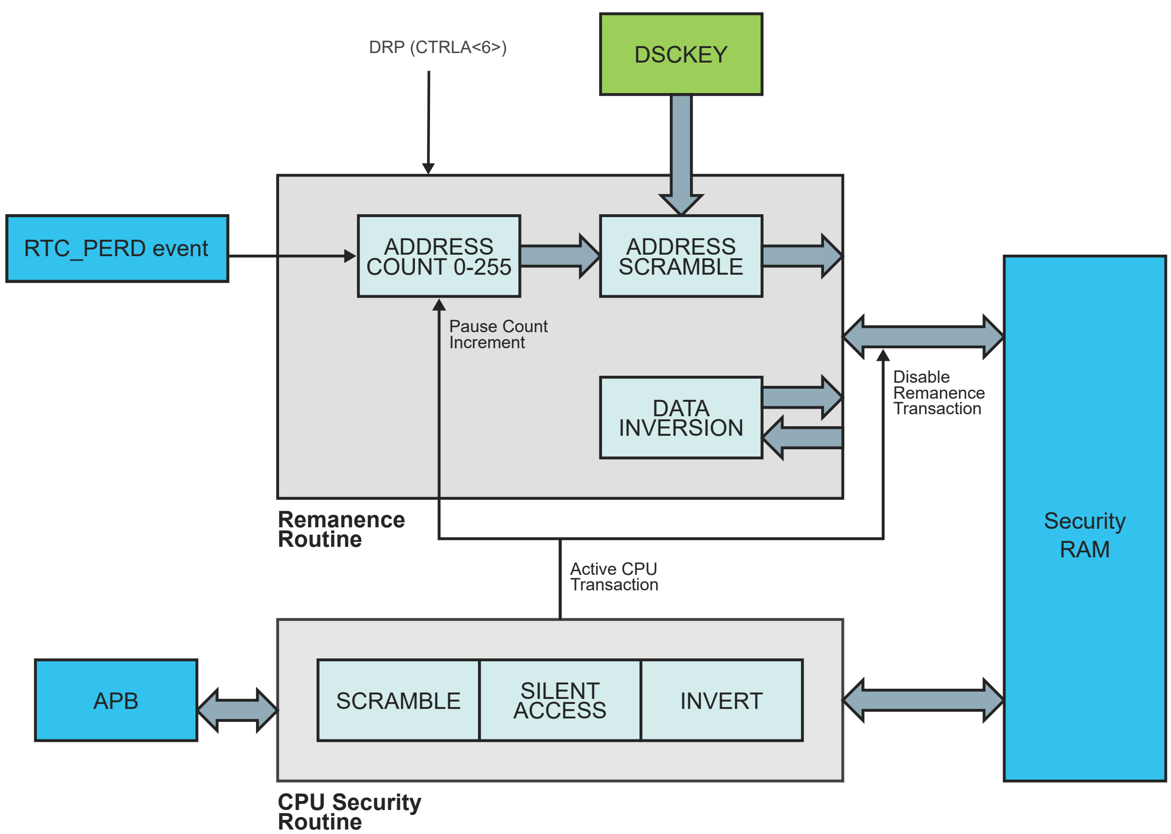
- When the CPU writes to the security RAM, the data and its bitwise invert are stored into the security RAM.
- When the CPU reads from the security RAM, both the data and its bitwise invert are retrieved from the security RAM. If the TRAM module cannot verify that both values complement each other, a bus error is returned.

49.5.2.9. Data Remanence Prevention

Data remanence prevention can be enabled by setting the CTRLA.DRP (CTRLA <6>) bit. The Data remanence prevention bit (CTRLA.DRP (CTRLA <6>)) must be configured before the CTRLA.ENABLE bit (CTRLA <1> is set. This setting cannot be changed while the module is enabled. When this feature is enabled, the RTC Periodic Interval Event (RTC_PERD) will trigger the automated data remanence routine. An internal counter will count from 0 to 2047 and serves as the address access bus to the security RAM. For every address iteration, the TRAM module 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 (RAMINV bit (STATUS <0>)) as showin in the following figure.

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 49-2. Remanence Routine



49.5.2.10. Tamper Full Erase

Tamper full erase bit (CTRLA.TAMPERS (CTRLA <4>)) must be configured before CTRLA.ENABLE bit (CTRLA <1>) 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 module software reset and the reset of the Data Scramble Key (DSCC.DSCKEY bits (DSCC <29:0)) register. All the TRAM registers are reverted to the default reset value. Data inside the security RAM is written to ‘0’ for all address locations.

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.

49.5.3. Interrupts

The TRAM module 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 ‘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 TRAM is reset. See INTFLAG for details on how to clear interrupt flags. The interrupt request lines are connected to the interrupt controller. Both interrupt requests from the TRAM module are read together on the system level to generate one combined interrupt request to the NVIC. This combined interrupt is called the TRAM interrupt. Using the TRAM module interrupts requires the interrupt controller to be configured first. Refer to *Nested Vector Interrupt Controller* for details. The user must read the INTFLAG register to determine which interrupt condition is present.

The following additional steps are needed for the TRAM module to operate in the interrupt mode before enabling the TRAM module:

- Configure the NVIC by setting group priority, sub priority and by enabling TRAM IRQ (Vector no 38)
- Enable TRAM Data Read Error (ERR) interrupt by setting INTENSET.ERR bit (INTENSET<0>)
- Enable TRAM Data Remanence Prevention (DRP) interrupt by setting INTENSET.DRP bit (INTENSET<1>)
- In the TRAM Interrupt Service Routine write '1' in INTFLAG.DRP bit (INTFLAG <1>) to clear the DRP interrupt and write '1' in INTFLAG.ERR bit (INTFLAG <0>) to clear the ERR interrupt

49.5.4. Events

The RTC Period Interval event (RTC_PERD) for Data Remanence Prevention and the RTC Tamper (RTC_TAMPER) event are directly connected from the RTC to the TRAM, without going through the Event System. No output events are connected from the TRAM module to Event Systems.

49.5.5. Sleep Mode Operation

The TRAM continues to operate during sleep. When it receives events from the RTC, it will request its own clock in order to perform the requested operation (remanence and tamper erase).

The TRAM will continue to operate in any sleep mode, as long as its source clock is running. The TRAM interrupt can be used to wake up the device from sleep mode.

49.6. Register Summary

For descriptions and definitions of both Register and bitfield properties, refer to Register Properties.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRLA	31:24								
		23:16								
		15:8								
		7:0	SILACC	DRP		TAMPERS			ENABLE	SWRST
0x04	INTENCLR	31:24								
		23:16								
		15:8								
		7:0							DRP	ERR
0x08	INTENSET	31:24								
		23:16								
		15:8								
		7:0							DRP	ERR
0x0C	INTFLAG	31:24								
		23:16								
		15:8								
		7:0							DRP	ERR
0x10	STATUS	31:24								
		23:16								
		15:8								
		7:0							DRP	RAMINV
0x14	SYNCBUSY	31:24								
		23:16								
		15:8								
		7:0							ENABLE	SWRST
0x18	DSCC	31:24	DSCEN		DSCKEY[29:24]					
		23:16	DSCKEY[23:16]							
		15:8	DSCKEY[15:8]							
		7:0	DSCKEY[7:0]							
0x1C ... 0x0FFF	Reserved									
0x1000	RAM0	31:24	DATA[31:24]							
		23:16	DATA[23:16]							
		15:8	DATA[15:8]							
		7:0	DATA[7:0]							
...										
0x107C	RAM31	31:24	DATA[31:24]							
		23:16	DATA[23:16]							
		15:8	DATA[15:8]							
		7:0	DATA[7:0]							

49.6.1. Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00000000
Property: PAC Write Protection

Notes:
1. Only ENABLE and SWRST bits are Write Synchronized.
2. Read/Write access to this register is limited to 32-bit width. Byte level access is not allowed.
3. Reserved bits must always be written as '0'.
4. TRAM module registers are not reset during a soft system reset.

Table 49-1. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

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.

Value	Description
0	Silent access is disabled.
1	Silent access is enabled.

Bit 6 – DRP Data Remanence Prevention
Enables periodic Data remanence prevention (DRP) in the TRAM module.

Value	Description
0	Data remanence prevention is disabled.
1	Data remanence prevention is enabled.

Bit 4 – TAMPERS Tamper Erase
This bit enables auto-erase of the security RAM and DSCKEY bits (DSCC <29:0) on tamper event.

Value	Description
0	Tamper erase is disabled.
1	Tamper erase is enabled.

Bit 1 – ENABLE Enable
This bit is not Enable-Protected.

Value	Description
0	The TRAM module is disabled.
1	The TRAM module 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 TRAM module to their initial state, and the TRAM module will be disabled. This bit can also be set via hardware when a tamper occurs while CTRLA.TAMPERS bit (CTRLA <4>) is set.
Writing a one to CTRLA.SWRST bit (CTRLA <0>) will always take precedence, meaning that all other writes in the same write operation will be discarded.
Due to synchronization there is a delay from writing CTRLA.SWRST bit (CTRLA <0>) until the reset is complete. CTRLA.SWRST bit (CTRLA <0>) and SYNCBUSY.SWRST bit (SYNCBUSY <0>) will both be cleared when the reset is complete.
Reading this bit provides the following information.
Notes:

1. When the CTRLA.SWRST is written, the user should poll the SYNCB.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	The reset operation is not ongoing.
1	The reset operation is ongoing.

49.6.2. Interrupt Enable Clear

Name: INTENCLR
Offset: 0x04
Reset: 0x00000000
Property: PAC Write-Protection

Notes:
1. Access to this register is limited to 32-bit width. Byte level access is not allowed.
2. Reserved bits must always be written as ‘0’.

Table 49-2. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

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 ‘0’ to this bit has no effect.
Writing a ‘1’ to this bit will clear the Data Remanence Prevention Complete Interrupt Enable bit, which disables the data remanence prevention complete interrupt. Reading this bit provides the following information.

Value	Description
0	Data remanence prevention complete interrupt is disabled.
1	Data remanence prevention complete interrupt is enabled.

Bit 0 – ERR TRAM Read Error Interrupt Enable
Writing a ‘0’ to this bit has no effect.
Writing a ‘1’ to this bit will clear the TRAM Read Error Interrupt Enable bit, which disables the TRAM read error interrupt. Reading this bit provides the following information.

Value	Description
0	TRAM read error interrupt is disabled.

Value	Description
1	TRAM read error interrupt is enabled.

49.6.3. Interrupt Enable Set

Name: INTENSET
Offset: 0x08
Reset: 0x00000000
Property: PAC Write-Protection

Notes:
1. Access to this register is limited to 32-bit width. Byte level access is not allowed.
2. Reserved bits must always be written as ‘0’.

Table 49-3. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

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 ‘0’ to this bit has no effect.
Writing a ‘1’ to this bit will set the Data Remanence Prevention Complete Interrupt Enable bit, which enables the data remanence prevention complete interrupt. Reading this bit provides the following information.

Value	Description
0	Data remanence prevention complete interrupt is disabled.
1	Data remanence prevention complete interrupt is enabled.

Bit 0 – ERR TRAM Read Error Interrupt Enable
Writing a ‘0’ to this bit has no effect.
Writing a ‘1’ to this bit will set the TRAM Read Error Interrupt Enable bit, which enables the TRAM read error interrupt. Reading this bit provides the following information.

Value	Description
0	TRAM read error interrupt is disabled.

Value	Description
1	TRAM read error interrupt is enabled.

49.6.4. Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x0C
Reset: 0x00000000
Property: -

Notes:
1. Access to this register is limited to 32-bit width. Byte level access is not allowed.
2. Reserved bits must always be written as ‘0’.
3. Interrupt flags must be cleared and then read back to confirm they are cleared before exiting the ISR to avoid double interrupts.

Table 49-4. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

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 bit (INTENSET <1>) is set to '1'.
Writing a ‘0’ to this bit has no effect.
Writing a ‘1’ to this bit clears the data remanence prevention complete interrupt flag.

Bit 0 – ERR TRAM Read Error Interrupt
This flag is set when an error is detected in the TRAM readout, and an interrupt request will be generated if INTENSET.ERR bit (INTENSET <0>) is set to '1'.
Writing a ‘0’ to this bit has no effect.
Writing a ‘1’ to this bit clears the TRAM read error interrupt flag.

49.6.5. Status

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

Note: Access to this register is limited to 32-bit width. Byte level access is not allowed.

Table 49-5. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

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 provides the status of the data remanence prevention routine.

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 provides the status of the TRAM bit values inversion function.

Value	Description
0	The TRAM physical bit information is normal.
1	The TRAM physical bit information is inverted.

49.6.6. Synchronization Busy

Name: SYNCBUSY
Offset: 0x14
Reset: 0x00000000
Property: -

Note: Access to this register is limited to 32-bit width. Byte level access is not allowed.

Table 49-6. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
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 (CTRLA <1>) bit is complete.
1	Write synchronization for CTRLA.ENABLE bit (CTRLA <1>) bit is ongoing.

Bit 0 – SWRST Software Reset Synchronization Busy Status

This bit will set in two ways:

- Writing ‘1’ to CTRLA.SWRST bit (CTRLA <0>)
- A tamper event can occur when CTRLA.TAMPERS (CTRLA <4>) = ‘1’ (i.e., Tamper erase is enabled)

Value	Description
0	Write synchronization for CTRLA.SWRST bit (CTRLA <0>) bit is complete.
1	Write synchronization for CTRLA.SWRST bit (CTRLA <0>) bit is ongoing.

49.6.7. Data Scramble Control

Name: DSCC
Offset: 0x18
Reset: 0x00000000
Property: PAC Write-Protected, Enable-Protected

Notes:
1. Access to this register is limited to 32-bit width. Byte level access is not allowed.
2. Reserved bits must always be written as '0'.

Table 49-7. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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
Writing '1' to this bit enables the TRAM scrambling function. Reading this bit provides the following information:

Value	Description
0	TRAM scrambling function is disabled
1	TRAM scrambling function is enabled

Bits 29:0 – DSCKEY[29:0] Data Scramble Key
The key value used for address and data scrambling and descrambling. Any value written to this field is XOR'ed with the previous data. Writing '1' to CTRLA.SWRST bit (CTRLA <0>) will reset this field to 0. These bits will always return '0' when read.

49.6.8. RAM

Name: RAM
Offset: 0x1000 + n*0x04 [n=0..31]
Reset: 0x00000000
Property: -

Access to the Security RAM is only permitted when CTRLA.ENABLE bit (CTRLA <1>).

Note: Access to this register is limited to 32-bit width. Byte level access is not allowed.

Table 49-8. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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] Security RAM Data

50. Peripheral Touch Controller (PTC)

50.1. 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/0 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 (IT0) 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-line and Y-line are depending on both package type and device configuration. Refer to the *Signal Description* table for additional information.

Important: If AVDD < 2.5v, users must enable analog charge pumps in the SUPC.VREGCTRL.CPEN[1:0].

50.2. Features

The following are key features of the Peripheral Touch Controller module:

- Low-power, high-sensitivity, environmentally robust capacitive touch buttons, sliders, and wheels
- 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 VDD range
  - Auto calibration and recalibration of sensors
- Single-shot charge measurement
- Hardware noise filtering and noise signal desynchronization for high conducted immunity
- Polarity control, allowing Parallel Acquisition (through the QTouch Library) individually controls the polarity of each line
- Driven Shield+ 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
- 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

50.3. Block Diagram

Figure 50-1. PTC Block Diagram Mutual Capacitance

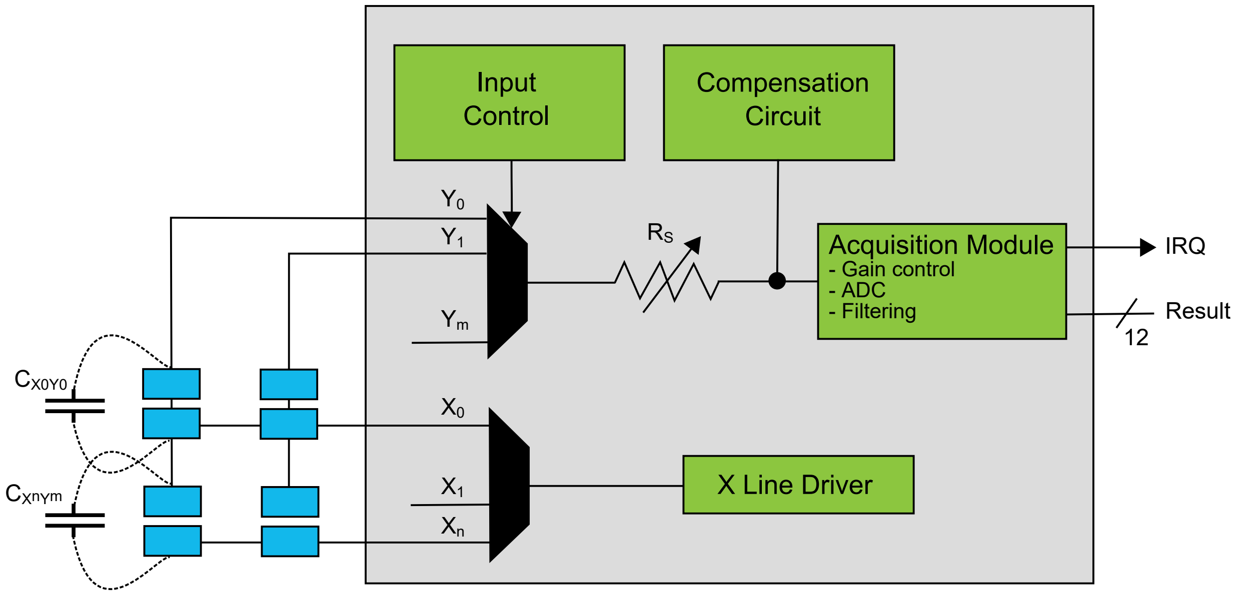
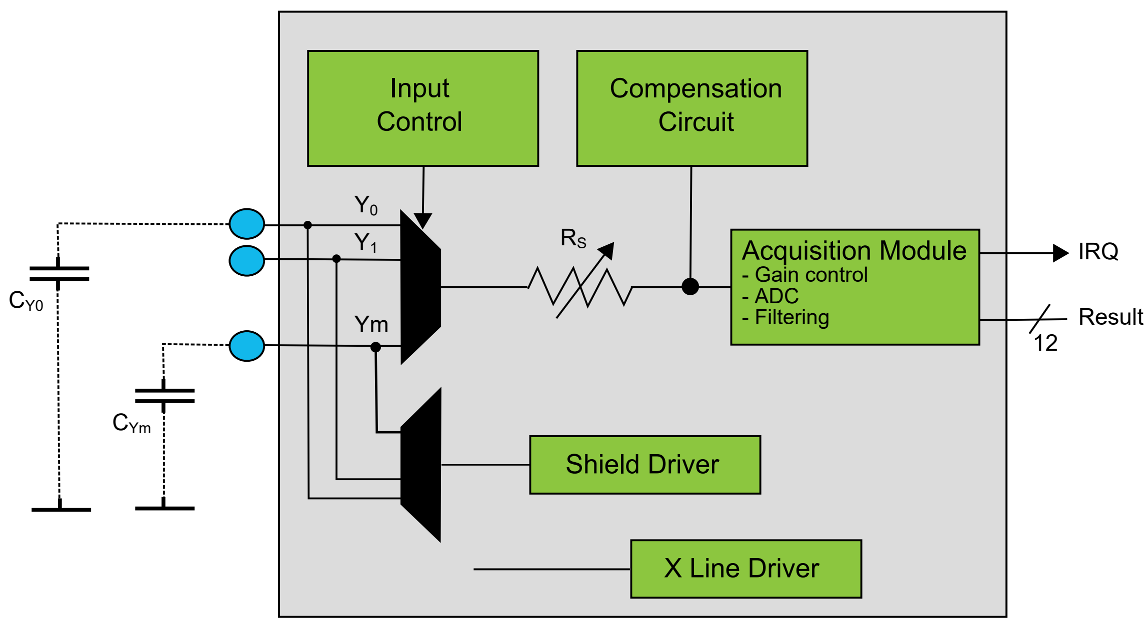


Figure 50-2. PTC Block Diagram Self Capacitance



50.4. Signal Description

Table 50-1. Signal Description for PTC

Name	Type	Description
Y[m:0]	Analog	Y-line (Input/Output)

Table 50-1. Signal Description for PTC (continued)

Name	Type	Description
X[n:0]	Digital	X-line (Output)

Note: The number of X- and Y-lines are device dependent. Refer to the Configuration Summary for additional information.

Refer to the Pinout section for details on the pin mapping for this peripheral. One signal can be mapped on several pins.

50.5. Peripheral Dependencies

Peripheral Name	Base Address	NVIC IRQ Index: Source	MCLK AHBx/APBx Clock Enable Mask Bit	GCLK Peripheral Channel Clock Name:Register	PAC Peripheral Identifier (PAC.WRCTRL.PERIDx)	EVSYS Users (EVSYS.USERm)	EVSYS Generator (EVSYS.CHANNELn.EVGENx)	DMA Trigger Index:Source (DMAC.CHCTRLBk.TRIGx)	Power Domain
PTC	0x4501_4000	139 : EOC	MCLK.CLKMSK3[10]	GCLK_PTC : GCLK.PCHCTRL[35]	46	91 : DSEQR	111 : EOC	62 : SEQ 63 : WCOMP	VDDREG

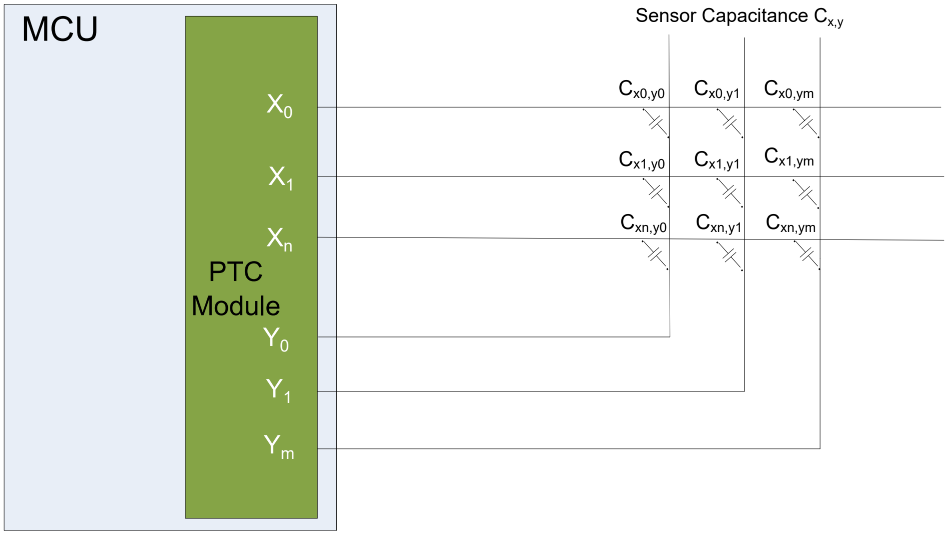
50.5.1. I/O Lines

The I/0 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. However, to improve the EMC performance, a series resistor of 1kΩ or more can be used on X-lines and Y-lines.

50.5.1.1. Mutual-Capacitance Sensor Arrangement

A mutual-capacitance sensor is formed between two I/0 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.

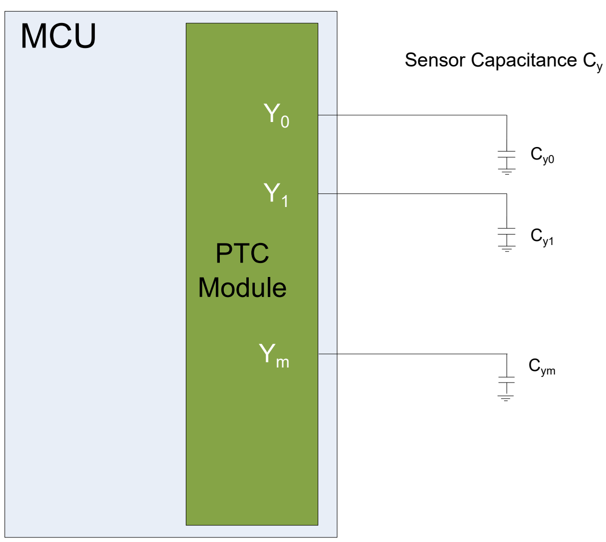
Figure 50-3. Mutual Capacitance Sensor Arrangement



50.5.1.2. Self-Capacitance Sensor Arrangement

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 50-4. Self-Capacitance Sensor Arrangement



For more information about designing the touch sensor, refer to Buttons, Sliders and Wheels Touch Sensor Design Guide.

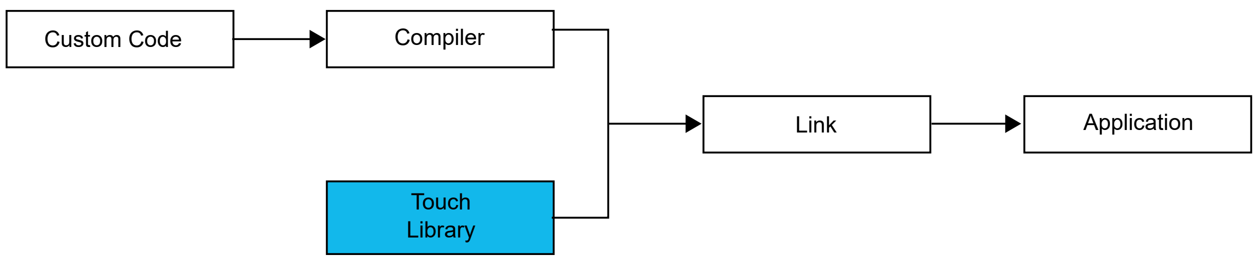
50.5.2. Clocks

The PTC is clocked by the GCLK_PTC clock. The PTC operates from an asynchronous clock source and the operation is independent of the main system clock and its derivative clocks, such as the peripheral bus clock (CLK_PTC_APB enabled by default in the MCLK.CLKMSK3.MSK10 register). A number of clock sources can be selected as the source for the asynchronous GCLK_PTC. The clock source is selected by configuring the Generic Clock Selection ID in the Generic Clock Control register. For more information about selecting the clock sources, refer to GCLK - Generic Clock Controller.

50.6. Functional Description

In order to access the PTC, the user must use the Touch Configurator 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.

Figure 50-5. Touch Library Usage



For more information about Touch Library, refer to the Touch Library Peripheral Touch Controller User Guide.

50.6.1. Principle of Operation

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.

50.6.2. Basic Operation

50.6.2.1. Mutual Capacitance Mode

The I/O Port pins that are chosen for touch sensing channels must be set to Analog input/output mode using the Port control signals. The chosen X- and Y-channel set must be enabled for PTC use by writing the desired configuration into I/O Port control registers.

50.6.2.2. Self Capacitance Mode

The I/O Port pins that are chosen for touch sensing channels must be set to Analog input/output mode using the Port control signals. The chosen Y-channel set must be enabled for PTC use by writing the desired configuration into I/O Port control registers.

50.6.3. Driven Shield

Driven shield is the capability to drive an output line in synchronization with TOUCH input. This will provide better noise immunity from noise sources beneath the sensor, and improved sensitivity of the touch signal. Driven shield will only be used in self-cap mode.

50.6.4. Interrupts

The PTC has the following interrupt sources:

- End of Conversion:EOC
- Window Monitor: WCOMP
- Analog Core Ready:ACRRDY

These interrupts are asynchronous wake-up sources. See Sleep Mode Controller for details.

Note: Interrupts must be globally enabled for interrupt requests to be generated. Refer to Nested Vector Interrupt Controller for details.

50.6.5. Events

The PTC can generate the following output events:

- End Of Conversion (EOC): Generated when the conversion is complete and the result is available. Refer to EVCTRL for details.
- Window Monitor (WCOMP): Generated when the window monitor condition match.

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 PTC can take the following actions on an input event:

- Start a conversion
- Restart DMA sequencing

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.

The PTC uses only asynchronous events, so the asynchronous Event System channel path must be configured. By default, the PTC will detect a rising edge on the incoming event. If the PTC 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).

51. Inter-IC Sound Controller (I^2S)

51.1. Overview

The I^2S controller supports I^2S audio, with up to 8-channels.

The additional Audio codex support includes TDM allowing multiple channels of data to be transmitted on a single data line. The TDM interface is similar to the 2 channel serial audio interface I^2S, with the exception that more channels are transmitted within a sample frame or sample period. As with the I^2S interface the TDM interface is comprised of two control clocks, a serial audio left/right clk (LRCK) and serial audio bit clk (SCLK) and the serial audio data line (SDATA).

Additional TDM support for a wide variety of serial data formats. The TDM format is typically used to transfer data to or from a DSP to a MCU. The TDM format consist of three components clock, data, and frame sync (I^2S could be considered a subset of TDM).

51.2. Features

The I^2S has the following features:

- Host and client mode support
- Full-duplex operation with 8/16/20/24/32-bit communication.
- Status bit to indicate the activity of the SPI
- Four different clock formats
- Interrupt event on every byte/half-word/word received
- Separate transmit and receive buffer events
- DMA support
- SDO pin disable option
- Two 64byte FIFO data buffers are provided, one for transmit and one for receive.
- Enhanced FSYNC operation
- Clock TAP in delay
- Audio CODEC Serial Support
  - I^2S protocol
  - I^2S left justified
  - I^2S right justified
  - I^2S 32bit fp audio
  - TDM standard protocol
  - TDM left or right justified
  - TDM,(I^2S) AM824 24, 20,16-bit raw audio data
  - TDM 32-bitdata
  - TDM 24, 20,16-bit MSB aligned with mute of lower bits.
  - TDM, (I^2S) Packed 4x24 bit raw
  - TDM, (I^2S) Packed 2x16 bit raw packed upper or lower
- PCM
- I^2S Transmit Packed data (I2STPD) Host with multiple client transmit operation
- TDM - Additional features for DSP and Framed SPI host/client protocol support up to 32 slots per frame sync pulse

51.2.1. Codec Feature Set

- Codecs have complex clock divider modes to support their desired relationships between MCLK, SCLK, and LRCK
  - Most codecs support using a xtal osc so they can generate their own host clock
  - Some CODEC only support receiving MCLK (ala Cirrus LogicCS42*)
  - Codecs contain complex clocking schemes to generate their SCLK and LRCLK from MCLK.
  - Clocks are free running
  - Some codecs support using non-audio MCLK frequency such as USB 12/24/48MHz. these codecs have special divider ratios to achieve accurate PC Audio frequencies and near accurate (1.25%) CD Audio frequencies. (ala wolfson, national, adi, ti)
  - LRCK can have non 50:50 duty cycle - to support USB clocks for MCLK
  - Typical SCLK frequency is 64 Fs (LRCK)
  - Typical MCLK frequency is 125, 128,192, 250, 256, 272, 384, 512 x Fs (i.e., also the divider to achieve the sample rate from MCLK)
  - 125, 250, 272 ratios are for USB clocks
- I^{2}S interface operates in Host or Client
  - Most systems use the CODEC in host mode, in which case the CODEC provides SCLK and LRCK.
  - Systems that use the CODEC in client mode must provide a Host clock (MCLK), Serial clock (SCLK), and Sample clock (LRCK)that all have a CODEC supported relationship.
- I^{2}S for ADC and I^{2}S DAC
  - Some have two separate ports to support different sample rates for ADC and DAC
  - Some have 1 SDI and 1 SDO sharing a single LRCK and SCLK forcing the sample rate to be the same for ADC and DAC
- 16, 20, 24, 32 bit audio data sample sizes
  - Pad LSBs of smaller words to native size
  - Strips LSBs of larger words to native size
- Data direction is always MSB first
- Rx/Tx data is sample or transmitted on the rising edge, however timing diagrams show Rx data(to the DAC) driven on the falling edge
- SCLK= 25Mhz -- 4KHz to 192KHz Sample Rate
- SCLK= 50Mhz -- 4KHz to 384Khz Sample Rate
- Audio CODEC Support
  - I^{2}S protocol
  - Left Justified
  - Right Justified (DAC only)
- PCM (SPI like w/ 1-bit FSync and two data words sent - FSync occurs at Sampling Rate)

51.3. Block Diagram

Figure 51-1. I^2S Block Diagram

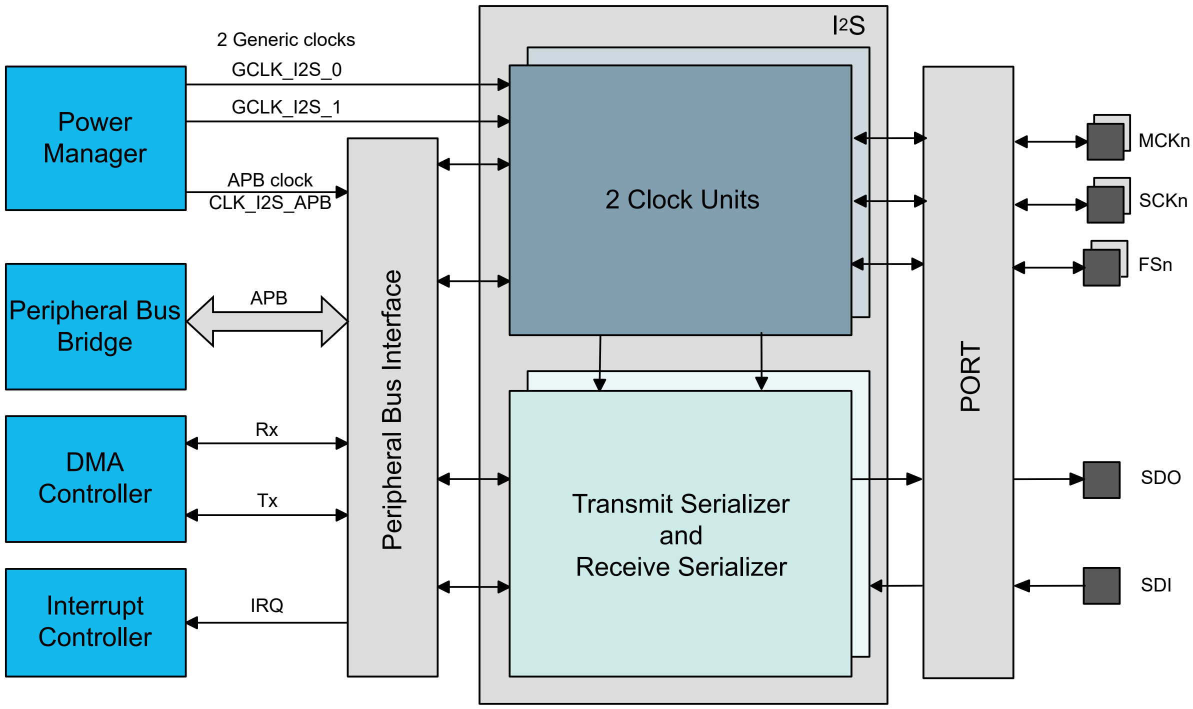
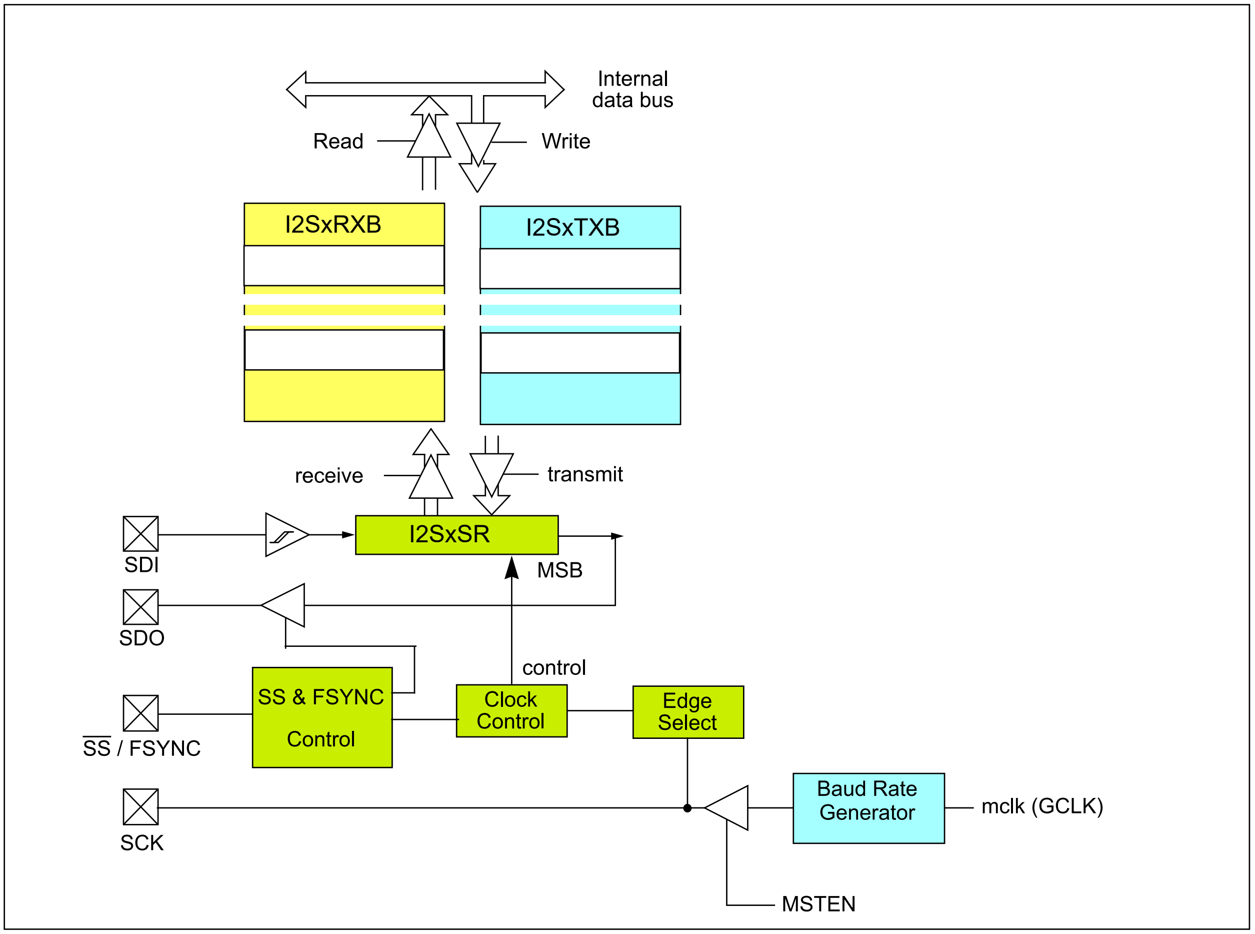


Figure 51-2. I²S Macro Block Diagram



51.4. Peripheral Dependencies

Peripheral Name	Base Address	NVIC IRQ Index: Source	MCLK AHBx/APBx Clock Enable Mask Bit	GCLK Peripheral Channel Clock Name:Register	PAC Peripheral Identifier (PAC.WRCTRL.PERIDx)	EVSYS Generator (EVSYS.CHANNELn.EVGENx)	DMA Trigger Index:Source (DMAC.CHCTRLBk.TRIGx)	Power Domain
I²S	0x4501_6000	140 : SPI_IXS	MCLK.CLKMSK3[11]	GCLK_SPI_IXS : PCHCTRL[36]	47	113 : GEN	65 : RX 66 : TX	VDDREG

51.4.1. Power Management

The peripheral will continue to operate in any sleep mode where the selected source clocks are running.

51.4.2. Clocks

One clock, the GCLK_I2Sx is a required peripheral, GCLK_I2Sx can be set to a wide range of frequencies and clock sources. The GCLK_I2Sx must be enabled and configured before use. Refer to the clock peripheral configuration section for details on the GCLK_I2Sx configuration. The clock is only used in Host mode.

51.4.3. DMA

The IxS peripheral is connected to the DMA Controller (DMAC). Using the IxS DMA requests requires the DMA Controller to be configured first (Refer to DMAC section in this document).

51.4.4. Interrupts

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

51.5. Functional Description

51.5.1. SPI Overview

The Serial Peripheral Interface (SPI) is a synchronous serial interface useful for communicating with other peripheral or microcontroller devices. These peripheral devices may be Serial EEPROMs, shift registers, display drivers, A/D converters, etc. SPI communication is full-duplex, which means that transmission and reception proceed simultaneously. It is an example of synchronous communication, as both communicating entities use the same clock signal, and is compatible with Motorola's SPI and SIOP interfaces.

Legacy compatibility is kept for the 8-bit and 16-bit modes with the addition of 32-bit operation.

51.5.2. Audio Protocol Overview

This IxS peripheral adds support for Audio CODEC serial protocols such as I^{2}S, Left Justified, Right Justified and PCM/DSP modes for 16, 24, and 32-bit audio data. The macro supports only half-duplex for audio communication. Also, the audio CODEC protocols differ from SPI as they require a free running clocks.

51.5.3. SPI Operation

When MODE_EN = 00 (AUDEN = 0 or TDM_EN =0 or TPD_EN = 0) the macro operates like a normal SPI protocol.

The serial port consists of a 32-bit register (SPIxSR) used for sending data in and out. A receive buffer (SPIxRXB) and a transmit buffer (SPIxTXB) share one SFR address, SPIxBUF. The control registers (SPIxCTRL_*) configures the peripheral, and the status register (SPIxSTAT) indicates various status conditions.

Four pins make up the serial interface:

- SDI: Serial Data Input
- SDO: Serial Data Output
- SCK: Shift Clock input or output
- $\overline{\text{SS}}$/FSYNC: Active low Client Select, or Frame Synchronization pulse

For a SPI data transfer to take place between two devices, one device must be configured as a host and the other as a client. Host mode is enabled by setting the SPIxCTRL_*.MSTEN bit. The host generates the serial clock pulses and does so only when there is data to be transmitted. In other words, the host controls the rate at which the data bits get transmitted. So, SCK is clock output in Host mode while in Client mode, it is clock input.

The $\overline{\text{SS}}$/FSYNC can be driven by the peripheral in host mode. FRMEN (FRMEN = TDM_EN || AUDEN ||TPD_EN) and SPIxCTRL_*.MSSEN register bits control this feature.

Figure 51-3. SPI Host/Client Connection

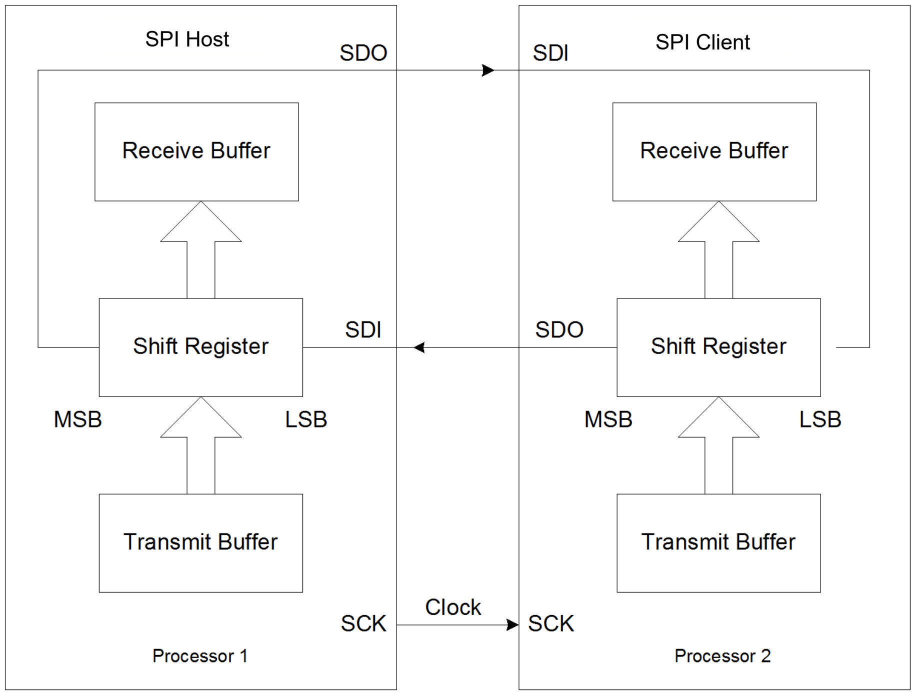
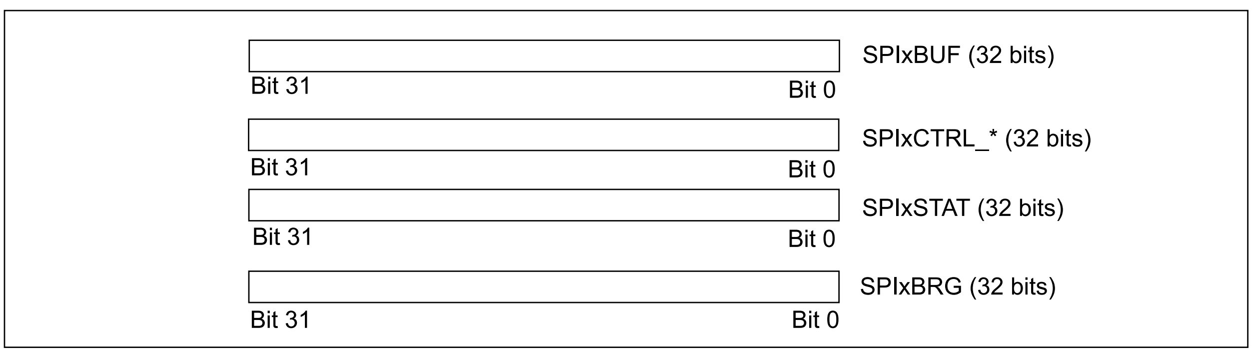


Figure 51-4. Programmer's Model



51.5.4. SPI Data Transmission/Reception

Control bits SPIxCTRL_*.MODE[32,16] allow the peripheral to communicate in either 8-bit (both MODE32 and MODE16 are cleared), 16-bit (MODE16 is set) or 32-bit (MODE32 is set) mode (See the following table). The functionality is the same for each mode except for the number of bits received and transmitted.

Table 51-1. Communication Mode Selection

MODE32	MODE16	MODE_EN	COMMUNICATION
0	0	0	8-bit
0	1	0	16-bit
1	0	0	32-bit
1	1	0	32-bit

Additionally, the following should be noted in this context:

- Data is transmitted through and received by SPIxSR
  - In 8-bit operation, data is transmitted from bit7
  - In16-bit operation, data is transmitted from bit 15
  - In32-bit operation, data is transmitted from bit 31
  - In all modes, data is not have a SHIFT register but a register that fills MSB first and MSB-1 till end
- A series of eight, sixteen, or thirty-two clock pulses send out 8/16/32 bits from the SPIxSR to the SDO pin and simultaneously store in 8/16/32-bit data from the SDI pin. An event is generated when the transfer is complete.

16/32-bit (0/1) operation is identical to 8-bit operation except that the number of bits transmitted is sixteen or thirty-two, instead of eight.

Once the peripheral is enabled (SPIxCTRL_*.ENABLE) and set up for Host mode of operation, transmission/reception will start as soon as the data is written to SPIxBUF, thereby loading the SPIxTXB register, and the transmit buffer empty status flag (SPIxSTAT.SPITBE) is cleared. This flag will automatically be set in hardware after the data is transferred from SPIxTXB to SPIxSR. Now, the data to be transmitted next can be loaded into the SPIxTXB register at any time as long as the SPIxSTAT.SPITBE bit is set. The write can occur while SPIxSR is sending the previously written data, allowing continuous transmission. When the host/client transfer finishes, the data received is moved from SPIxSR to SPIxRXB and the receive buffer full status flag (SPIxSTAT.SPIRBF) is set. This flag will automatically be cleared in hardware after the data received is read by the user software.

Note: The SPIxSR register cannot be directly written to by the user. All writes to this register are performed via the SPIxBUF register.

In Host mode, the APBm_clk is divided based on the value loaded into SPIxBRG[12:0] and then output via SCK pin to the client devices.

In Client mode, data is received and transmitted as external clock pulses appear on SCK pin and the interrupt event will be generated as the last bit of the receive data is latched.

51.5.5. Receive Overflow

If the receive buffer is full when the protocol engine needs to transfer the data from SPIxSR to SPIxRXB, the SPI sets the receive overflow status flag (SPIxSTAT.SPIROV). During the overflow condition the SPI does not complete the data transfer from SPIxSR to SPIxRXB.

51.5.5.1. Ignore Receive Overflow

The SPIxCTRL_*.IGNROV bit controls how the SPI recovers from a receive overflow condition.

If IGNROV=0 , the SPI does not push receive data into the SPIxRXB until the user clears both the overflow condition of the SPIxRXB (by reading data out of it) and the SPIROV (by writing it to zero).

If IGNROV=1, the SPI does not require the user to clear the SPIROV bit to continue to receive data into the SPIxRXB. The user only needs to make room in the SPIxRXB (by reading data out of it).

Note: For either setting of IGNROV, once SPIROV is set, it remains so until software clears it or SPIxCTRL_*.ENABLE = 0.

51.5.5.2. Receive Overflow Interrupt Enable

The SPIxCTRL_*.SPIROVEN bit controls SPI Error Interrupt generation from SPIROV. When SPIROVEN = 1, the SPI asserts its Error Interrupt persistent with SPIROV = 1. When SPIROVEN = 0, the SPI does not assert its Error Interrupt based on SPIROV.

51.5.6. Sign Extend Serial Data

The SPI allows the user to sign extend data read from the SPIxRXB. This option is only valid for serial words whose length is less than 32-bits. The data in the SPIxRXB is not changed, only on the read of the data does the sign extension occur.

When SPISGNEXT = 1, reads from the SPIxRXB provide 32-bits worth of data. The SPI simply replicates the most significant serial word bit, as defined by AUD- WD_MODE[1,0] or MODE[32,16], though bit 31 of the read data.

51.5.7. Clock Formats

The SPI peripheral supports four different serial clock formats. The user software can select one of these formats by configuring the Clock Polarity Select bits (SPIxCTRL_*.CPOL) and the Clock Edge Select bits (SPIxCTRL_*.CPHA).

The CPOL bit determines whether the serial clock is at a high logic level or a low logic level when the SPI macro is in an Idle state.

- If CPOL is set, the SCK signal is interpreted as 'active-low'.
- If CPOL is cleared, the SCK signal is interpreted as'active-high'.

The CPHA bit determines whether the serial data output changes its state on an idle-to-active transition of the serial clock or on an active-to-idle transition of the serial clock. That is, CPHA=0 means sampling on the first clock edge, while CPHA=1 means sampling on the second clock edge, regardless of whether that clock edge is rising or falling. Note that with CPHA=0, the data must be stable for a half cycle before the first clock cycle.

- If CPHA is cleared, SDO changes on an active-to-idle transition.
- If CPHA is set, SDO changes on an idle-to-active transition.

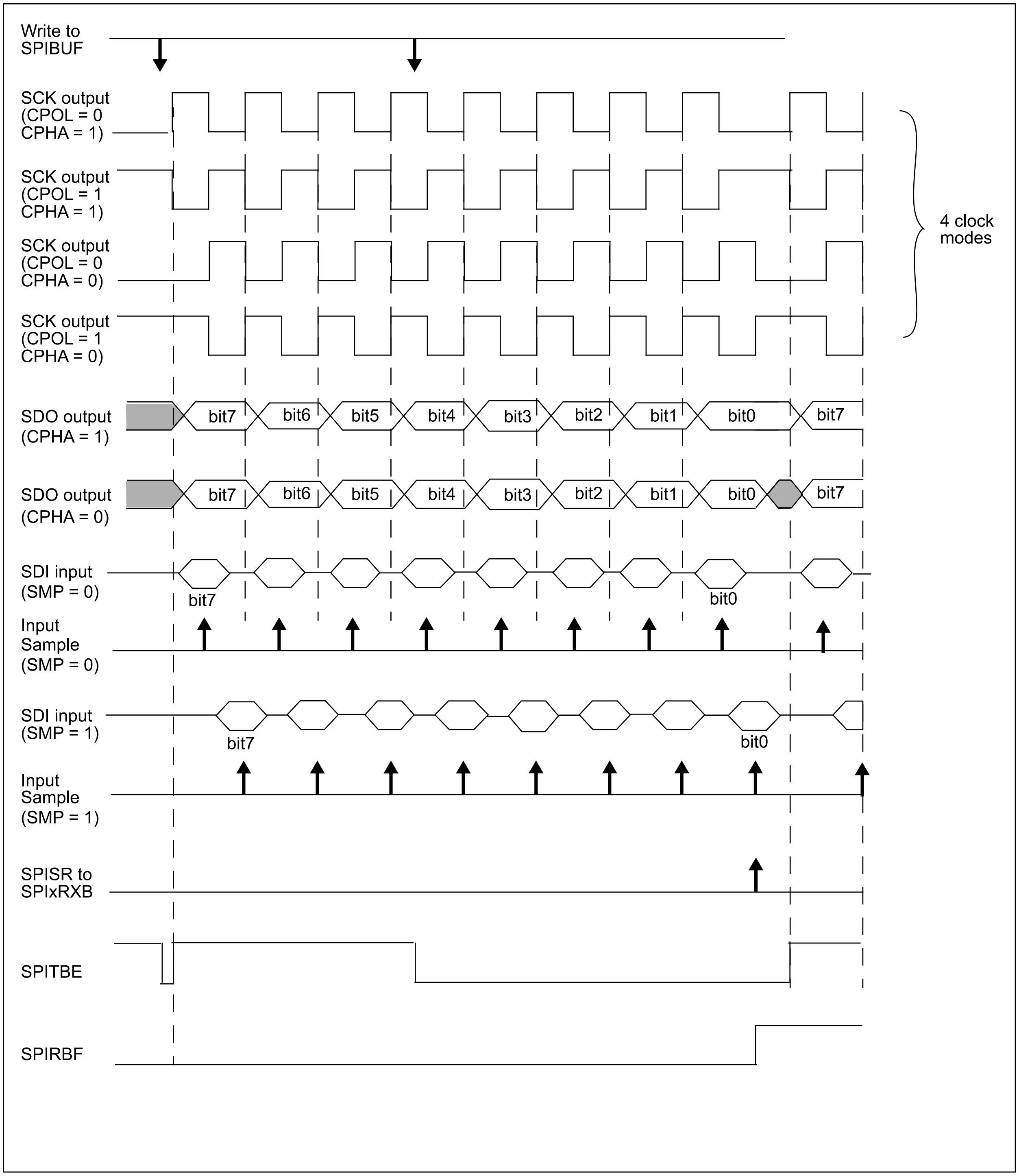
Note: The user must turn the macro off prior to changing the CPHA or CPOL bits; otherwise the behavior of the macro is not guaranteed.

Table 51-2. SPI Four Modes Combine Polarity and Phase

Mode	CPOL	CPHA	Description
SPI_MODE0	0	0	SPI_CLK is active high and sample commences on the rising edge
SPI_MODE1	0	1	SPI_CLK is active high and sample commences on the falling edge
SPI_MODE2	1	0	SPI_CLK is active low and sample commences on the rising edge
SPI_MODE3	1	1	SPI_CLK is active low and sample commences on the falling edge

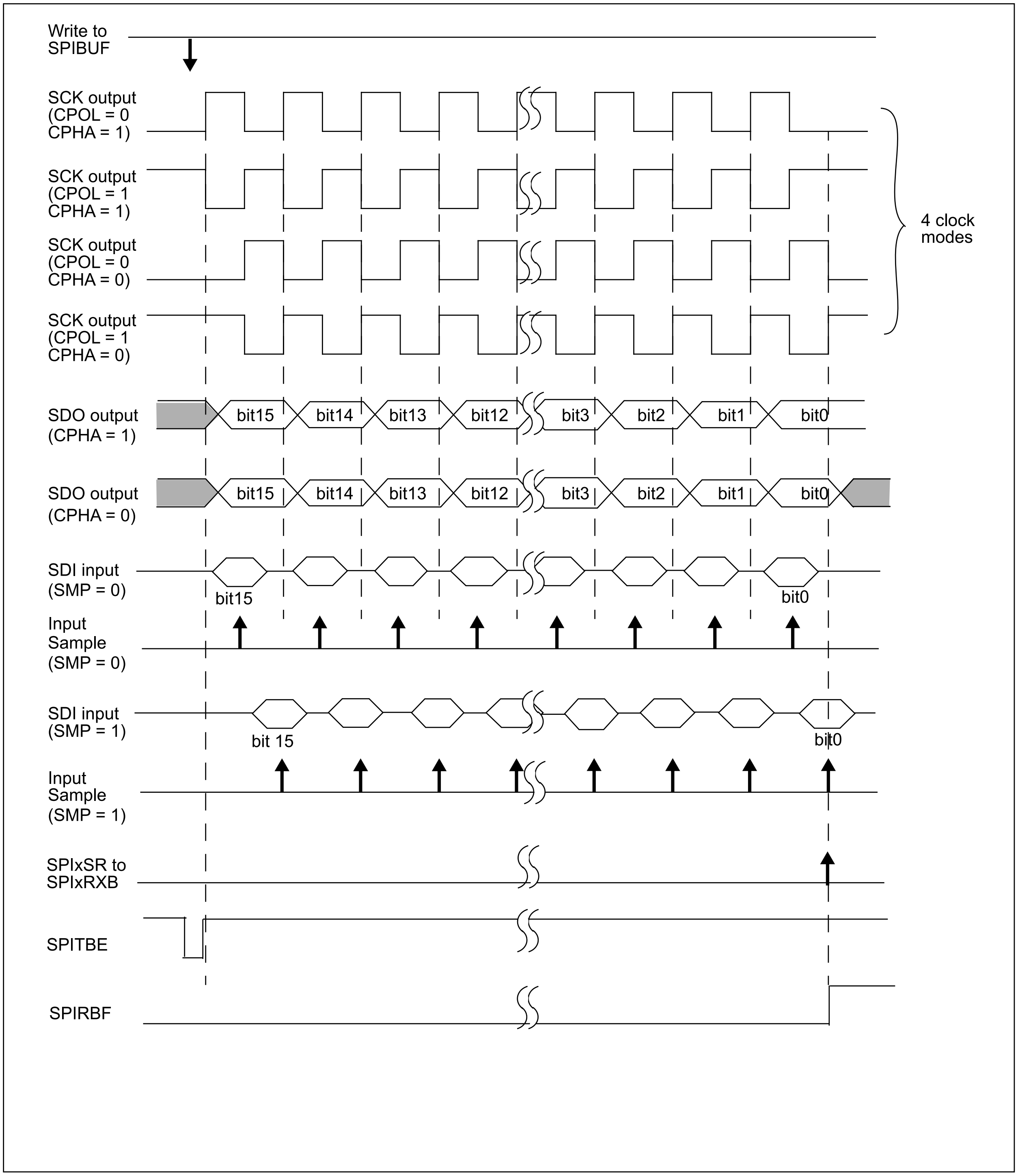
On the first write to SPIxBUF, the SPIxCTRL_*.SPITBE bit is cleared for a very short period of time; however, on the second write, it will stay cleared until the completion of the first transmission and the beginning of the next transmission, meaning that the data loaded cannot be transferred to SPIxSR until the previous transmission has completed. The macro will sample the incoming data (via SDI pin) as specified by the SPIxCTRL_*.SMP bit.

Figure 51-5. SPI Mode Timing (Host Mode; Mode16=0, Mode32=0, Mode_EN=0)



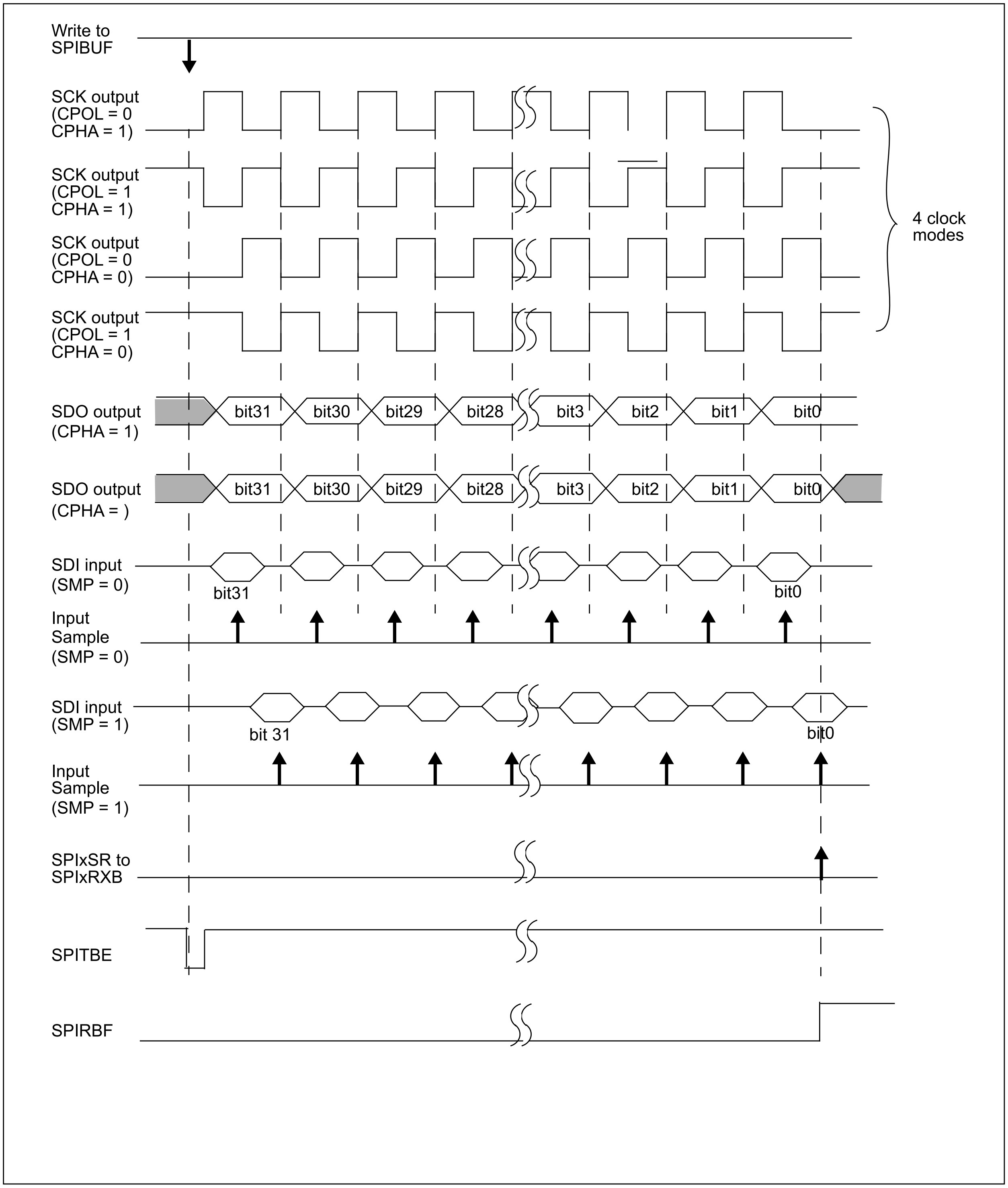
The 16-bit mode simply has more clock pulses than 8-bit mode.

Figure 51-6. SPI Mode Timing (Host Mode; Mode16=1, Mode32=0, Mode_EN=0)



The 32-bit mode simply has more clock pulses than 8-bit or 16-bit mode.

Figure 51-7. SPI Mode Timing (Host Mode; Mode32=1, Mode16=0/1, Mode_EN=0)

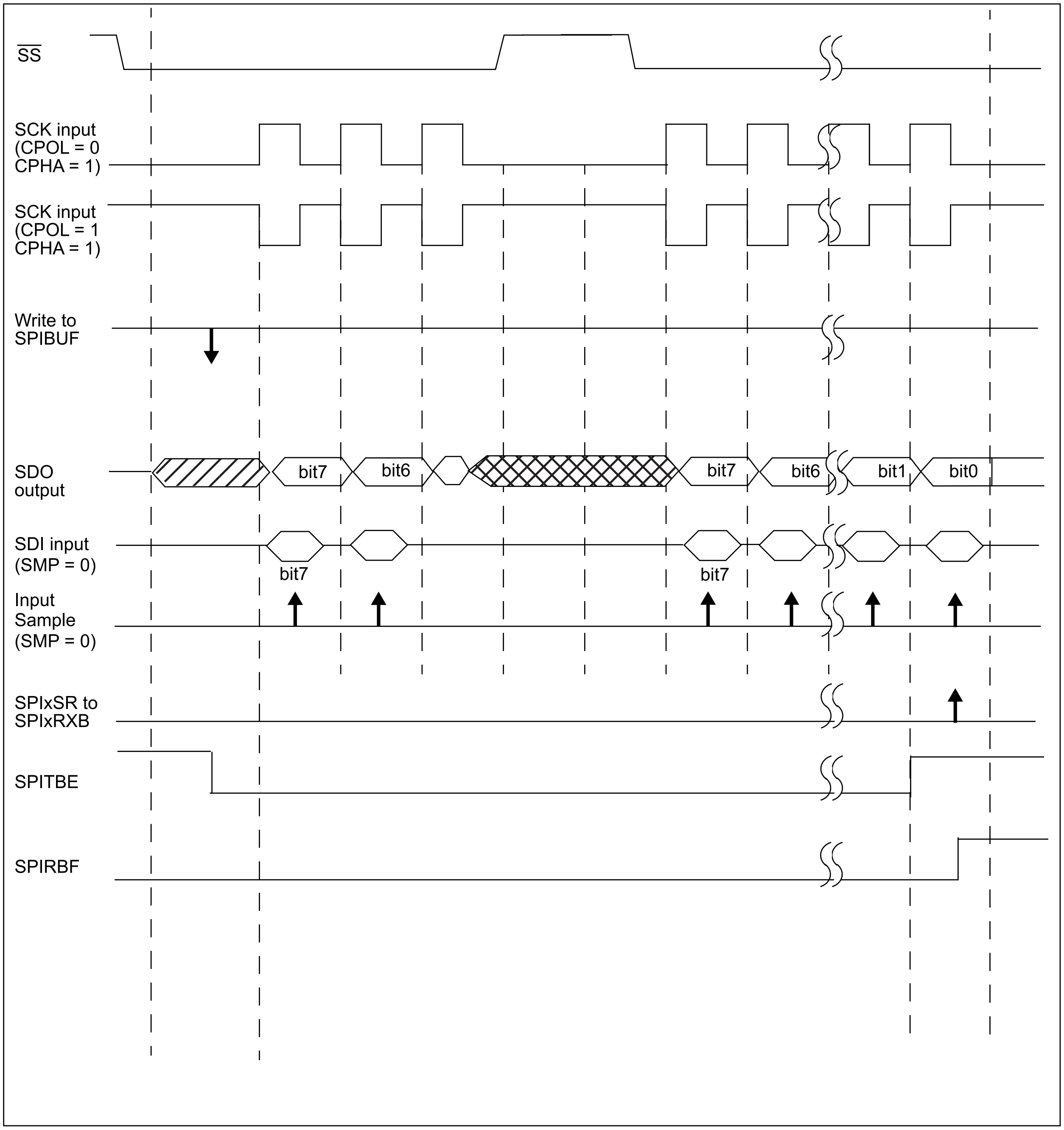


51.5.8. Client Select Synchronization

The Client Select pin ($\overline{SS}$) allows a synchronous Client mode. It can be used as a switch to enable or disable the function of a SPI client device. If the SPIxCTRL_*.MSSEN bit is set, transmission/reception is enabled in Client mode only if the $\overline{SS}$ pin is driven to a low state. If the SPIxCTRL_*.MSSEN bit is set and the $\overline{SS}$ pin is driven high, the SDO pin is no longer driven and will tri-state even in the middle of a transmission. An aborted transmission will be retried the next time the $\overline{SS}$ pin is driven low. After one transmit/receive sequence, if the $\overline{SS}$ pin is still held low, the transmission should continue with the data available in the SPIxTXB register. If the transmit buffer (SPIxTXB) were empty at that time, the data received in the store register would be sent out. If the SPIxCTRL_*.MSSEN bit is not set, the $\overline{SS}$ pin does not affect the macro operation in Client mode. The $\overline{SS}$ pin must asynchronously control the SDO tri-state function. If the $\overline{SS}$ pin is asserted again after it had been de-asserted in the middle of a transmit/receive sequence, the transmission/reception will again begin with the most significant bit of the previously aborted data.

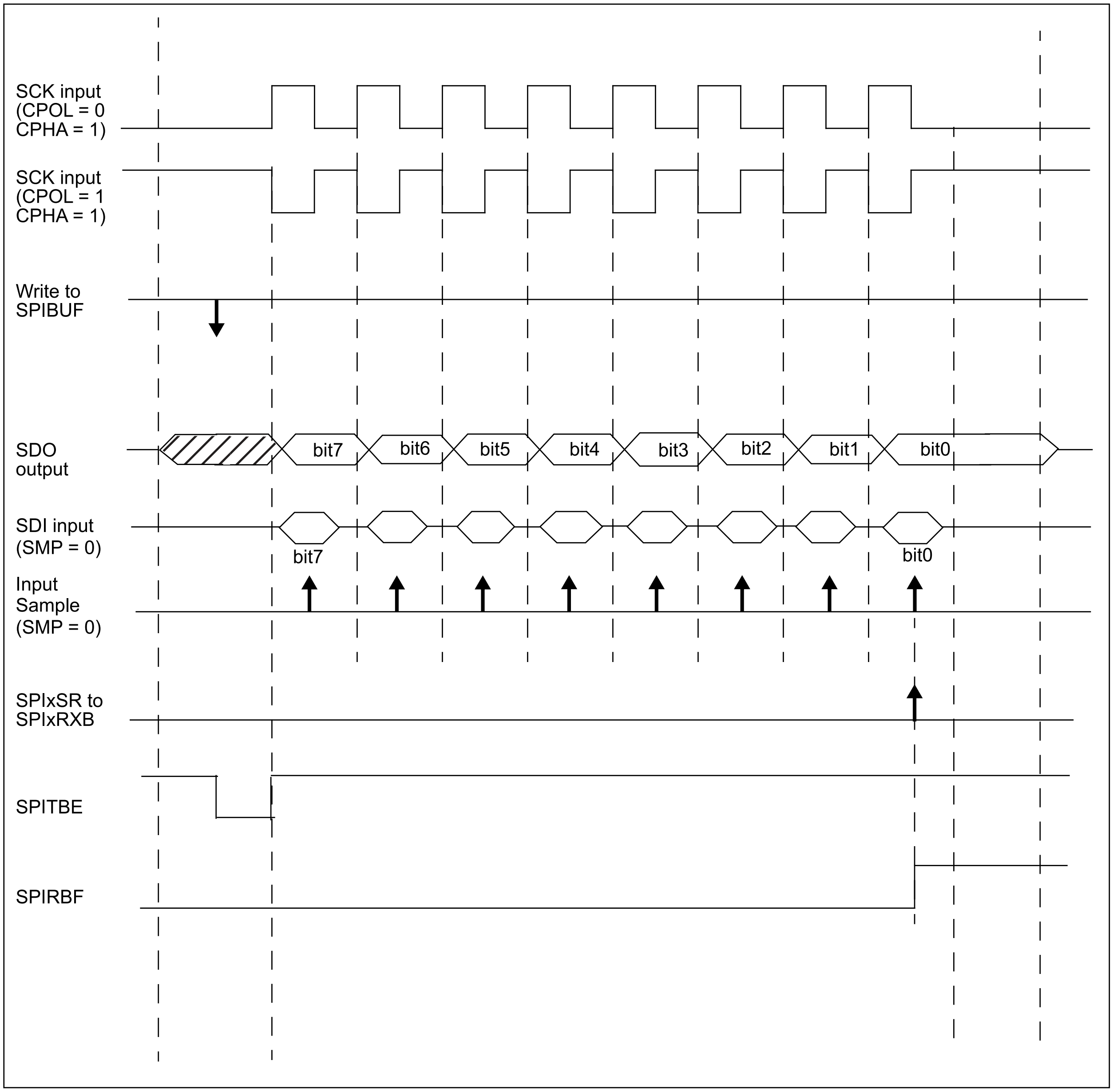
If SPIxCTRL_*.MSSEN is clear, the function of SPIxSTAT.SPITBE in Client mode is similar to that of the Host mode. When it is set, the SPIxSTAT.SPITBE is set only when the macro completes data transmission, unlike in Host mode of operation. Since transmission will be aborted at any time the $\overline{SS}$ pin goes high, to complete the transmission when $\overline{SS}$ is again driven low, the data must be held in SPIxTXB until all bits are transmitted to the receiver.

Figure 51-8. Client Synchronization Timing (Mode16=0, Mode32=0, MSSEN=1, Mode_EN=0)



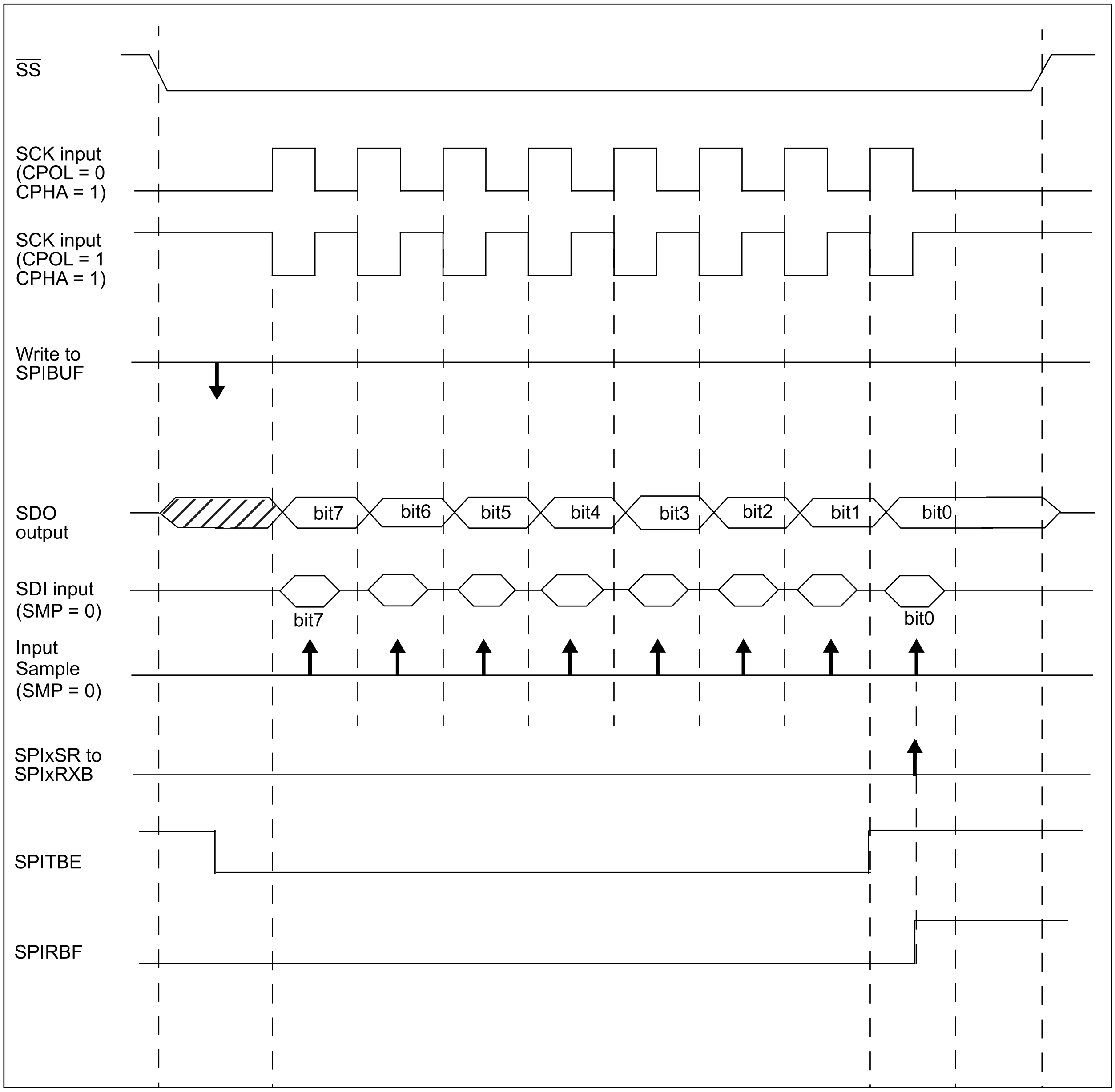
In Client mode, the clocks are input via the SCK pin. In this case, SPIxCTRL_*.MSSEN = 0, and therefore, the $\overline{\text{SS}}$ pin is unused. The incoming data is sampled at the middle of data output time.

Figure 51-9. SPI Mode Timing (Client Mode W/CPHA=1 and MSSEN=0; Mode16=0, Mode32=0, Mode_EN=0



In this case, SPIxCTRL_*.MSSEN = 1, and therefore, the $\overline{SS}$ pin is used by the macro. Since the transmission may be interrupted when the $\overline{SS}$ pin goes high, the behavior of the SPIxSTAT.SPITBE bit is different than when SPIxCTRL_*.MSSEN = 0.

Figure 51-10. SPI Mode Timing (Client Mode WW/CPHA=1 and MSSEN=1; Mode16=0, Mode32=0, Mode_EN=0



51.5.9. SPIXBUF Register Operation

51.5.9.1. Standard Buffer Mode

In standard buffer mode, the transmit buffer and receive buffer are each one element deep. Writes to the SPIxBUF are clocked in to the SPIxTXB register. Reads to the SPIBUF are read out of the SPIxRXB register. The data from these registers is not synchronized. As a consequence of this, the CPU must never read out of this register (SPIxBUF) until the SPIRBF is set, and must never write to this register until the SPITBF is cleared.

51.5.9.2. Buffer Mode

The buffer mode, a multi-element FIFO synchronizes data between the CPU clock domain and the SCK domain. The FIFO depth is always 64 bytes deep. AUDWD_MODE[1:0] field selects 32(24,20)-bit data lengths. If AUDWD_MODE selects 16-bit data lengths. If AUDWD_MODE selects 8-bit data lengths.

The SPIxBUF provides access for both the Rx FIFO and the Tx FIFO. CPU reads complete from the Rx FIFO and CPU writes complete into the Tx FIFO. There is no protection (status of an error) against reading an empty Rx FIFO or writing a full Tx FIFO. However, the SPI Shift Register (SPIxSR) provides Transmit Under-run (SPITUR) and Receive Overflow (SPIROV) status.

The SPI provides four status flags for determining the state of Tx and Rx data in the FIFOS and SPIxSR. The SPIxSTAT.SPIBUSY bit indicates the SPIxSR is sending/receiving data or copying data to/from a FIFO. The SPIxSTAT.SRMT indicates if the SPIxSR is empty or not.

51.5.9.3. Transmit Buffer Under-run in Client Mode

When configured for non-framed, client mode with CPHA = 1, the SPI must drive the first bit of data on SDA with out using SCK. Therefore, the first bit of data is driven directly from the SPIxTXB and the second and subsequent bits from the SPIxSR. This condition causes asynchronous interaction with the SPIxTXB. When software can keep up with the selected data transfer rate of the SPI, the first bit of data driven out of the SPIxTXB has plenty of setup time.

However, if software cannot fully keep up with the configured transfer rate causing the SPIxTXB to be empty at the start of a transaction, one of two things can happen. If the SPIxTXB stays empty until after the first bit transmits, the SPI transmits the previous data. If data arrives in the SPIxTXB during the first bit time, the SPIxTXB drives the first bit on SDA and pushes the data into the SPIxSR. When the SPIxTXB becomes not empty during the first bit time, the first bit cannot be guaranteed to meet setup time.

To ensure this case does not cause issues, software needs to keep up with the selected transfer rate.

51.5.10. SDO Disable

The DISSDO bit determines if the SPI uses the SDO pin. Changing the value of DISSDO immediately changes the control of the SDO pin between the SPI and Port function. The DISSDO bit is applicable to all SPI operating modes.

If DISSDO = 0, the SPI pops data from the SPIxTXB into the register and transmits data through the SDO pin. If DISSDI = 1, the SPI does not pop data from the SPIxTXB and does not use the SDO pin.

DISSDO can be modified while transmitting data. However, doing so usually corrupts the data in that word as the Port logic supplies the values for the remaining bits.

51.5.11. SDI Disable

The DISSDI bit determines if the SPI uses the SDI pin. Changing the value of DISSDI immediately changes the control of the SDI pin between the SPI and Port function. The DISSDI bit is applicable to all SPI operating modes.

If DISSDI = 0, the SPI receives data through the SDI pin and pushes data into the SPIxRXB. If DISSDI = 1, the SPI does not use the SDI pin and does not push data into the SPIxRXB.

DISSDI can be modified while receiving data. However, disabling SDI in the middle of a receive transaction usually results in corrupting that word of receive data (which is also the last word received).

51.5.12. Interrupt Events

The interrupt events from the peripheral are persistent, which means they remain active until the condition that generated the interrupt is cleared by the CPU.

Note: In order to clear any pending interrupt flag in the status register, it is recommended to use the SPIxSTATCLR SFR instead of directly writing the SPIxSTAT SFR.

51.5.12.1. Standard Buffer Mode Interrupts

In standard buffer mode, when the transmit buffer is empty, the SPI generates a transmit service request interrupt. This activity matches the activity of SPITBE. When the receive buffer is full, the SPI generates a receive service request interrupt. This activity matches the activity of SPIRBF.

51.5.12.2. Buffer Mode Interrupts

The SPI generates transmit and receive service request interrupts based on the amount of data in the respective FIFO. Software controls this level independently for transmit and receive using the SPIxCTRL_*.STXISEL and SPIxCTRL_*.SRXISEL fields.

The generation of the service request interrupts does not necessarily match the operation of SPITBE and SPIRBF.

51.5.12.3. Error Interrupts

The peripheral reports error status for receive overflow (SPIROV), transmit underrun (SPITUR) and frame error (FRMERR). If enabled by SPIROVEN, SPITUREN, or FRMERREN, respectively, the SPI peripheral generates an error event interrupt.

The SPI sets SPIROV if it receives a new serial-word when receive buffer is full (as defined by SPIRBF).

For FRMEN=1, the SPI sets SPITUR if it detects a transmit underrun condition.

The SPI sets FRMERR if it detects multiple frame sync pulses during a burst.

51.5.13. DMA Support

The buffer mode of operation supports selectable DMA burst lengths. This feature allows system designers to tune SPI and DMA interaction to account for system latency and throughput.

51.5.14. Host Mode Clocking

The SPI macro has a 13-bit baud rate generator to allow flexibility in baud rate generation. The SPIxBRG register is readable and writable and determines the baud rate. The GCLK_IxS provided to the peripheral is a divider function of the CPU core clock. This clock can then again be divided, based on the value loaded in the SPIxBRG register, before it is provided to the external devices via the SCK pin. The SCK clock is of 50% duty cycle. Some sample SPI clock frequencies are shown in Sample SCK Frequencies.

Note: In normal SPI mode, the host clock (via SCK pin) is not free-running. It will only run for 8, 16 or 32 pulses when SPIxBUF is loaded with data. However, the clock will run continuously in Framed SPI mode a subset of the TDM mode.

51.5.14.1. SCK Frequency Calculation

F_{PB} = Peripheral Bus clock frequency

Baud Rate = F_{PB} / (2* (SPIxBRG + 1))

Therefore, the maximum baud rate possible is F_{PB}/2 (SPIXBRG = 0) and the minimum baud rate possible is F_{PB}/16384.

Table 51-3. Sample SCK Frequencies

				SPIxBRG				
APBm_clk	1	15	31	63	85	127	255	511
100.00E+6	25.00E+6	3.13E+6	1.56E+6	781.25E+3	581.40E+3	390.63E+3	195.31E+3	97.66E+3
				SPIxBRG				
APBm_clk	1	15	31	63	85	127	255	511
96.00E+6	24.00E+6	3.00E+6	1.50E+6	750.00E+3	558.14E+3	375.00E+3	187.50E+3	93.75E+3
				SPIxBRG				
APBm_clk	1	15	31	63	85	127	255	511
80.00E+6	20.00E+6	2.50E+6	1.25E+6	625.00E+3	465.12E+3	312.50E+3	156.25E+3	78.13E+3
				SPIxBRG				
APBm_clk	1	15	31	63	85	127	255	511
60.00E+6	15.00E+6	1.88E+6	937.50E+3	468.75E+3	348.84E+3	234.38E+3	117.19E+3	58.59E+3
				SPIxBRG				

APBm_clk	0	15	31	63	85	127	255	511
50.00E+6	25.00E+6	1.56E+6	781.25E+3	390.63E+3	290.70E+3	195.31E+3	97.66E+3	48.83E+3
SPIxBRG								
APBm_clk	0	15	31	63	85	127	255	511
48.00E+6	24.00E+6	1.50E+6	750.00E+3	375.00E+3	279.07E+3	187.50E+3	93.75E+3	46.88E+3
SPIxBRG								
APBm_clk	0	15	31	63	85	127	255	511
40.00E+6	20.00E+6	1.25E+6	625.00E+3	312.50E+3	232.56E+3	156.25E+3	78.13E+3	39.06E+3
SPIxBRG								
APBm_clk	0	15	31	63	85	127	255	511
33.00E+6	16.50E+6	1.03E+6	515.63E+3	257.81E+3	191.86E+3	128.91E+3	64.45E+3	32.23E+3
SPIxBRG								
APBm_clk	0	15	31	63	85	127	255	511
25.00E+6	12.50E+6	781.25E+3	390.63E+3	195.31E+3	145.35E+3	97.66E+3	48.83E+3	24.41E+3
SPIxBRG								
APBm_clk	0	15	31	63	85	127	255	511
20.00E+6	10.00E+6	625.00E+3	312.50E+3	156.25E+3	116.28E+3	78.13E+3	39.06E+3	19.53E+3
SPIxBRG								
APBm_clk	0	15	31	63	85	127	255	511
12.00E+6	6.00E+6	375.00E+3	187.50E+3	93.75E+3	69.77E+3	46.88E+3	23.44E+3	11.72E+3
SPIxBRG								
APBm_clk	0	15	31	63	85	127	255	511
10.00E+6	5.00E+6	312.50E+3	156.25E+3	78.13E+3	58.14E+3	39.06E+3	19.53E+3	9.77E+3
SPIxBRG								
APBm_clk	0	15	31	63	85	127	255	511
8.00E+6	4.00E+6	250.00E+3	125.00E+3	62.50E+3	46.51E+3	31.25E+3	15.63E+3	7.81E+3

51.5.15. Audio Protocol Operation

The register SPIxCTRL_* contains the audio specific protocol control bits. When AUDEN = 1 the macro operates like an audio CODEC host or client. The audio protocol requires certain features of the SPI protocol and therefore overrides some SPI settings.

The macro uses the serial audio protocol defined by AUDMOD, AUDWD_MODE, and AUDFMT. In each of the modes the serial clock is free running and audio data is always transferred.

Four pins make up the serial interface. However, each audio connection is only half-duplex so SDO exists only on the transmit side and SDI exists only on the receive side of the interface. The four pins are:

- SDI: Serial Data Input
- SDO: Serial Data Output
- SCK: Serial Clock
- LRC: Left/Right Clock (on $\overline{SS}$/FSYNC)

Some codecs refer to Serial Clock (SCK) as Baud/Bit Clock (BCLK). Also, the Left/Right Clock is commonly referred to as LRC or LRCK. The I^2S (and other audio) protocol refers to LRC as Word Select (WS). This section refers to signal on SS/FSYNC as LRC (to be consistent with CODEC naming conventions).

An Audio Protocol data transfer takes place between two devices. Usually one device is the host and the other is the client. However, audio data can be transferred between two clients. Because the audio protocols require free running clocks the Host can be a third party controller. In either case the Host generates two free running clocks: SCK and LRC. The (clock) host must generate LRC and SCK continuously, regardless of the availability of transmit data.

Host mode for Audio is enabled by setting the SPIxCTRL_*.MSTEN bit. The clocks (SCK and LRC) continue to generate pulses as long as SPIxCTRL_*.ENABLE = 1 and AUDEN = 1. In other words, the host controls the rate at which the data bits get transmitted. So, SCK is clock output in Host mode while in Client mode, it is clock input.

Most codecs require a host clock, usually called MCLK. MCLK provides a higher speed clock that has a timing relationship to the Sample Frequency (fs) and therefore SCK and LRC. Codecs that require an MCLK are almost always the host device. Many codecs are capable of generating their own MCLK from a crystal. Using an audio crystal with the CODEC provides accurate audio sample rates.

Certain codecs that require MCLK but without a crystal oscillator may still be usable. In this case codecs that support USB frequencies for MCLK (i.e., 12MHz) work best (because of their supported divider ratios). Also, the micro-controller that uses this macro must have a Clock Out feature for the USB clock. Using a USB frequency for MCLK does not provide 100% accurate audio sample rates for all audio frequencies.

51.5.15.1. Audio at Start Up

This section describes how the Audio Protocol behaves immediately after startup. Audio startup is defined as AUDEN = 1 and the ON bit being written from a 0 to a 1.

After startup, when configured for Client Mode, the SPI drives zeros out of SDO but does not send data out or in (SDI) until it receives the leading edge of LRC (i.e. the edge that precedes the left channel). Once it receives the leading edge of LRC, it starts receiving data (if DISSDI = 0). If there is no data in the TX FIFO, the I^2S continues to transmit zeros.

After audio startup, when configured for Host Mode, the SPI drives the leading edge of LRC and SCK within 1 SCK period. If the TX FIFO is empty the SPI continues to transmit zeros.

For either Client or Host Mode, the SPI does not generate an underrun on the TX FIFO after startup. This allows software to setup the SPI, setup the DMA, turn on the SPI's Audio Protocol and then turn on the DMA without getting an error. After the first write to the TX FIFO (SPIxBUF), the SPI enables underrun detection and generation.

Also, to keep the RX FIFO empty until the DMA is enabled, set DISSDI = 1. After enabling the DMA, set DISSDI = 0 to start receiving.

51.5.15.2. Audio Data Length

The SPIxCTRL_*.AUDWD_MODE[1:0] determines the audio data length. In Audio Protocol Mode it selects different lengths than it does in SPI Mode. It supports audio data transmit/receive lengths of 16, 20, 24, and 32 bits. Actual data can be any length up to 32-bits, but must be packed in one of those four formats.

AUDWD_MODE [1,0] also controls the (left/right) channel length or the I^8S Audio format protocols which can be different than the audio data length. For I^2S format16-bit data it provides the option of either a 16-bit channel or a 32-bit channel. For 20-bit, 24-bit and 32-bit data or I8S formats the channel is always comprised of a 32-bit word. Channel length inherently controls the Frame length as a Frame of audio data is made up of two channels. For I^2S 16-bit channels, a Frame is 32 Serial Clocks. For I^2S 32-bit channels, a Frame is 64 Serial Clocks, and for all I^8S modes a frame is 256 Serial Clocks

Further, AUDWD_MODE[1,0] determines the width of the data in the FIFO. For 32, 24, and, 20-bit audio data (and some 16-bit modes), the FIFO data is 32-bits wide and for some 16-bit audio modes, the FIFO is 16-bits data. The FIFO supports data writes of 8-bits, 16-bits, 20-bits, 24-bits, or 32-bits per transaction. However, if the written data is a greater length than selected, the upper bytes are ignored. Also, if the written data is a lesser length than selected, the FIFO pointers change on the write to the Most Significant Byte of the selected length.

For example, the audio data length is set to 24-bits and the data writes are 8-bits each. Software then writes a byte at a time to the FIFO starting at its lowest address, offset 0020h. The next byte write is to address offset 0021h, followed by 0022h. On the write the 0022h the data is fully pushed into the FIFO and the next audio data word write must be to offset 0020h.

Data written to unused bytes is ignored. Also, transactions that are only to unused bytes are also ignored. Therefore a byte write to address offset 0023h is completely ignored and does not cause a FIFO push.

51.5.15.3. Audio Data Alignment

Audio data is transmitted through and received by SPIxSR. Regardless of the selected alignment on the serial interface.

- In 16-bit operation, data is transmitted from bit15
- In 20-bit operation (24bits-lower bits padded 0), data is transmitted from bit 23
- In 24-bit operation, data is transmitted from bit 23
- In 32-bit operation, data is transmitted from bit 31

51.5.15.4. LRC Error (Frame Error)

LRC (frame) errors produce different behavior in Audio mode than in Framed SPI mode. Though the FRMERR and FRMERREN behave the same, the check happens on every LRC edge that defines a channel start. An LRC error occurs when an LRC edge occurs before the correct number of bits (as defined by MODE[32:16] or AUDWD_MODE[1:0]) in a channel completes. Note that in DSP/PCM mode the check only occurs on the edge that defines the combined left/right channel start.

When an LRC error occurs, the SPI immediately set FRMERR. It pushes the data in the SPIxSR into the SPIxRXB and pops data from the SPIxTXB into the SPIxSR. Under this condition, the receive and transmit data is almost certainly corrupted, but the actions of the SPI preserve the audio mode.

The SPI provides detection of FRMERR for LRC for debugging. Since LRC is a free-running clock, its period is expected to be constant, so this error condition would not be automatically recoverable in a system.

51.5.15.5. Audio Protocol Modes

The Audio function supports several protocol modes of operation using the AUDMOD, FRMCNT, FRMSYPW and AUDFMT registers. I2STPD - Hosts with clients Transmit Packed data mode is supported using the TPD_EN, DATFMT_LR, MST_SLV_EN[4:0], SLV_M-ST_UPPR, FRMCNT, and FRMSYPW registers. TDM mode is supported using the FRMCNT, TDMSSZ, TDMWSZ, TDM_EN, DATFILL, FRMCNT, and FRMSYPW. The macro uses these modes to communicate with different types of codecs. These modes control the edge relationships of LRC and SDI/SDO with respect to SCK.

For most combinations of audio data length, channel length and frame length, there are at least as many or more serial clocks than data to transmit. When this macro is the transmitter it pads all extra clocks with zeros.

All protocol modes transmit MSB first, followed by MSB-1, and so on, until the LSB transmits. Unlike the I^2S (I^8S) standard all Audio Protocol functions (including I^2S. I^8S) implemented by SPI require at least as many or more serial clocks as data to transmit or receive the audio data correctly.

Clock requirements for SCK and LRC differ between host and client setting. The client setting is less stringent. When a host, the Audio Protocol function only supports frame sizes of 32, 64, or 256 clocks. However, when a client and I^2S legacy mode AUDFMT=000, the Audio Protocol function only requires at least the number of clocks selected by MODE[32,16] or AUDWD_MODE[1,0], other wise in client mode and with new Audio Protocol functions selected AUDFMT ~= 000, the number of clocks required is 32 clocks per channel word.

I^8S is allowing multiple channels of data to be transmitted on a single data line. The I^8S interface is similar to the 2 channel serial audio interface I^2S with the exception that more channels are transmitted within a sample frame or sample period, basically defined as an 8 channel TDM interface (left or right justified) with a 50% duty cycle LRC clock. As with the I^2S interface the I^8S

interface is comprised of two control clocks, a frame synchronization pulse (LRC), a serial clock (SCK) and the serial audio data line (SDO/SDI). Several Audio Modes can be selected, i.e., I^2S, I^8S, right justified, left justified, packed 24x4bit raw, packed 2x16 raw, Host with clients transmit Packed data and TDM.

51.5.15.5.1. I^2S (AUDMOD=00, AUDFMT=000)

In I^2S mode, the transmitter drives the MS of the audio data on the first falling edge of SCK after an LRC transition. The receiver samples the MSB on the second rising edge of SCK. The left channel data transmits while LRC is low and the right channel transmits while LRC is high. A frame transmits left channel first then right channel.

To be I^2S compliant, the configuration bits in SPIxCTRL_* must be set as follows: AUDEN=1, AUD- MOD=00, FRMPOL=0, CPOL=1, CPHA = 1, FRMSYPW=0001, FRMCNT=001, AUDFMT[2:0] =000, FRMCOINC=0. These values set SDO and LRC transitions to occur on the falling edge of SCK and sampling of SDI to occur on the rising edge of SCK. It also starts a frame with LRC falling edge transition.

Figure 51-11. I^2S with 16-bit Data/Channel or 32-bit Data/Channel

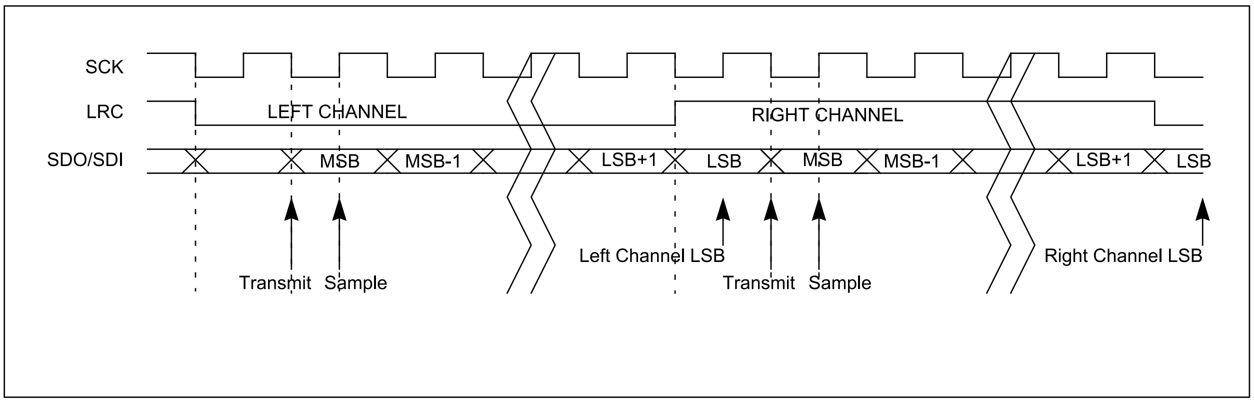
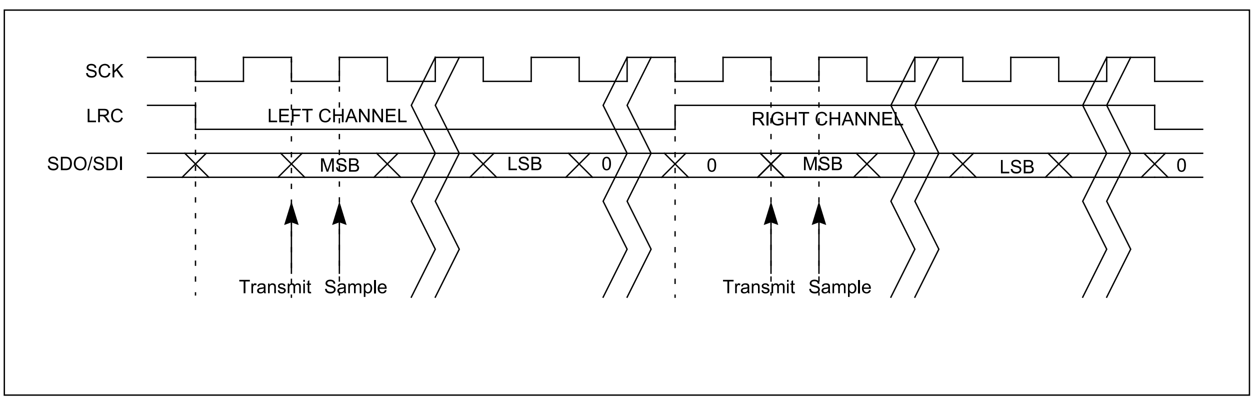


Figure 51-12. I^2S with 16/20/24-bit Data and 32-bit Channel



51.5.15.5.2. Left Justified (AUDMOD=01 AUDFMT=000)

In Left Justified mode, the transmitter drives the audio data's MSB on the SCK edge that is coincident with an LRC transition. The receiver samples the MSB on the next SCK edge.

Codecs using justified protocols usually default to transmitting data on the rising edge of SCK and receiving data on the falling edge of SCK. Another convention is that LRC is high for the left channel and low for the right channel which is opposite of I^2S. But they maintain left channel followed by right channel (in a frame).

Many codecs support other options but to configure for the left justified standard convention set the following bits in SPIxCTRL_* as follows: AUDEN=1, AUD- MOD=01, FRMPOL=1, CPOL=0, CPHA = 1, FRM- SYPW=0001, FRMCNT=001, AUDFMT[2:0] = 000, FRMEN=1. FRMCOINC=1. The following figures show waveforms for this configuration.

Format for 20 bit Audio Data and 32-bit channel, transmitted or received as if it were a 24bit data word made up of the 20 bit sample data MSB left justified in the 24 bit word and remaining 4 bits are zeros, then left justified mode as a 24 bit word.

Figure 51-13. Left Justified with 16-bit Data/Channel or 32-bit Data/Channel

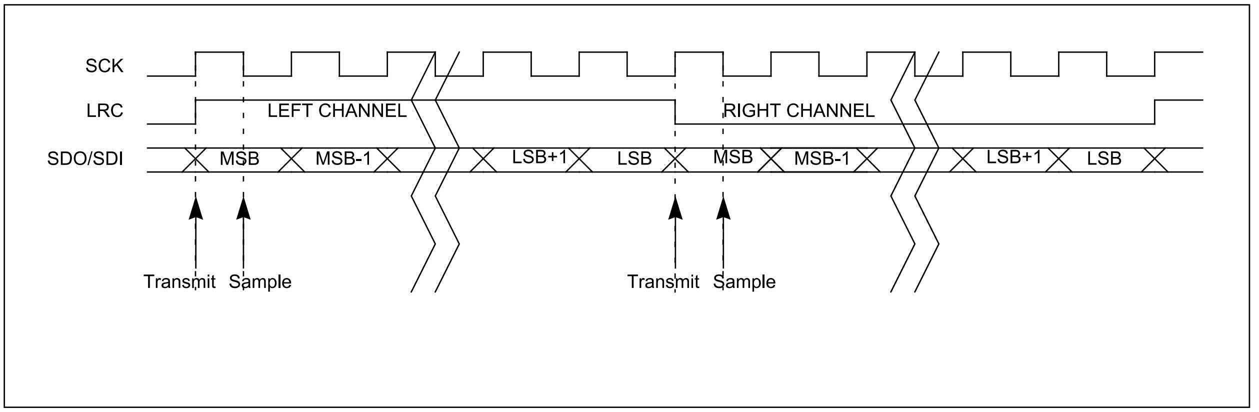
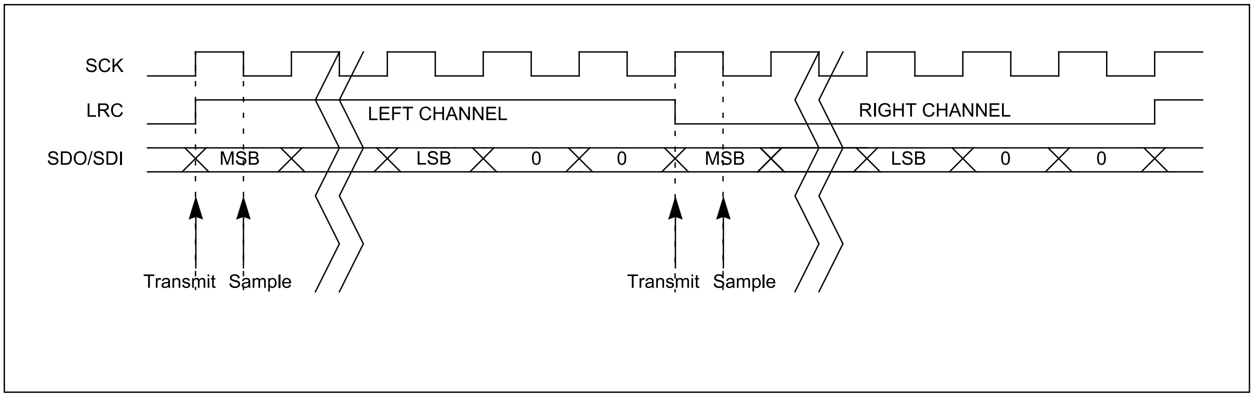


Figure 51-14. Left Justified with 16/20/24-bit Data and 32-bit Channel



51.5.15.5.3. Right Justified (AUDMOD=10 AUDFMT=000)

In Right Justified mode, the transmitter drives the audio data’s MSB on the nth transmit edge of SCK such that the LSB is available on the receive edge of SCK preceding a transition of LRC.

When set to transmit (DISSDO = 0), this device drives the unused bit slots (preceding the audio data) with logic level 0. When set to receive (DISSDI = 0), this device ignores the unused bit slot.

The following figures show right justified mode configured as follows: AUDEN=1, AUDMOD=10, FRMPOL=1, CPOL=0, CPHA = 1, FRMSYPW=0001, FRMCNT=001, AUDFMT[2:0] = 000, FRMCOINC=1.

Format for 20 bit Audio Data and 32-bit channel, transmitted or received as if it was a 24bit data word made up of the 20 bit sample data MSB left justified in the 24 bit word and remaining 4 bits are zeros, then right justified mode as a 24 bit word.

Figure 51-15. Right Justified with 16-bit Data/Channel or 32-bit Data/Channel

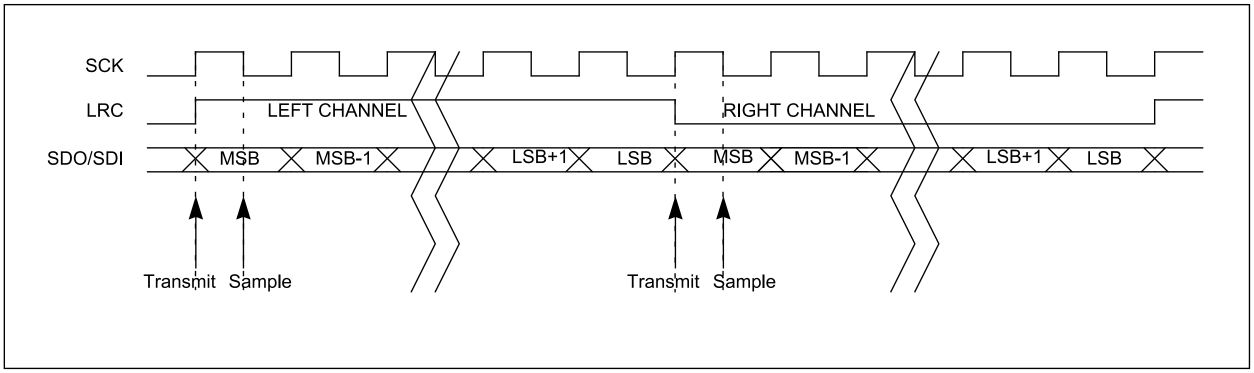
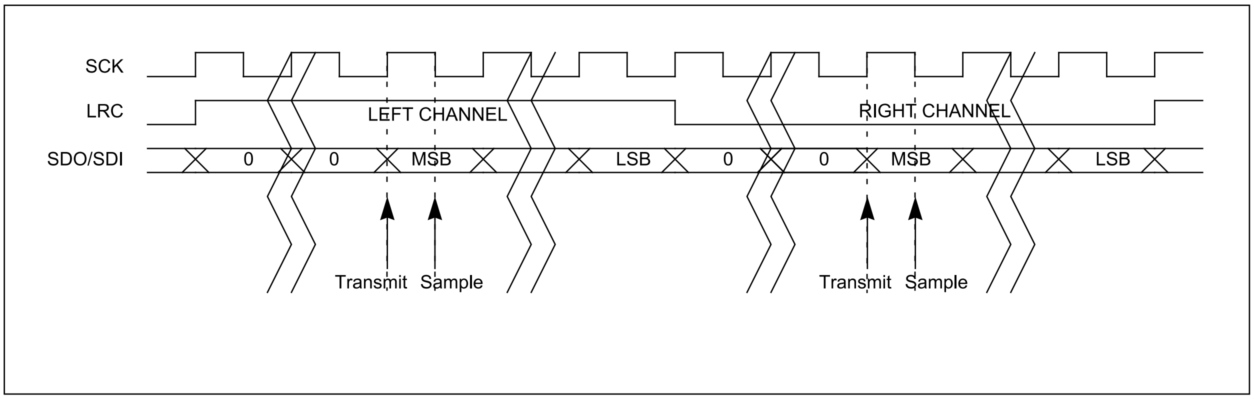


Figure 51-16. Right Justified with 16/20/24-bit Data and 32-bit Channel



51.5.15.5.4. PCM/DSP (AUDMOD=11 AUDFMT=000)

The PCM/DSP protocol mode is available for communication with some codecs and certain DSP devices. This mode modifies the behavior of LRC and audio data spacing.

In PCM/DSP mode the LRC can be single bit wide (i.e., 1 SCK) or as wide as the audio data (16,20,24,32-bits). The audio data is packed in the frame with the left channel data immediately followed by the right channel data. The frame length is still either 32 or 64 clocks when this device is the host.

In PCM/DSP mode, the transmitter drives the audio data's (left channel) MSB on the first or second transmit edge (See SPIxCTRL_*.FRMCOINC) of SCK (after an LRC transition). Immediately after the (left channel) LSB, the transmitter drives the (right channel) MSB.

Figure 51-17. PCM/DSP with 16-bit Data/Channel or 32-bit Data/Channel

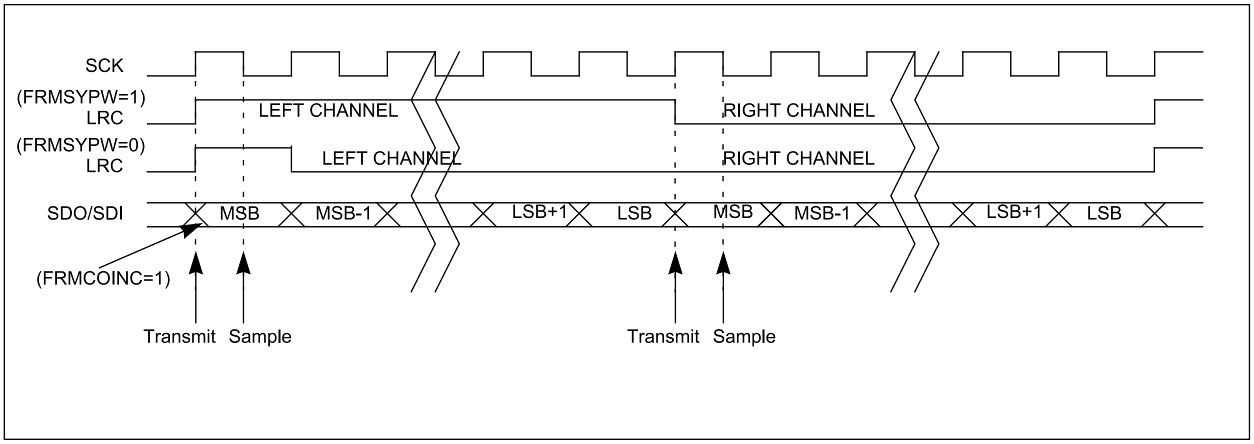
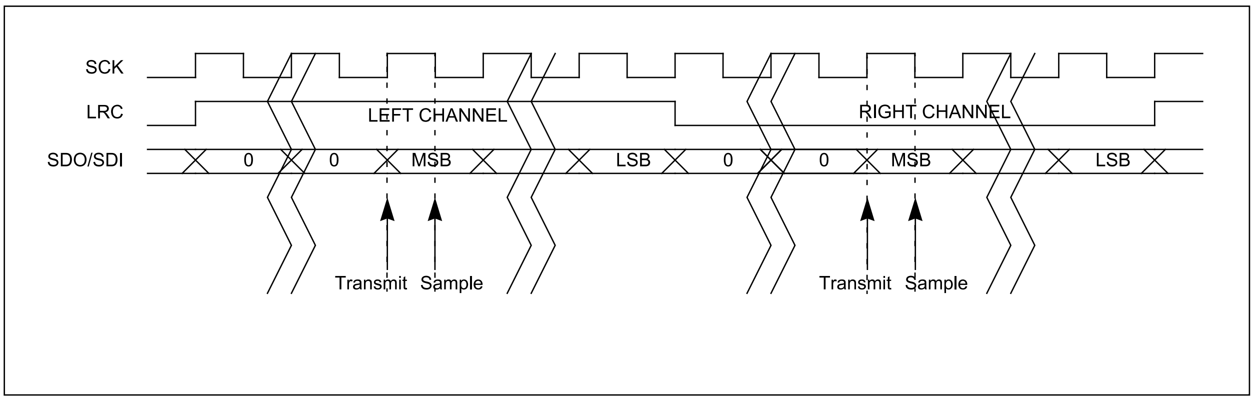


Figure 51-18. PCM/DSP with 16/20/24-bit Data and 32-bit Channel



51.5.15.6. Mono/Stereo Audio Data (AUDMONO)

The Audio Protocol function can transmit mono audio data on both the left and right channels. When AUDMONO = 1, and in a I^{2}S mode the register uses each FIFO location twice. This gives each channel the same mono stream of audio data. When AUDMONO = 0, the register uses each FIFO location once. This gives each channel a unique stream of data for stereo audio.

Receive data is not affected by AUDMONO.

51.5.15.7. Streaming Data Support

Due to the nature of audio data, it is often more important to keep the data channel actively transmitting or receiving to guarantee 100% time accurate data. This is especially true for streaming audio feeds that may be bursty or have packet drop. When this is the case the SPIxTXB may underrun frequently causing software to get involved with fixing the underrun.

51.5.15.7.1. Ignore Transmit Underrun (IGNTUR)

For cases when software does not care or need to know about the underrun condition, IGNTUR = 1 provides the serial engine the ability to ignore the underrun. When an underrun does occur, the SPI still sets the SPIxSTAT.SPITUR flag and obeys SPITUREN. Once SPITUR is set, it remains so until software clears it or SPIxCTRL_*.ENABLE = 0.

During the underrun condition, the SPI logic loads the SPIxSR with zeros instead of data from the SPIxTXB and continues to transmit. This allows gaps in audio data to sound like white space. Logic samples the underrun condition on channel boundaries, so transmission of zero data can start with either the left or right audio channel. When the condition clears (i.e., SPIxTXB is not empty), logic loads the audio data from the transmit buffer into the SPIxSR on the next LRC frame boundary.

Since recovery from the underrun condition occurs on the LRC frame boundary, software must make sure that the left and right audio data is always transferred to the FIFO in pairs. If the FIFO underruns between the left and right channel, left and right data could be swapped when the underrun condition clears. Generally, this should not be a problem since this feature is not meant to mask system performance problems but assist in recovery of streaming data packet delays.

51.5.15.7.2. Ignore Receive Overflow

Ignoring receive overflow is seldom beneficial, since it usually means there is a general performance problem in the system that software must handle properly. However, the SPI does implement the IGNROV bit.

Alternately, receive overflow can be prevented by using DISSDI when the system does not need to receive audio data. DISSDI can be changed on-the-fly and the receive register starts a receive on the leading LRC edge.

51.5.15.8. Host Mode Clocking

The SPI uses GCLK to generate SCK and LRC.

The use of an on chip ClockOut source. This clock source can also be driven out a pin to be used as MCLK by the CODEC.

Typically the ClockOut feature supports the System Clock and, if available, the USB Clock as a source. The ClockOut peripheral divides the source clock to a frequency that can be driven out of the chip.

51.5.15.8.1. SCK and LRC Clock Generation

MODE[32,16] or AUDWD_MODE[1:0] defines the relationship between LRC and SCK. When a frame is 64-bits SCK is 64x the frequency of LRC. When a frame is 32-bits SCK is 32x the frequency of LRC.

Since LRC toggles at the sample rate (Fs), SCK's frequency must be derived from it. To setup SPIxBRG, divide the desired sample rate by the GCLK or MCLK frequency (whichever is being used). Then, divide the resulting number by the frame size (either 32 or 64). Program this value into SPIxBRG. If a whole number is not the result, error will be present in your actual sample rate.

51.5.15.8.2. MCLK Support

The use of a RefOut (reference clock output) peripheral to generate MCLK for the CODEC is not a perfect choice. Driving a clock out an I/O Pad induces jitter that may degrade audio fidelity of the CODEC. The best solution is for the CODEC to use a crystal and be the host I^2S/Audio device.

In lieu of the CODEC generating MCLK (including SCK and LRC), the next best choice is for the RefOut peripheral to generate MCLK with the CODEC being the Host I^2S. In this configuration the CODEC can use any of its timing dividers to achieve the necessary clocking results for SCK and LRC.

Also, RefOut peripherals generally support 12MHz and 24MHz derived from a USB PLL. Typical audio MCLK frequencies of 12.288MHz and 11.2896MHz cannot be obtained using USB clock frequencies.

Figure 51-19. Device Level Clock Out Usage Example 1

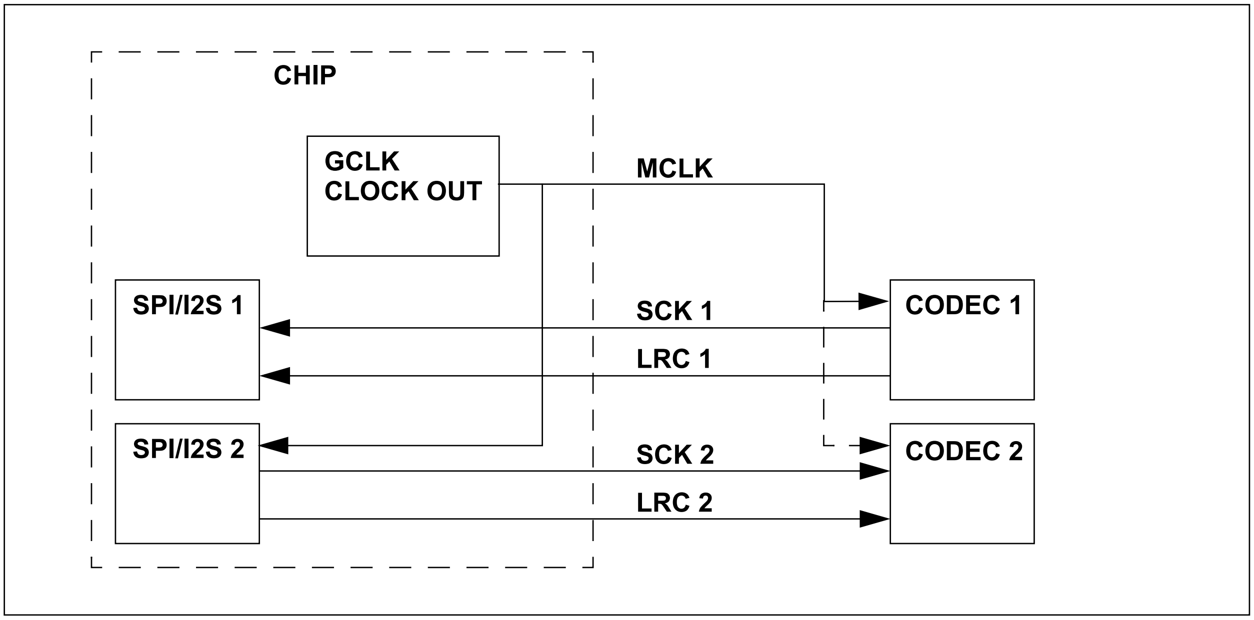
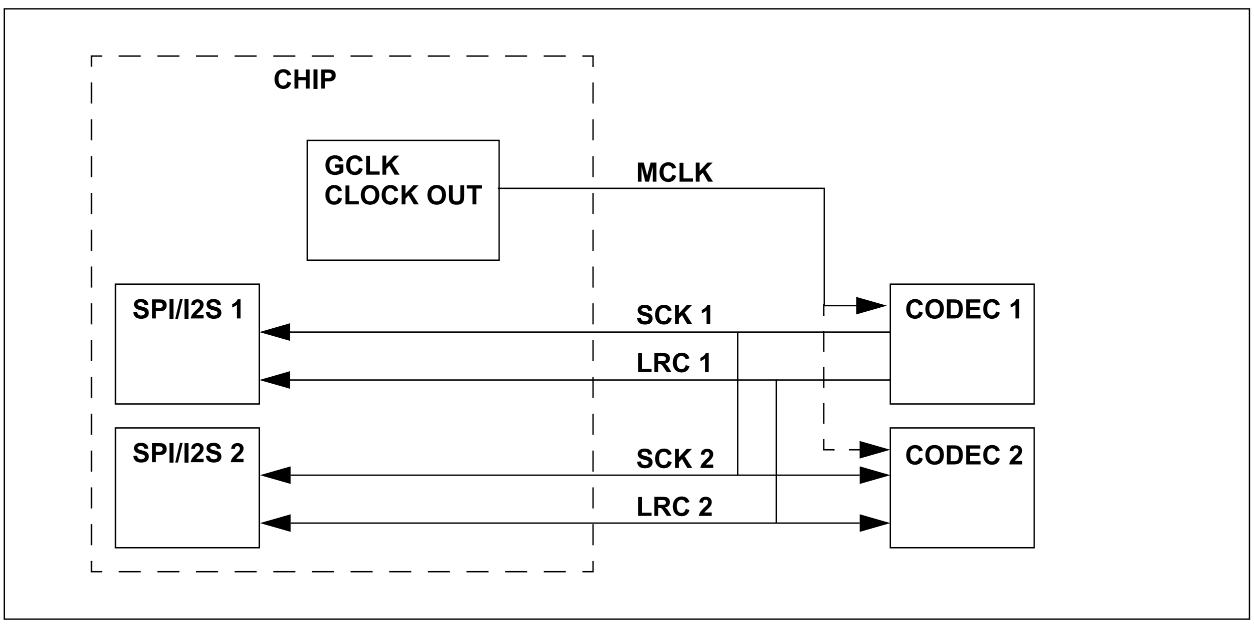


Figure 51-20. Device Level Clock Out Usage Example 2

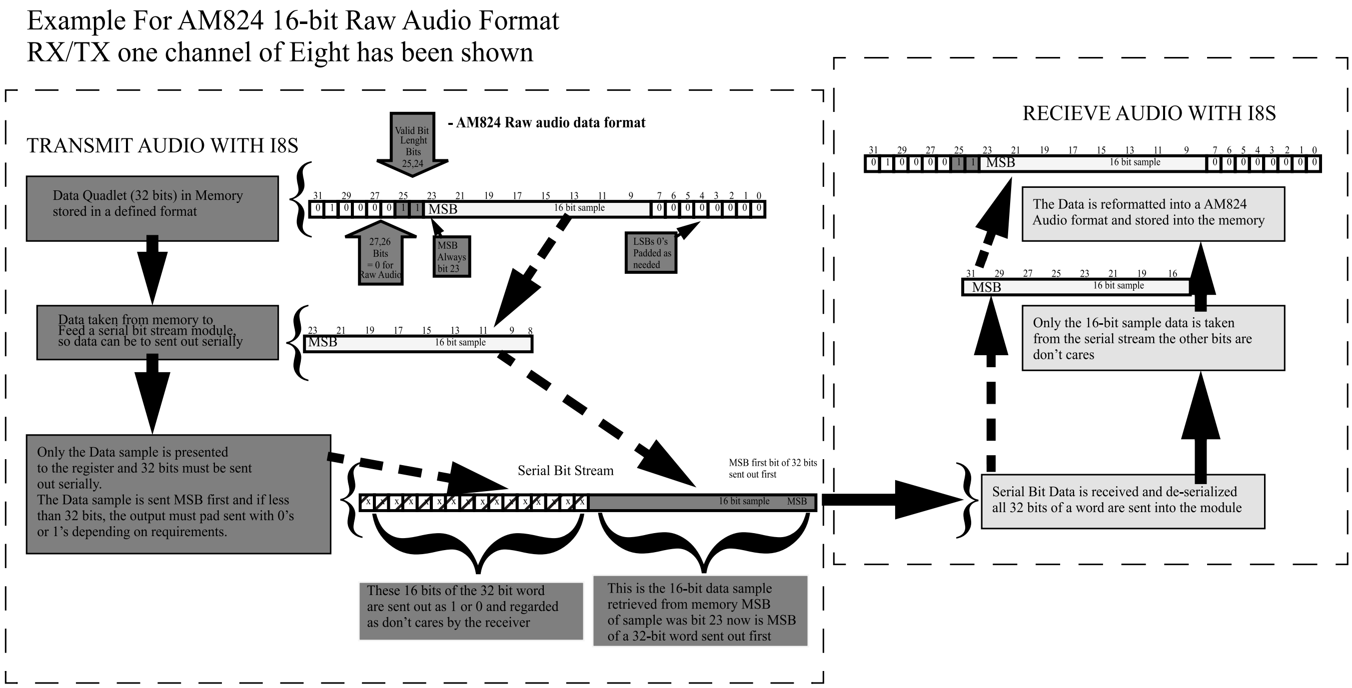


51.5.16. Audio Protocol I^8S (I^2S) (I2STPD) Operation New Features

The register SPIxCTRL_AUD contains the audio specific protocol control bits. When AUDEN = 1 the peripheral operates like an audio CODEC host or client. The audio protocol requires certain features of the SPI protocol and therefore overrides some SPI settings.

The peripheral uses the serial audio I^8S protocol defined herein. In each of the modes available for the I^8S the serial clock is free running and audio data is always transferred. For an example of AM824 data flow see the following figure.

Figure 51-21. Example of AM824 Data Flow Transmit/Receive



Four pins make up the serial interface. However, each audio connection is only half-duplex so SDO exists only on the transmit side and SDI exists only on the receive side of the interface. The four pins are:

- SDI: Serial Data Input
- SDO: Serial Data Output
- SCK: Serial Clock
- LRC: Left/Right Clock (on $\overline{SS}$/FSYNC)

I^{8}S allows multiple channels of data to be transmitted on a single data line. The I^{8}S interface is similar to the 2 channel serial audio interface I^{2}S with the exception that more channels are transmitted within a sample frame or sample period. Basically defined as an 8 channel TDM interface (left or right justified) with a 50% duty cycle LRC clock.

As with the I^{2}S interface the I^{8}S interface is comprised of two control clocks, a frame synchronization pulse (LRC), a serial clock (SCK), and the serial audio data line (SDO/SDI). Several Audio Modes can be selected, standard I^{8}S, right justified, left justified, AM824 24, 20,16-bit (slot) Raw Audio, 32 bit data, 16-bit x2 packed, 24-bit x 4bit packed, 16, 20, 24-bit MSB aligned with lower bits muted (ie filled with '1 or '0 determined by user with DATFILL bit), and Host with multiple client transmit operation.

Each channel block is comprised of the audio data word (32, 24, 20, or 16). When specified the remainder of the 32bit word for 24, 20, 16 is padded with zeros or don't cares depending on the type of Audio mode selected. In all modes the audio word is transmitted with the MSB first 2's compliment format and the word size of 32 bits.

The function of the FSYNC pulse (LRC) is to identify the beginning of the frame and is indicated by the rising edge of the pulse and the frame rate is at the audio sample rate such as 48Khz. The FSYNC pulse (LRC) has two separate required representations:

1. For the width to be equivalent to a channel block or,
2. Where the width is equivalent to a single period of the serial clock which is more common with a TDM interface.

In TPD mode is recommended to not enable the devices dedicated as clients interrupts. The SPITUREN, SPIROUEN, and FRMERREN for the Clients should all be disabled and only enable the Host's interrupts as needed.

51.5.16.1. Channel Block Alignment with the FSYNC (LRC) Pulse

There are two options for the alignment of the first channel block with the rising or falling edge of the FSYNC (LRC): 1. the beginning of the channel block aligns with the rising edge of FSYNC (LRC) or 2. the beginning of the channel block aligns with the falling edge of FSYNC (LRC).

The purpose of the SCK is to send the audio data into and out of the serial audio ports. The frequency of the SCK is directly proportional to the system audio sample rate, the number of channel blocks in a frame and the bit-width of each channel block.

SCK operates at a maximum of 256*Auto sample rate. An Example with the sample rate at 48Khz:

- 8 channel frame with 32 bit channel blocks at 48Khz requires a 12.2880 Mhz SCK

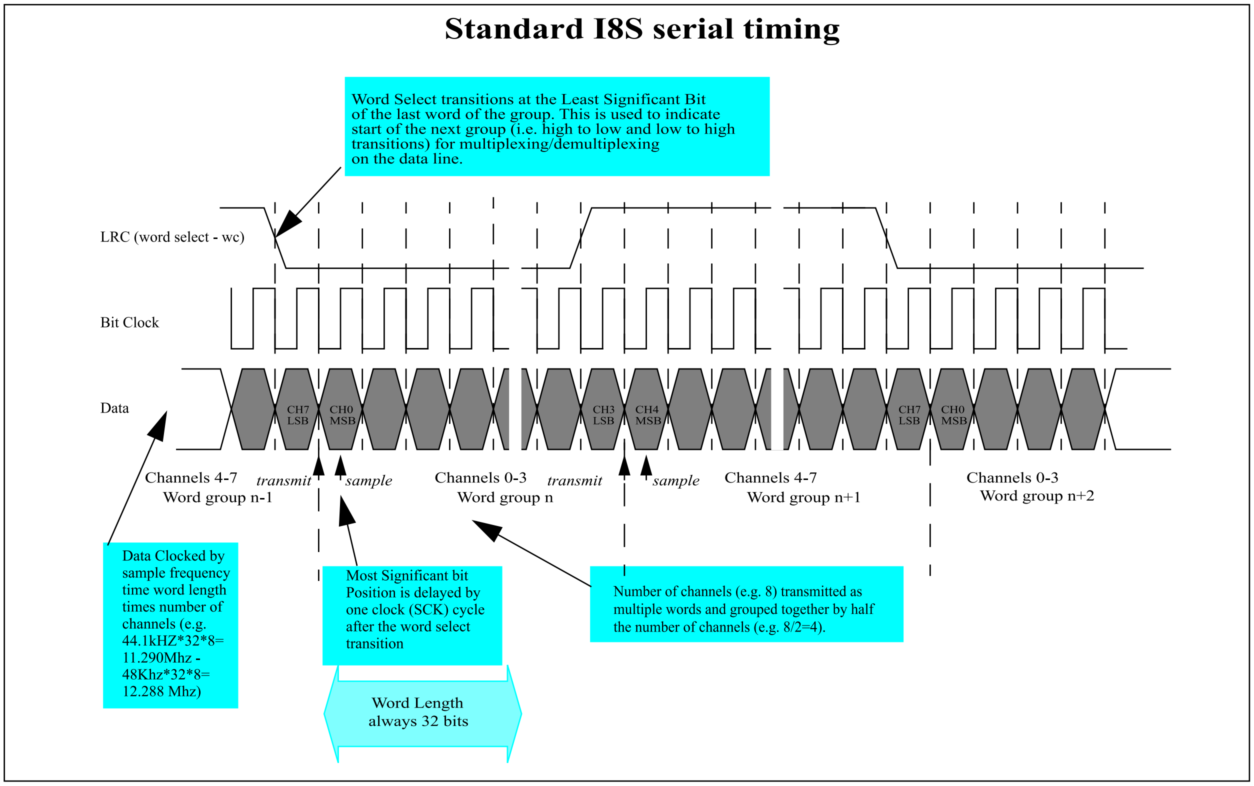
51.5.16.2. I^{8}S (I^{2}S) Standard Mode AM824 24-bit Raw Audio Format Mode (16, 20 24 Sample Data) AUDMOD=00, AUDFMT[2:0]=101(001), AUDWD_MODE[1:0]=10

In I^{8}S (I^{2}S) AM824 mode, A frame transmits left channels first then right channels. For the left channels, the data is transmitted while LRC is low and for the right channels, the data is transmitted while LRC is high. The transmitter drives the audio data's MSB on the first falling edge of SCK after an LRC transition. The receiver samples the MSB on the second rising edge of SCK.

For I8S standard mode AM824 24-bit raw audio for- mat, the configuration bits in SPIxCTRL_* must be set as follows: AUDMOD=00, AUDEN=1, FRMPOL=0, CPOL=1, CPHA=1, FRMSYPW=0100, FRMCNT=011, AUDFMT[2:0]=101, AUDWD_-

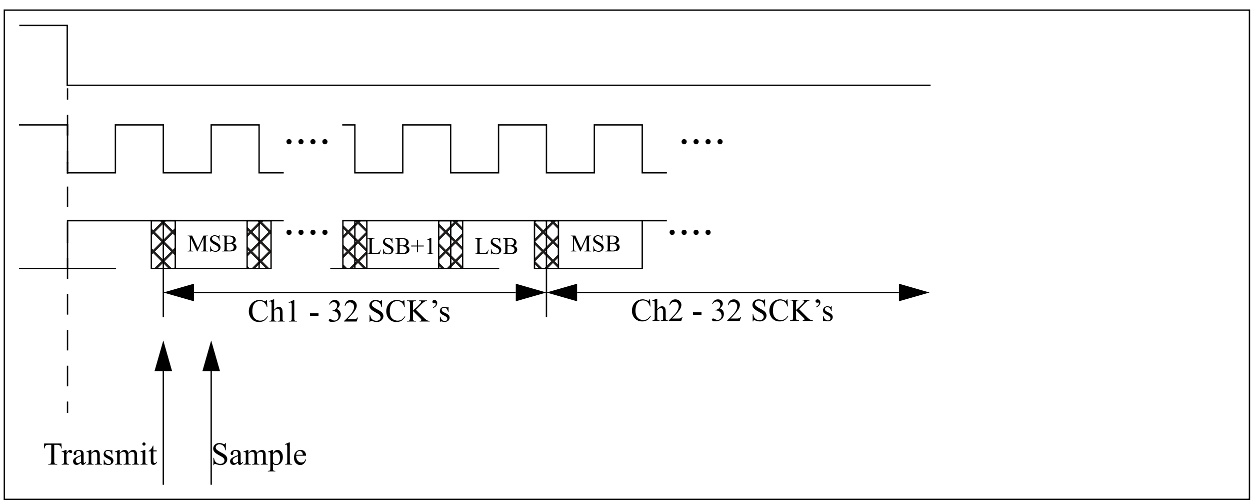
MODE[1:0]=10, FRMCOINC=0. These values set SDO and LRC transitions to occur on the falling edge of SCK and sampling of SDI to occur on the rising edge of SCK. It also starts a frame with LRC falling edge transition. The following figure shows the waveform for this 24 bit configuration with relationship of the LRC with Falling Edge of LRC with transmit on the next falling edge of CLK and sampling on the rising edge of CLK FRMPOL=0, CPOL=1, CPHA=1, FRMCOINC=0.

Figure 51-22. I[8]S Standard Timing with Falling Edge Transition of LRC



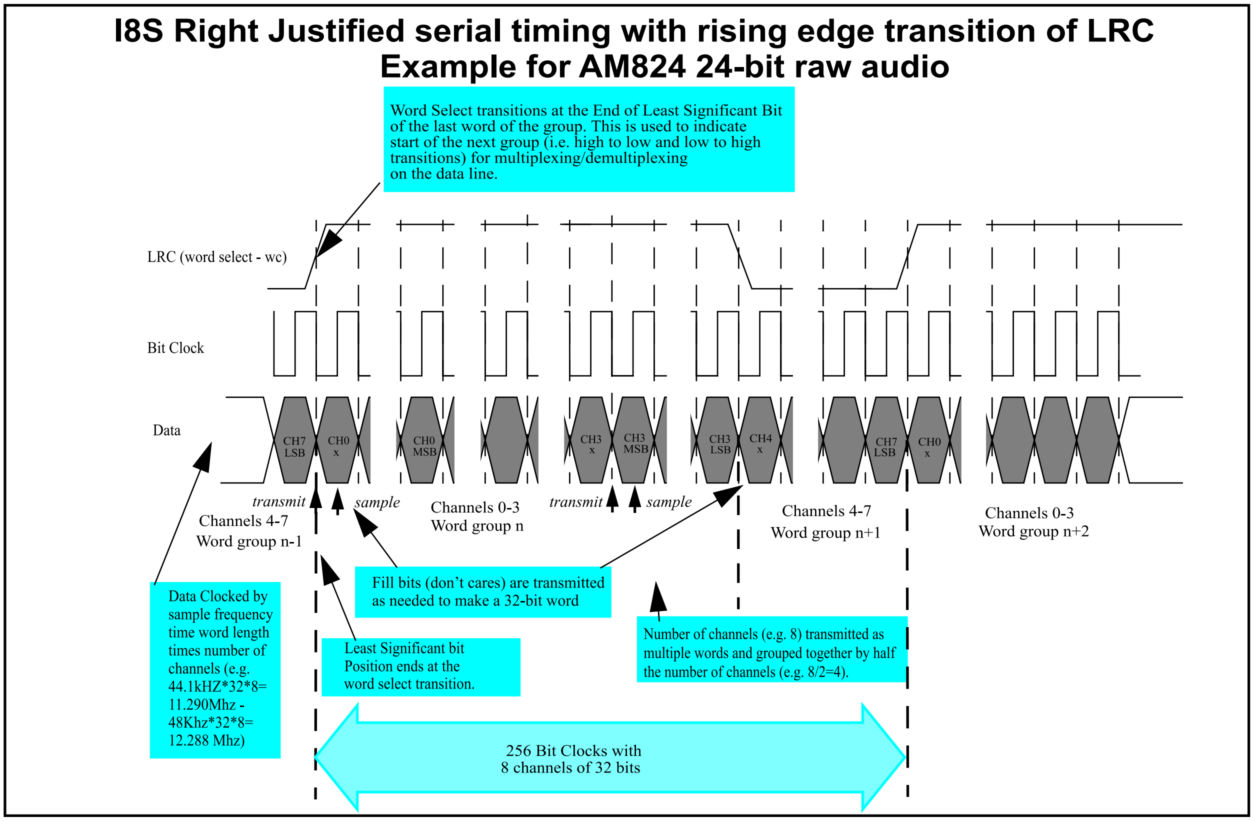
The following figure shows an expanded view of this waveform with, starting of a frame on the falling edge transition of LRC. AM824 formats with 20, 16 bit raw audio are similar and are specified using the AUDWD_MODE[1:0] register set to 01 and 00.

Figure 51-23. Expanded I8S Standard Timing with Falling Edge Transition of LRC



The following figure shows relationship of the LRC with rising edge of CLK with transmit on falling edge of CLK and sampling on the rising edge of CLK FRMPOL=1, CPOL=0, CPE=1,FRMCOINC=1.

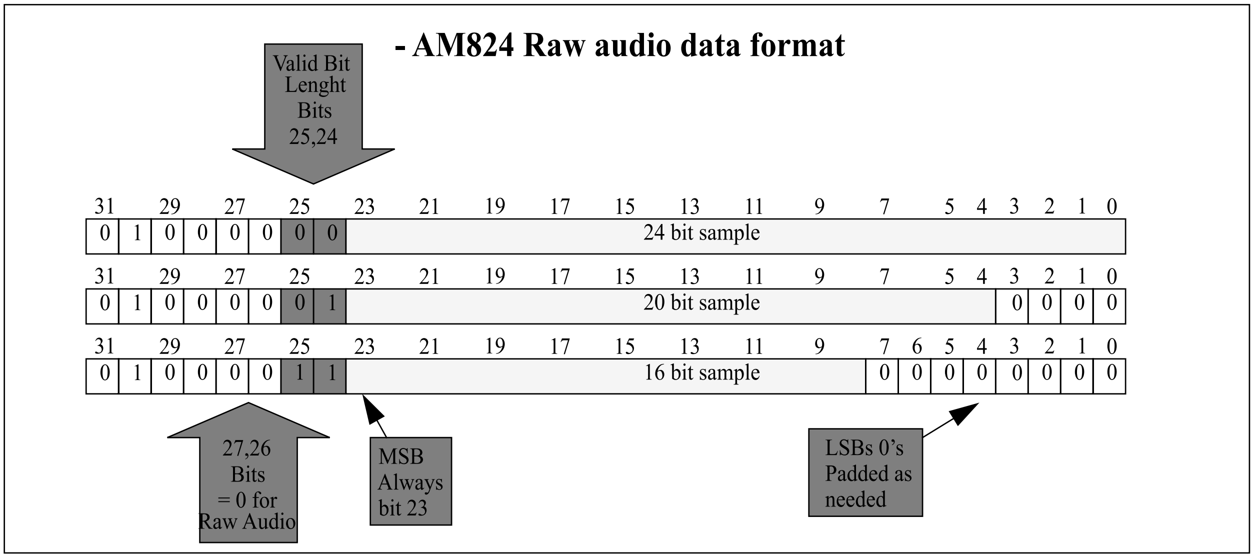
Figure 51-24. I8S Standard Timing with Rising Edge Transition of LRC and Sample Data on Falling Edge of CLK AM824 - Raw Audio Format



An I²S example for AM824 would be the same as the I⁸S in the previous figure. With the differences of 2 channels in-place of 8 channels and the frame would be 64 bit clocks in stead of 256 bit clocks. For I²S standard mode AM824 24-bit raw audio format, the configuration bits in SPIxCTRL_* must be set as fol- lows: AUDMOD=00, AUDEN=1, FRMPOL=0, CPOL=1, CPHA=1, FRMSYPW=0001, FRMCNT=001, AUDFMT[2:0]=001, AUDWD_MODE[1:0]=10, FRMCOINC=0.

As the data is serially sent in/out of the peripheral only the 24(20/16) bit sample is sent and the remaining lower bits not defined by the 16, 20 or 24 bit sample (total of 32 bits in a quadlet/word) are don't care values sent out. The don't care bits are sent out to make a complete 32-bit word per channel to keep channel alignment with the LRC. As the serial data is sent into the peripheral, the 32-bit word is reformed with the proper label for the AM824 format and 24(20/16) bit sample to form a AM824 24, 20, 16 bit sample raw for- mat to store in memory as shown in the following figure.

Figure 51-25. AM824 Packet Format

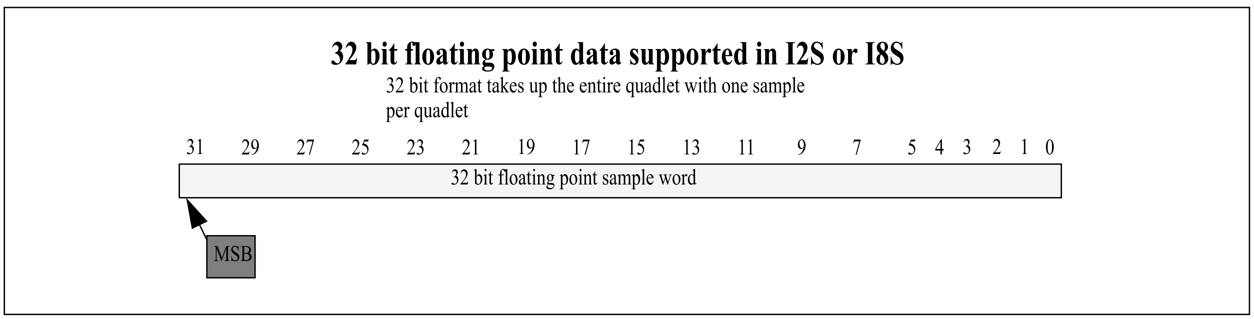


51.5.16.3. I8S (I2S) - 32-BIT FLOATING POINT AUDMOD = 00, AUDFMT[2:0] = 110, AUDWD_MODE[1:0] = 01

For I8S standard mode, AM other format mode 32-bit Floating Point format, the configuration bits in SPIxCTRL_* must be set as follows: AUDMOD=00, AUDEN=1, FRMPOL=0, CPOL=1, CPHA = 1, FRMSYPW=0100, FRMCNT=011, AUDFMT[2:0]=110, AUDWD_MODE[1:0]=01, FRMCOINC=0. These values set SDO and LRC transitions to occur on the falling edge of SCK and sampling of SDI to occur on the rising edge of SCK. The following figure shows the format of the 32-bit word to store in memory.

I2S is supported as part of the legacy mode of operation and is not described here.

Figure 51-26. 32-bit Packet Format



51.5.16.4. I8S (I2S)24-BIT X 4 AUDIO PACK - AUDMOD=00, AUDFMT[2:0]=110, AUDWD_MODE[1:0]=00

For I8S (I2S) standard mode, AM other format mode 24-bit x 4 audio pack format, the configuration bits in SPIxCTRL_* must be set as follows: AUDMOD=00, AUDEN=1, FRMPOL=0, CPOL=1, CPHA=1, FRMSYPW=0100, FRMCNT=011, AUDFMT[2:0]=110, AUDWD_MODE[1:0]=00, FRMCOINC=0. These values set SDO and LRC transitions to occur on the falling edge of SCK and sampling of SDI to occur on the rising edge of SCK.

As the data is serially sent in/out of the peripheral, only the 24 bit sample is captured or sent MSB first while the remaining lower bits of the 32 bit quadlet are don't cares values. And the don't care bits are also sent out to complete a 32-bit word per channel.

An Example of 24-bit x 4 audio data flow see the following figure. When the serial data is sent into the module, the 24-bit data sample are stored in a 32-bit word reformed/packed to store in memory.

An I^{2}S example for 24-bit x 4 audio packed format would be the same as the I^{8}S above with the differences of 2 channels in-place of 8 channels and the frame would be 64 bit clocks in stead of 256 bit clocks and the configuration is as follows:

AUDMOD=00, AUDEN=1, FRMPOL=0, CPOL=1, CPHA=1, FRMSYPW=0001, FRMCNT=001, AUD-FMT[2:0]=010, AUDWD_MODE[1:0]=00, FRMCO- INC=0, DATFMT_LR=0.

Figure 51-27. Example of 24-bit x 4 Data Flow Transmit/Receiver

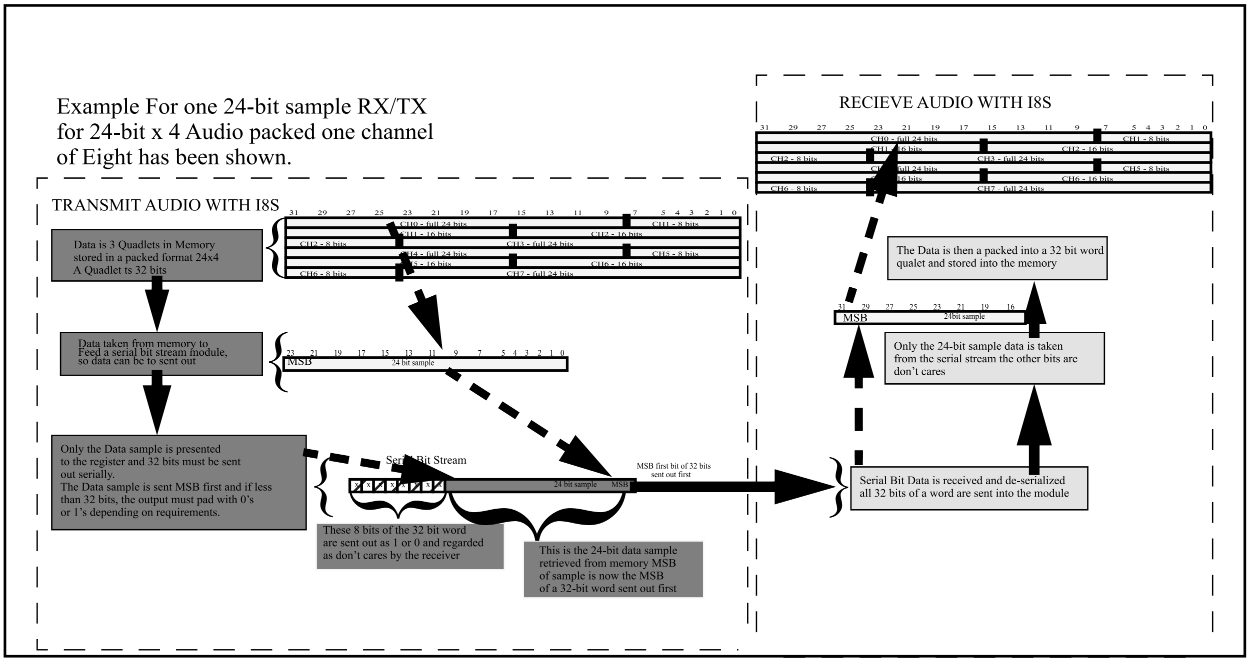


Figure 51-28. I^{8}S - 24-bit x 4 Audio Packed Format

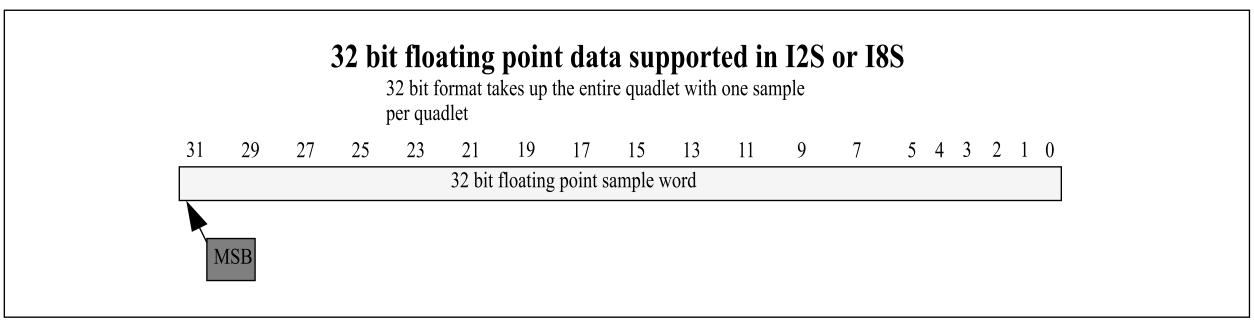
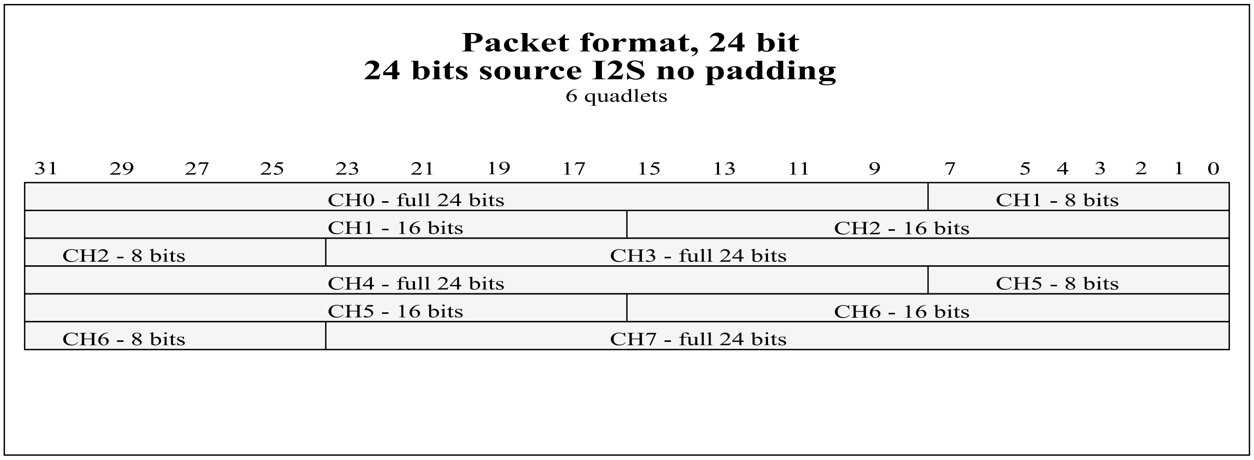


Figure 51-29. I²S - 24-bit x 4 Audio Packed Format



51.5.16.5. 16-BIT X 2 AUDIO PACK - AUDMOD=00, AUDFMT[2:0]=110(010), AUDWD_MODE[1:0]=10 OR 11, DATFMT_LR = 1 OR 0.

For I⁸S (I²S) standard mode, AM other formats mode 16-bit x 2 audio pack format, the configuration bits in SPIxCTRL_* must be set as follows: AUDMOD=00, AUDEN=1, FRMPOL=0, CPOL=1, CPHA=1, FRMSYPW=0100, FRMCNT=011, AUDFMT[2:0]=110, AUDWD_MODE[1:0]=10, FRMCOINC=0, DATFMT_LR=0. These values set SDO and LRC transitions to occur on the falling edge of SCK and sampling of SDI to occur on the rising edge of SCK. The following figure shows the format to store in memory.

As the data is serially sent in/out of the peripheral only the 16 bit sample is sent MSB first and the remaining lower bits not defined by the 16 bit sample (total of 32 bits in a quadlet/word) are don’t cares values but the bits are sent out to complete a 32-bit word per channel.

For an Example of 16-bit x 2 audio data flow see the following figure.

When the serial data is sent into the peripheral, the 32-bit word is reformed/packed to store in memory as shown in the following figures.

An I²S example for 16-bit x 2 audio packed format would be the same as the I⁸S above with the differences of 2 channels in-place of 8 channels and the frame would be 64 bit clocks instead of 256 bit clocks and the configuration is as follows: AUDMOD=00, AUDEN=1, FRMPOL=0, CPOL=1, CPHA=1, FRMSYPW=0001, FRMCNT=001, AUDFMT[2:0]=010, AUDWD_MODE[1:0]=10, FRMCOINC=0, DATFMT_LR=0.

Figure 51-30. Example of 16-bit x 2 Data Flow Transmit/Receiver

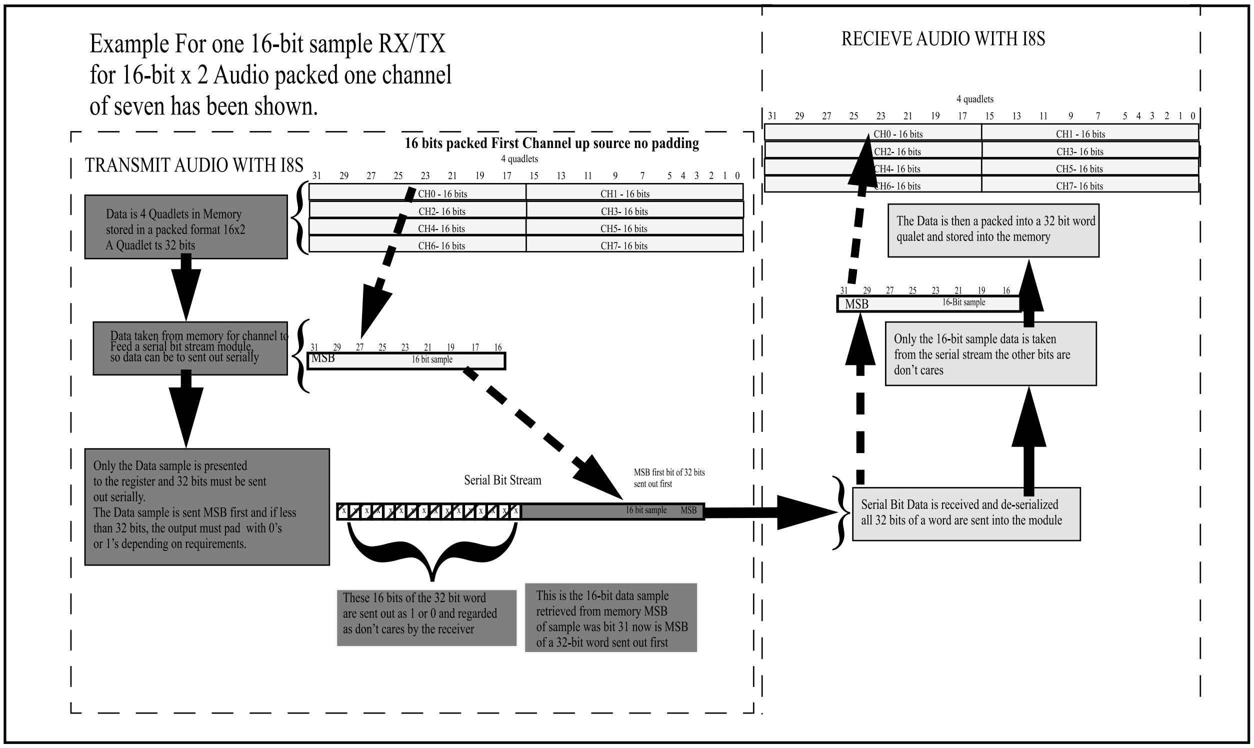


Figure 51-31. 16-bit x 2 Audio Packed Format AUDWD_MODE 10 Pack Left Up

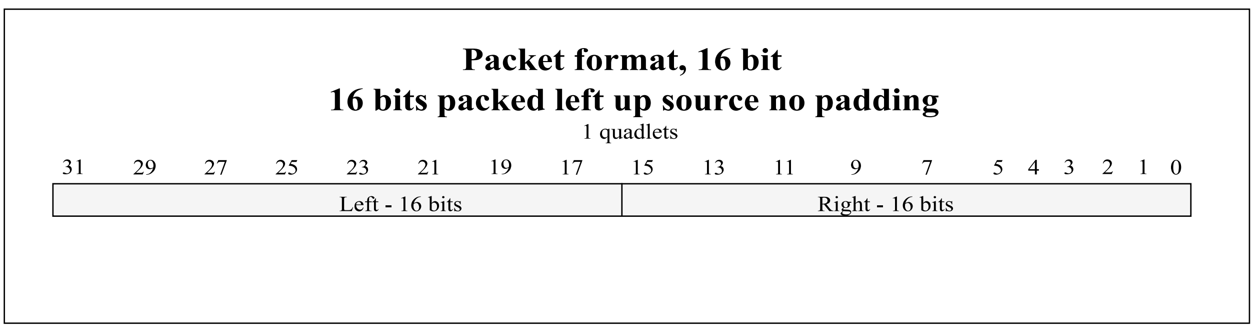
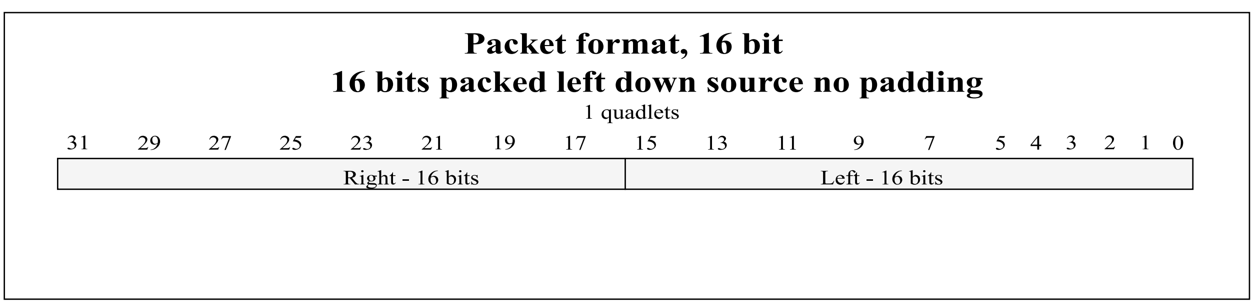


Figure 51-32. 16-bit x 2 Audio Packed Format Mode 11 pack Left Down



51.5.16.6. 24, 20, 16-BIT MSB ALIGNED - OTHER FORMATS

I^{8}S 24-bit MSB aligned format, the configuration bits in SPIxCTRL_* must be set as follows: AUDMOD=00, AUDEN=1, FRMPOL=0, CPOL=1, CPHA=1, FRMSYPW=0100, FRMCNT[2:0]=011, AUDFMT[2:0]=100, AUDWD_MODE[1:0]=10, FRMCOINC=0. These values set SDO and LRC transitions to occur on the falling edge of SCK and sampling of SDI to occur on the rising edge of SCK.

As the data is serially sent in/out of the peripheral only the 24bit sample with padding is sent MSB first (total of 32 bits in a quadlet/word). When the serial data is sent into the peripheral, the 32-bit word is stored in memory as shown in the following figures. When set to transmit (DISSDO = 0), this device drives the unused bit slots (preceding the audio data) with logic level 0. When set to receive (DISSDI = 0), this device ignores the unused bit slot I^{2}S is supported as part of the legacy mode of operation and is not described here.

For an Example of 24-bit MSB Aligned audio data flow see the following figure.

Figure 51-33. Example of 24-bit MSAB Aligned Data Flow Transmit/Receive

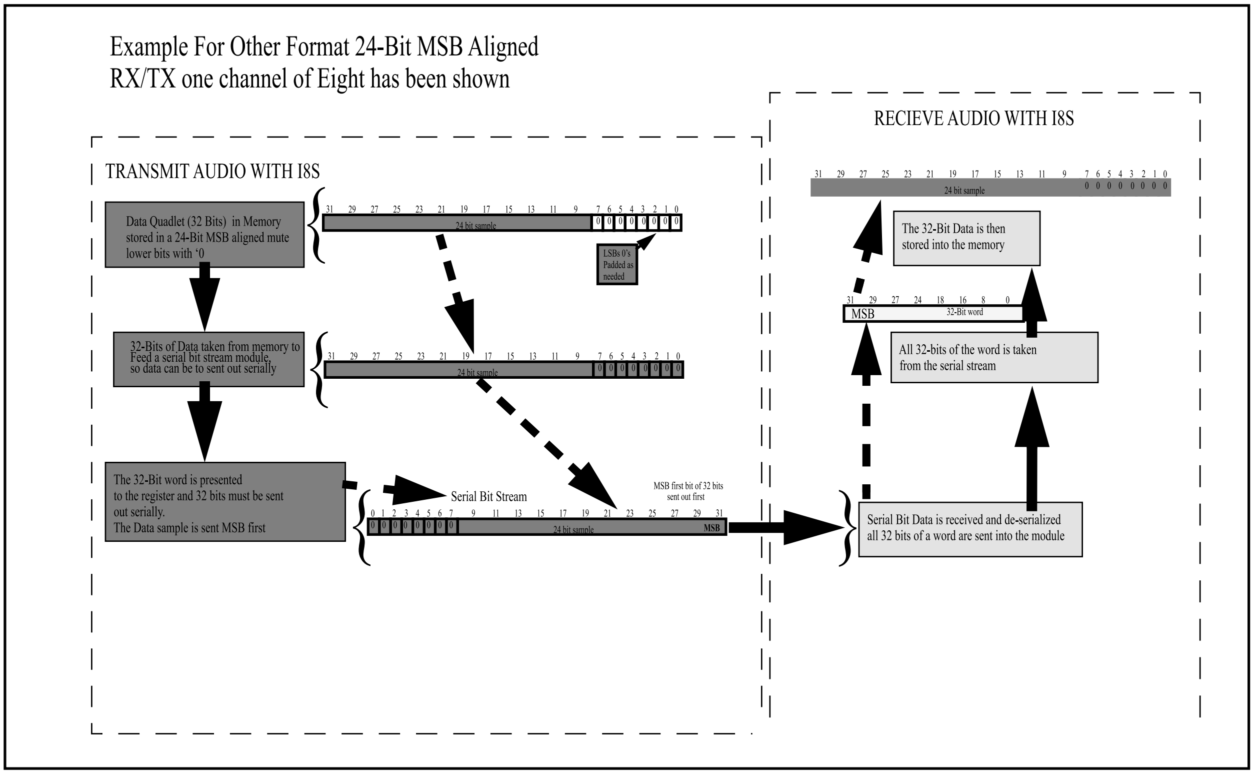
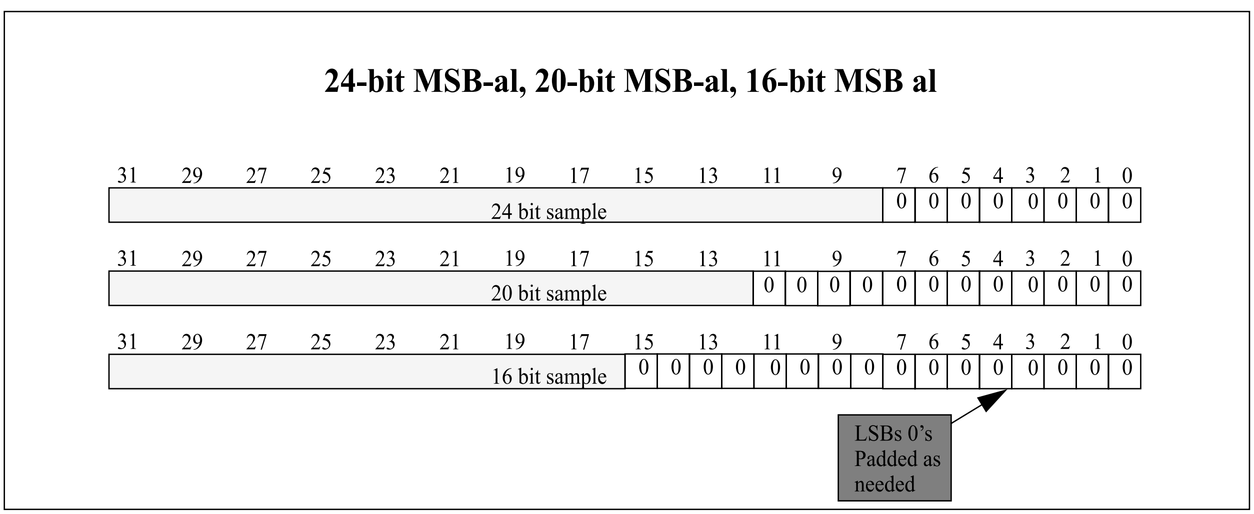


Figure 51-34. Other Formats 16,20,24 MSB Aligned Mute with Zeros

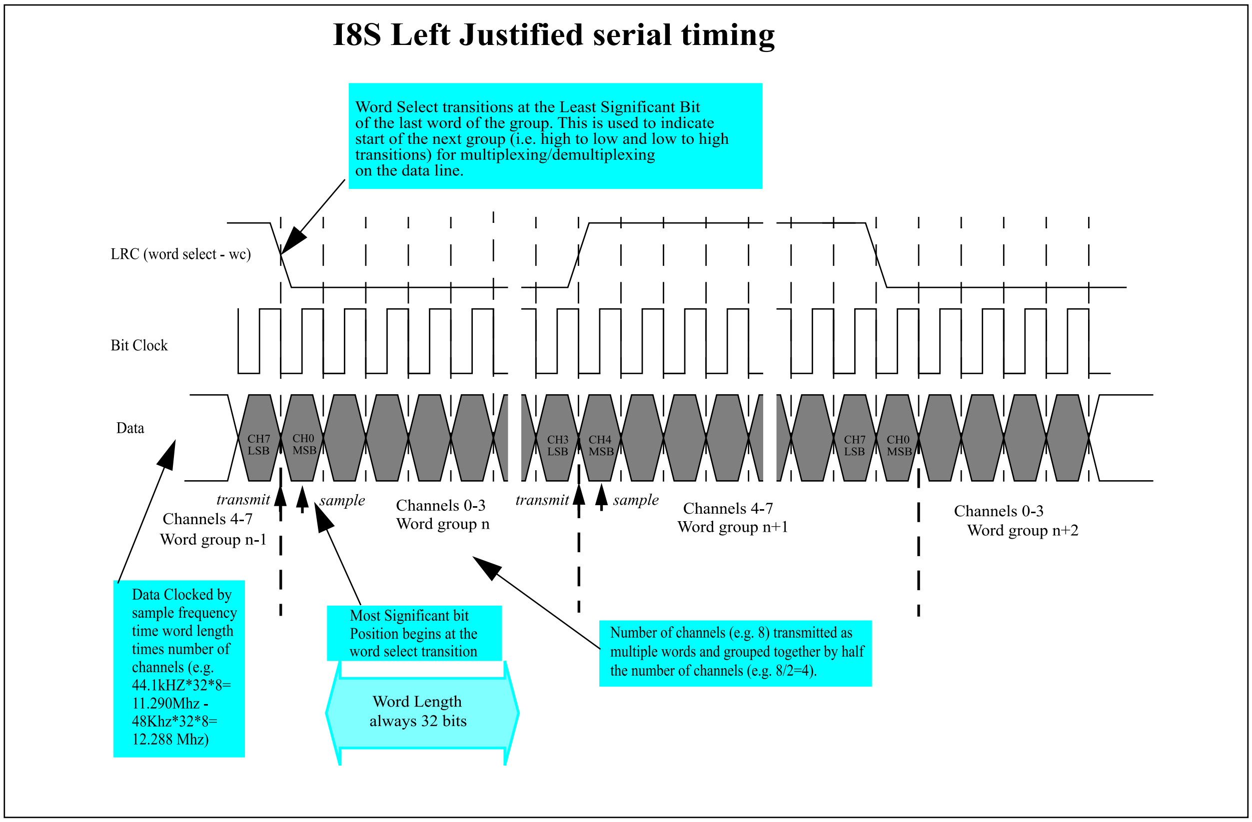


51.5.16.7. I⁸S LEFT JUSTIFIED MODE -AUDMOD=01 AUDFMT = 001, 010, 101, 110, 100

For example, I⁸S Left Justified mode, the transmitter drives the audio data's MSB on the SCK edge that is coincident with an LRC transition. The receiver samples the MSB on the next SCK edge. Left justified can be used for all I⁸S formats with AUDWD_MODE[1:0]= 00,01, 10 and AUDFMT[2:0] = 001, 010, 101, 110, 100 settings. To configure for the I⁸S left justified standard convention, AM824 24-bit raw data, set the following bits in SPIxCTRL_* as follows: AUDMOD=01, AUDEN=1, FRMPOL=0, CPOL=1, CPHA = 1, FRMSYPW=0100, FRMCNT=011, AUDFMT[2:0]=101, AUDWD_MODE[1:0]=10, FRMCOINC=1. When set to transmit. The following figure shows the waveform for this configuration, with starting of a frame the falling edge transition of LRC.

An I²S example for AM824 24-bit raw data format would be the same as the I⁸S above with the differences of 2 channels in-place of 8 channels and the frame would be 64 bit clocks in stead of 256 bit clocks. To configure I²S set the following: AUDMOD=01, AUDEN=1, FRMPOL=0, CPOL=1, CPHA=1, FRMSYPW=0001, FRMCNT=001, AUDFMT[2:0]=001, AUDWD_MODE[1:0]=10, FRMCOINC=1.

Figure 51-35. I⁸S Left Justified Mode



51.5.16.8. I⁸S RIGHT JUSTIFIED MODE AUDMOD=10 AUDFMT = 001, 010,101, 110, 100

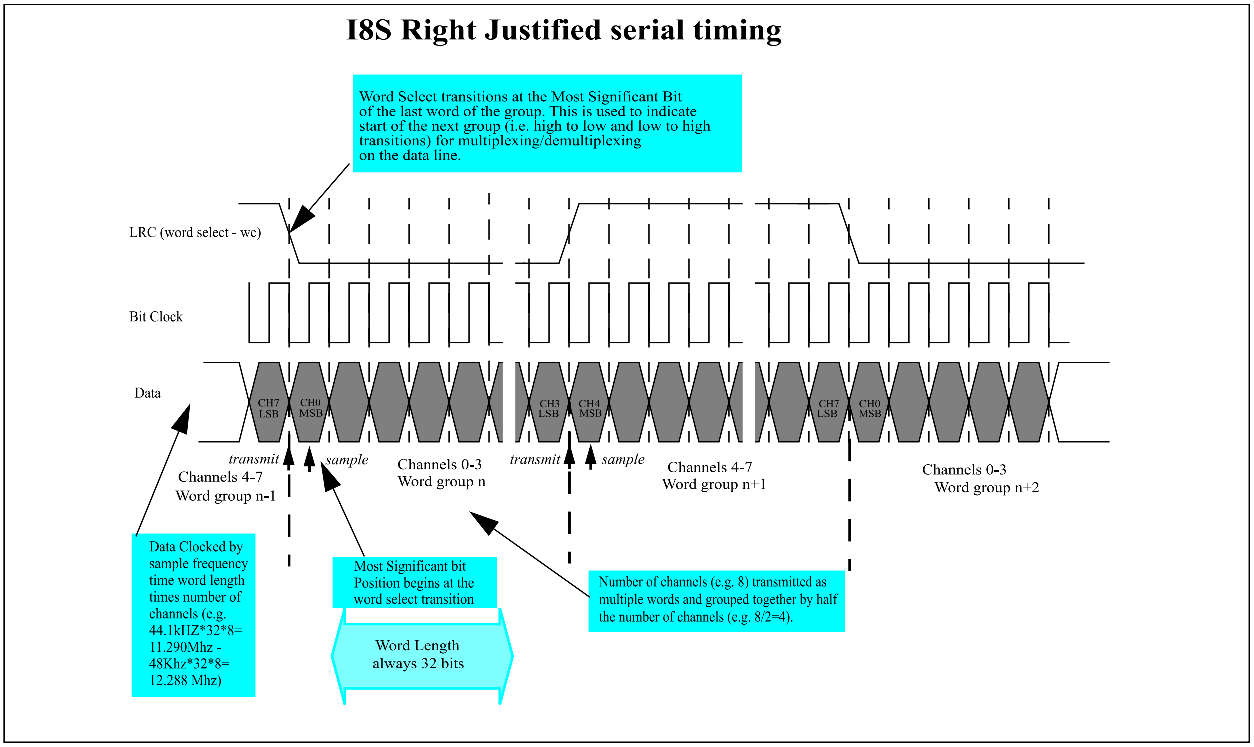
In Right Justified mode, the transmitter drives the audio data's MSB on the SCK edge that is coincident with an LRC transition. The receiver samples the MSB on the next SCK edge. Right justified can be used for all I⁸S formats with AUDWD_MODE[1:0] 00, 01, 10 and AUDFMT[2:0] = 001, 010, 101, 110, 100 settings.

To configure for the I⁸S right justified standard convention, AM824 24-bit raw data, set the following bits in SPIxCTRL_* as follows: AUDMOD=10, AUDEN=1, FRMPOL=0, CPOL=1, CPHA = 1, FRMSYPW=0100, FRMCNT=011, AUDFMT[2:0] = 101, AUDWD_MODE[1:0] = 10, FRMCOINC=1. The following figure shows the waveform for this configuration.

An I²S example for AM824 24-bit raw data format would be the same as the I⁸S above with the differences of two channels in-place of 8 channels and the frame would be 64 bit clocks in stead of

256 bit clocks. For I^{2}S AM824 24-bit use this configuration: AUD MOD=10 AUDEN=1, FRMPOL=0, CPOL=1, CPHA=1, FRMSYPW=0001, FRMCNT=001, AUDFMT[2:0]=001, AUDWD_MODE[1:0]=10, FRMCO INC=1.

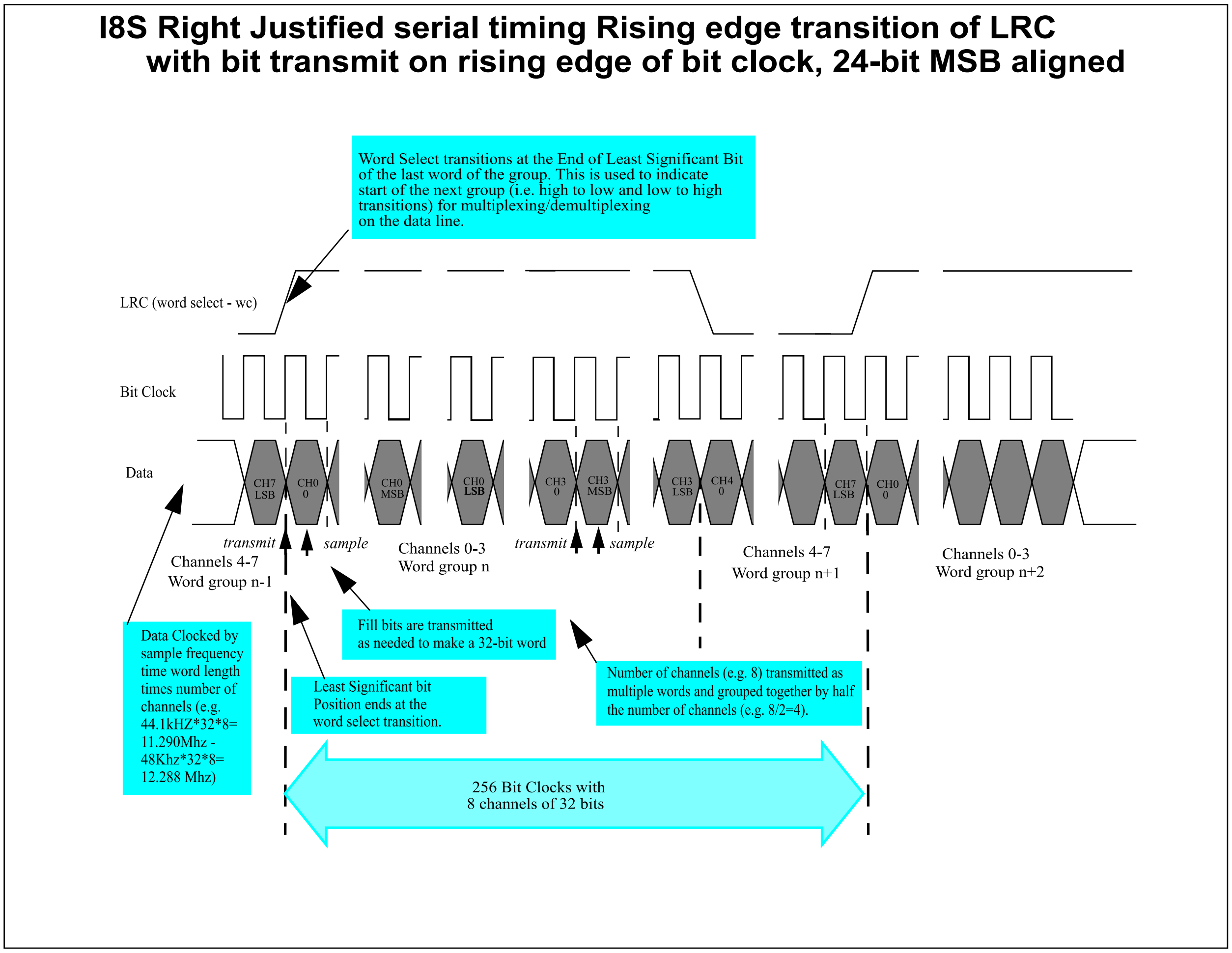
Figure 51-36. I^{8}S Right Justified Mode



To configure for the I^{8}S right justified standard convention, AM824 24-bit raw with starting of a frame with the rising edge transition of LRC data, set the following bits in SPIxCTRL_* as follows: AUDMOD=10, AUDEN=1, FRMPOL=1, CPOL=0, CPHA=1, FRM- SYPW=0100, FRMCNT=011, AUDFMT[2:0]=101, AUDWD_MODE[1:0]=10, FRMCOINC=1.

The following figure shows the waveform for this configuration.

Figure 51-37. I8S Right Justified Mode with Rising Edge Transition LRC



51.5.16.9. I2STPD Host with Multiple Client Transmit Operation (TPDFMT=100)

When in this mode (TPD_EN=1) all the configured SPI's FIFO(s) can be written by a single write to the HOST SPI and by utilizing the spi_ixs_buff_bus_in/out [31:0] bus to fill the client FIFOs.

In I2STPD Mode CLK and LRC system operation, the Client will feed through the CLK and LRC from the inputs (driven from device Host) to their respective Outputs. In addition the client will control the output enables as needed. This unique Microchip I2STPD mode all the SPI's in this configured in the device for this mode, look like Clients to a System. The device SPI Host drives its output CLK and the LRC to the device clients (pds_sck_in, pds_ssn_fsync_in driven from the device Host and not from ports).

The SPI's APBM interface needs to be configured in the I2STPD mode so that a single APB write will be utilized to fill all the configured SPI FIFO's at the same time.

51.5.16.9.1. I2STPD Standard Mode (TPDMOD=00, TPD_EN=1)

In I2STPD mode, the transmitter drives the audio data's MSB on the first falling edge of SCK after an LRC transition. The receiver samples the MSB on the second rising edge of SCK. The left channel data transmits while LRC is low and the right channel transmits while LRC is high. A frame transmits left channel first then right channel. In the I2STPD configuration all the SPI's (host and clients) must be configured with the same justification to ensure proper operation.

To be I^{2}S compliant, the configuration bits in SPIxCTRL_* registers for each SPI in the system must be set as follows: TPD_EN=1, TPDMOD=00, FRMPOL=0, CPOL=1, CPHA = 1, FRMSYPW=0001, FRMCNT=001, AUDFMT[2:0]=100, FRMCOINC=0.

MSTEN=as need per SPI case, SLV_MS_B_UPPR=as need per SPI case, MSB_SLV_EN[4:0]=as need per SPI case.

51.5.16.9.2. Left Justified (TPDMOD=01TPDFMT=100, DATFMT_LR=0)

In Left Justified mode, the transmitter drives the audio data's MSB on the SCK edge that is coincident with an LRC transition. The receiver samples the MSB on the next SCK edge. In the I2STPD configuration all the SPI's (host and clients) must be configured with the same justification to ensure proper operation.

51.5.16.9.3. Right Justified (TPDMOD=10 TPD_EN=1, DATFMT_LR=1)

In Right Justified mode, the transmitter drives the audio data's MSB on the SCK edge that is coincident with an LRC transition. In the I2STPD configuration all the SPI's (host and clients) must be configured with the same justification to ensure proper operation.

51.5.16.9.4. Host SCK and LRC Clock Generation

AUDWD_MODE[1,0] defines the relationship between LRC and SCK and rising/falling edge sampling. The frame is 64-bits SCK (frame pulse is 32-bits SCK) which is 64x the frequency of LRC.

Since LRC toggles at the sample rate (Fs), SCK's frequency must be derived from it. Only For the I2STPD Host SPI setup SPIxBRG, divide the desired sample rate by the GCLK or MCLK frequency (whichever is being used). Then, divide the resulting number by the frame size (64). Program this value into SPIxBRG. If a whole number is not the result, error will be present in your actual sample rate. The I2STPD clients are not generating the SCK or LRC but using them from the I2STPD Host SPI.

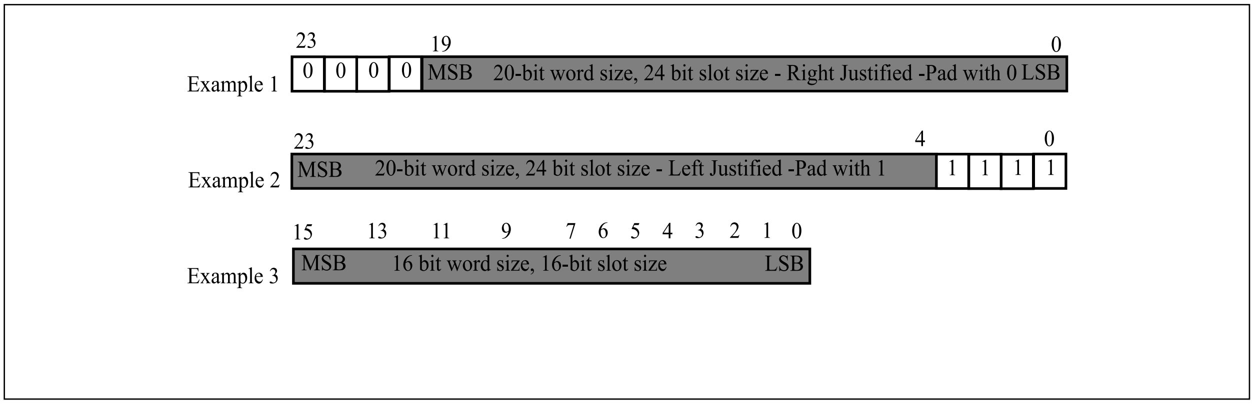
51.5.16.10. TDM (TDM_EN=1)

When configured for TDM protocol mode, TDM sup- ports a wide variety of serial data formats. The TDM format is typically used to transfer data to or from a DSP to the MCU. The TDM format consist of three components clock, data, and frame sync (I^2S could be considered a subset of TDM). The data bits are grouped into words and slots/channels as shown in the following figure. The data in the slot can be left or right justified with any unused bits filled with 1 or 0 as defined by AUDWD_MODE[1:0], when word size is < Slot size. The configured word must be "equal" or "less than" the slot/channel length size. A frame consist of multiple slots/channels and the TDM frame is defined by the frame sync pulse. The data transfer can be continuous and periodic because the TDM format is most commonly used to communicate at a fixed sample rate.

There are no delays between slots/channels, with the last byte of the slot/channel followed immediately on the next serial clock cycle with the first bit of the next slot/channel. But the frame sync may be offset from the first bit of the first slot/channel with 0 or 1-cycle delay.

TDM mode requires the transmitter and receiver in the system to mutually agree on the number of bits per slot/channel. This agreed to number of bits per slot/channel determines the slot/channel boundary which is not determined by the frame sync pulse signal. The frame sync pulse determines the beginning of the slot/channel 0 and the beginning of a new frame. Some examples of the word sizes and slot sizes that can be configured when in TDM mode are shown in the following figure.

Figure 51-38. Examples of Word and Slot Sizes for TDM Data



Example for a 6 channel TDM with 8 bit word and slot size =8 bits set up registers FRMCNT=0011, TDMSSZ=000, TDMWSZ=000, AVDFMT=011, FRMCNT[2:0]=110, FRMSYPW=0001, FRMSYPW=0000, CPOL=0, FRMCOINC=1, FRMPOL=1, TDM_EN=1.

The following figure shows an example configuration above 6channel TDM with 8 bit word size and 8 bit slot size. The following figures show the TDM format bit delays set by FRMCOINC (1= coincident, 0=precedes by one bit clk), relative from frame sync.

Figure 51-39. TDM Format 6-Slots with 0-bit Delay From Frame Sync

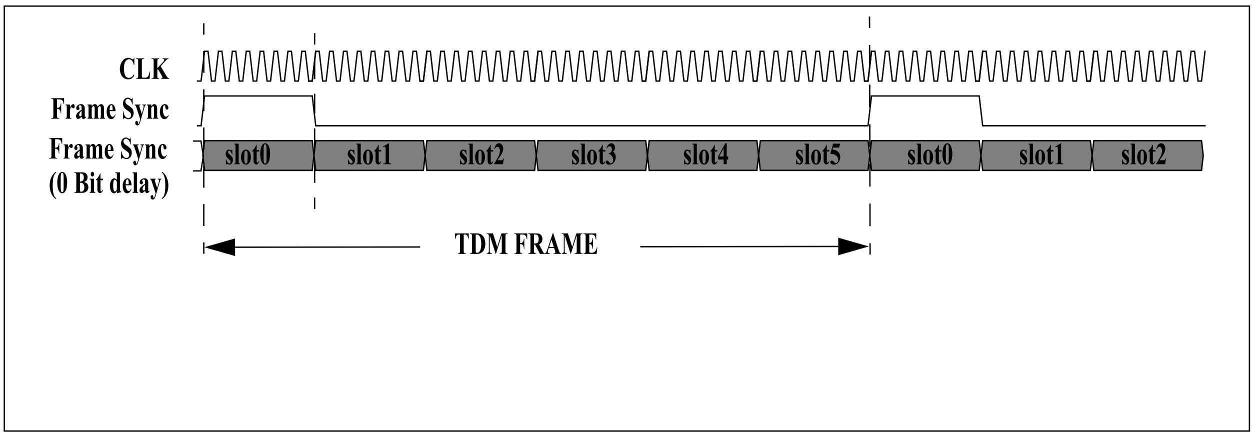
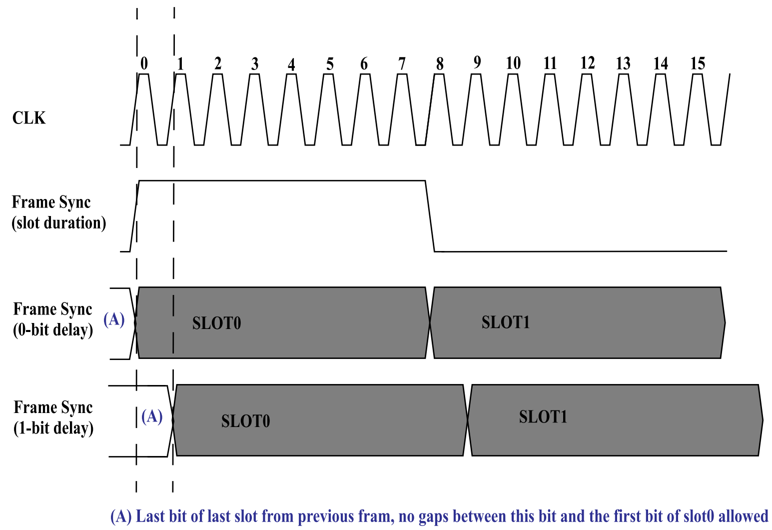


Figure 51-40. TDM Format Bit Delays from Frame Sync



51.5.16.11. Framed SPI Support, Subset of TDM

The macro supports Framed SPI protocol while operating in either Host or Client mode. The TDM_EN or bit enables causes the $\overline{SS}$ pin to be used as a frame synchronization pulse input or output pin. The state of SPIxCTRL_*.MSSEN bit is then ignored.

Unlike in normal SPI mode, the serial clock is continuous (free-running) in Framed SPI mode rather than being generated only when there is data to be transmitted. The data transmission/reception starts only when the frame synchronization pulse is generated at the $\overline{SS}$ pin. The device can be either a frame host if it generates the frame sync pulse or a frame client if it receives the frame sync pulse at the $\overline{SS}$ pin. In other words, only a frame host can generate the frame synchronization pulse.

Frame Host or Client mode is selected by clearing or setting the SPIxCTRL_*.FRMMST bit, respectively. The frame synchronization pulse can be an active-high or an active-low pulse of one SCK clock duration, or a multiple of (i.e. 8/16/24/32) bit character duration, based on the SPIxCTRL_*.FRMPOL, and SPIxCTRL_*.FRMSYPW settings.

Irrespective of which device is a host and which is a client, a framed SPI data transfer begins as soon as the frame host generates the frame sync pulse and writes the data to SPIxBUF. For full-duplex operation, the frame client should write to its buffer before the frame host does, in order to ensure that the data is ready at both ends when the data transfer begins.

Based on whether the SPI macro generates the serial clock and the frame synchronization pulse, four configurations are available to the user.

51.5.16.11.1. SPI Host, Frame Host

This mode is enabled by setting SPIxCTRL_*.TDM_EN = 1, SPIxCTRL_*.MSTEN = 1 and SPIxCTRL_*.FRMMST = 0. The serial clock is output at the SCK pin, regardless of whether the module is transmitting, and the $\overline{SS}$ pin is driven high on the next transmit edge of the SCK clock when the SPIx- BUF is written. Data will start transmitting on the subsequent transmit edge of the SPI clock.

51.5.16.11.2. SPI Host, Frame Client

This mode is enabled by setting SPIxCTRL_*.TDM_EN = 1, SPIxCTRL_*.MSTEN = 1and SPIxCTRL_*.FRMMST = 1. The SCK pin is an output while the $\overline{SS}$ pin is an input. When the $\overline{SS}$ pin is sampled high or low (SPIxCTRL_*.FRMPOL = 1/0), the data is transmitted on the subsequent transmit edge of the SPI clock. The user must make sure that the correct to-be-transmitted data is loaded into SPIx- BUF before the frame sync pulse is received at the $\overline{SS}$ pin.

51.5.16.11.3. SPI Client, Frame Host

This mode is enabled by setting SPIxCTRL_*.TDM_EN = 1, SPIxCTRL_*.MSTEN = 0 and SPIxCTRL_*.FRMMST = 0. The input clock at the SCK pin is continuous while the $\overline{SS}$ pin is an output.

51.5.16.11.4. SPI Client, Frame Client

This mode is enabled by setting SPIxCTRL_*.TDM_EN = 1, SPIxCTRL_*.MSTEN = 0 and SPIxCTRL_*.FRMMST = 1. Both SCK and $\overline{SS}$ pins are inputs.

51.5.16.11.5. SCK in Framed SPI Mode

SCK becomes an output when SPIxCTRL_*.TDM_EN = 1 and SPIxCTRL_*.MSTEN = 1. SCK becomes an input when SPIxCTRL_*.TDM_EN = 1 and SPIxCTRL_*.MSTEN = 0. In both cases, the source clock provided to the SCK pin is assumed to be free-running.

The polarity of the clock is selected by the SPIxCTRL_*.CPOL and SPIxCTRL_*.CPHA bits. Since the clock does not stop, the specification of transmission on transition from active to idle or idle to active clock states is moot. The end result is that there are only 2 actual cases of clock although the CPOL and CPHA bits can specify 4 cases.

When (CPOL = 0, CPHA = 1) or (CPOL = 1, CPHA = 0) the frame sync pulse output and the SDO data out- put change on the rising edge of the SCK clock.

When (CPOL = 1, CPHA = 1) or (CPOL = 0, CPHA = 0), the frame sync pulse output and the SDO data out- put change on the falling edge of the SCK clock.

Framed SPI mode works in all 8/16/32-bit environments. The frame sync pulse is generated for every 8/16/32-bits of data transmitted/received in 8/16/32-bit modes, respectively.

51.5.16.11.6. Frame Errors

A frame error occurs when the SPI detects a second frame sync pulse during a burst transfer. If this SPI is a frame host, it does not generate more than one frame sync pulse per frame burst. However, if the SPI is the frame client, it could receive multiple frame sync pulses if the data transmit size (as defined by FRMCNT) differs between it and the frame host. In such a case, the SPI captures the occurrence in the FRMERR bit, but continues with the original transfer count. If FRMERREN = 1, then that occurrence generates an error interrupt.

51.5.16.11.7. Data Buffers in Framed SPI Modes

When the macro is in frame host mode (SPIxCTRL_*.FRMMST = 0), the frame sync pulse is initiated when the user software writes to SPIxBUF, thereby loading the SPIxTXB register with the transmit data. Depending on SPIxCTRL_*.FRMCOINC, the data is transferred to SPIxSR and the send sequence begins. At the end of the send sequence, the data received is transferred to SPIxRXB and is available for the software to read from SPIxBUF.

Note: As long as the data is available in the transmit buffer, frame sync pulse is initiated (frame host mode) after completing a transmit/receive sequence.

When the macro is in frame client mode (SPIxCTRL_*.FRMMST = 1), the frame sync pulse is generated by an external source. When the macro samples the frame sync pulse, it transfers the contents of the SPIxTXB register to SPIxSR and the data transmission/reception begins. After the host/client transfer finishes, the received data is moved to SPIxRXB, which then can be read by the user software from SPIxBUF.

Note: Receiving a frame sync pulse (frame client mode) starts a transmit, regardless of the empty state of SPIxTXB. If the SPIxTXB is empty in the SCK cycle before the first bit time the SPI transmits zeros. If it is not empty it transmits the data. This prevents the corner case that is unavoidable in non-frame client mode.

51.5.16.11.8. Events in Framed SPI Mode

Event generation and timing in Framed SPI mode are similar to that of the normal SPI mode.

51.5.16.11.9. Enhanced Framed SPI Counter

For enhance framed SPI mode, the SPIxCTRL_*.FRMCNT register bits determines how many characters are sent/received for every frame sync pulse. The entire transaction is called a frame. If SPIxCTRL_*.FRMCNT = “000”, then a fsync pulse is generated for every data/character transmission. A simple case of generating a frame sync pulse for every 2 data characters (SPIxCTRL_*.FRMCNT= “1”).

In addition, the width of the frame sync pulse can be programmed to be either one clock wide or one character wide by programming SPIxCTRL_*.FRMSYPW.

51.5.16.11.10. Host Mode Client Select Enable

This mode is the same as the other non-framed host modes, but with the additional ability to drive a client select directly by using the FSYNC pin. In this mode, the SPIxCTRL_*.MSSEN and SPIxC-TRL_*.FRMPOL control bits determine the activation of the client select signal. The client select signal will be driven approximately one SCK cycle before and after the data transmission occurs.

51.5.16.11.11. Transmit Underrun Conditions

If the transmit buffer is empty when the SPI must load the send register to start (in the case of Framed Client) or continue (for either Framed Host or Client) a transfer, the SPI immediately sets SPITUR to indicate an underrun condition. If SPITUREN=1, the SPI asserts its error interrupt (to the interrupt controller).

While the SPI is in an underrun condition, the SPI transmits all zeroes until the end of the transaction as defined by FRMCNT. If the SPI is a frame client, another frame sync pulse can occur before the condition clears. In this case the SPI continues to transmit zeros. If the SPI is a frame host, it then waits for software to clear the under-run condition before initiating another sync pulse regardless of the state of the SPIxTXB.

51.5.16.11.12. Ignore Transmit Underrun

For cases when software does not care or need to know about the underrun condition, IGNTUR = 1 provides the serial engine the ability to ignore the under- run. When an underrun occurs, the SPI still sets the SPIxSTAT.SPITUR flag and obeys SPITUREN. Once SPITUR is set, it remains so until software clears it or SPIxCTRL_*.ENABLE = 0.

When the SPI is either a frame client or a frame host, an underrun event still causes the SPI to transmit zeros until the end of the frame as defined by FRMCNT. However, with IGNTUR = 1, the SPI can re-sample the underrun condition and continue to transmit data at each frame boundary.

If the SPI is a frame host, new data written to the SPIxTXB during a frame when an underrun condition exists does not get transmitted during that frame. But, the SPI evaluates the SPIxTXB continuously after the last frame. If data is in it, the SPI generates a frame sync and transmits the data.

If the SPI is a frame client, its transmit logic evaluates SPIxTXB for underrun during the next sync pulse. If the SPIxTXB contains data at the onset of the sync pulse,the SPI transmits that data. If not, the SPI transmits zero data until the end of the frame.

51.5.16.11.13. Transmit Underrun Recovery

When IGNTUR=1 and SPITUR=1, a software write of zero (0) to the SPITUR bit clears the condition; but it does NOT flush data in the SPIxTXB, which may have been put in after the condition occurred.

When IGNTUR=0 and SPITUR=1, a software write of zero (0) to the SPITUR bit clears the condition; and it flushes the data in the SPIxTXB. The SPI ignores writes to the SPIxTXB after clearing SPITUR until a read of SPIxSTAT when SPITUR = 0. This behavior ensures that a data service routine that is interrupted long enough to cause SPITUR, can’t inadvertently start a new framed transaction after the SPI error handler has cleared the error.

Note: Clearing the SPITUR affects the SPIxRXB.

Figure 51-41. SPI Host, Frame Host (CPOL=0, CPHA=1, FRMCOINC=0, FRMPOL=1, FRMCNT=0)

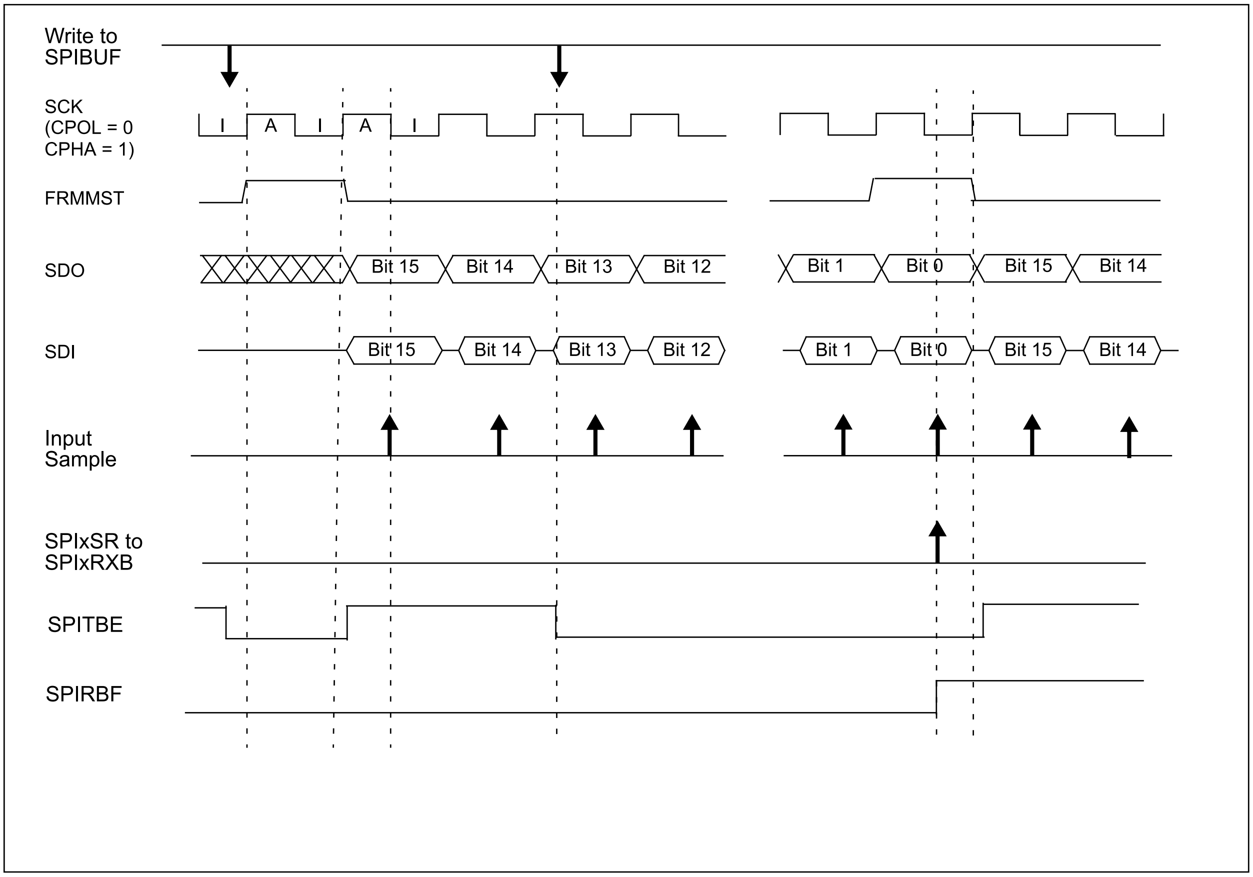


Figure 51-42. SPI Host, Frame Host (CPOL=0, CPHA=1, FRMCOINC=1, FRMPOL=1, FRMCNT=0)

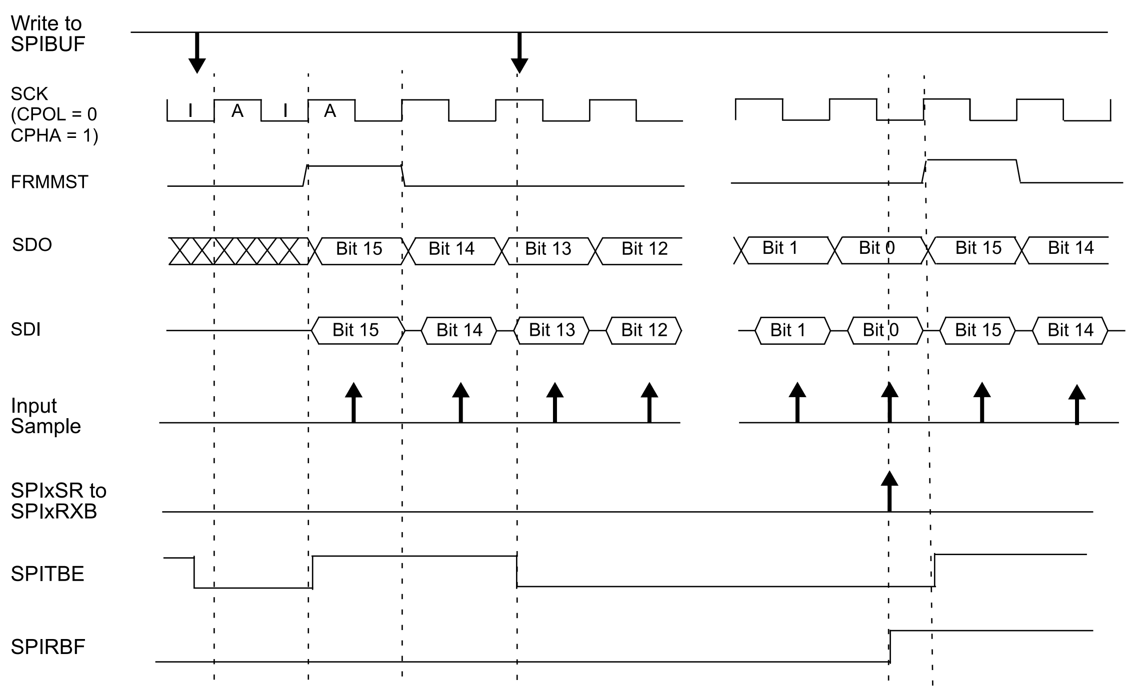


Figure 51-43. SPI Host, Frame Host (CPOL=0, CPHA=1, FRMCOINC=0, FRMPOL=1)

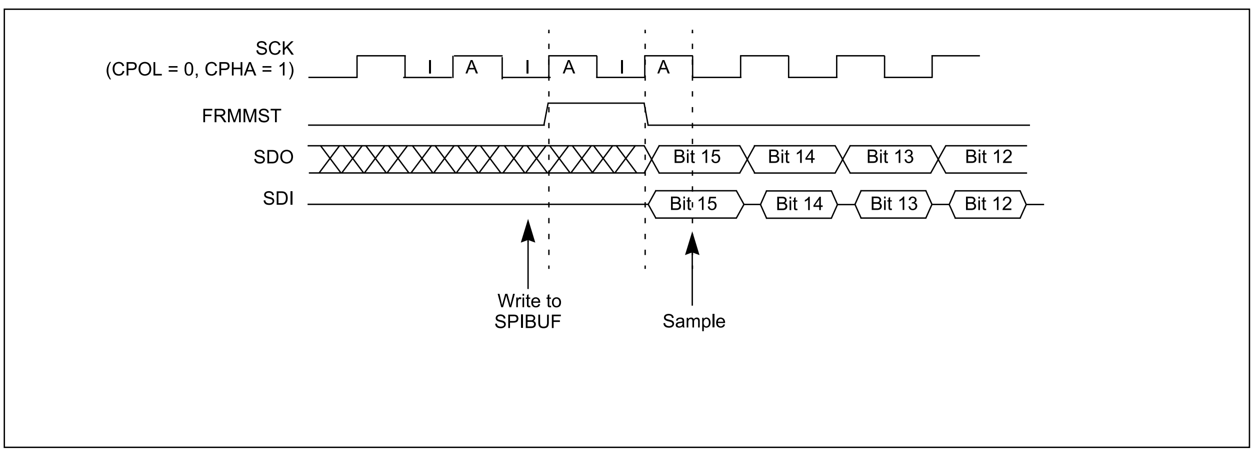


Figure 51-44. SPI Host, Frame Host (CPOL=0, CPHA=1, FRMCOINC=0, FRMPOL=0)

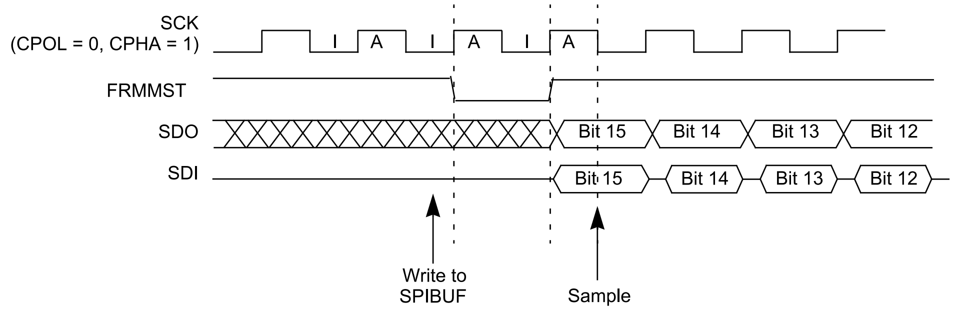


Figure 51-45. SPI Host, Frame Host (CPOL=CPHA, FRMCOINC=0, FRMPOL=1)

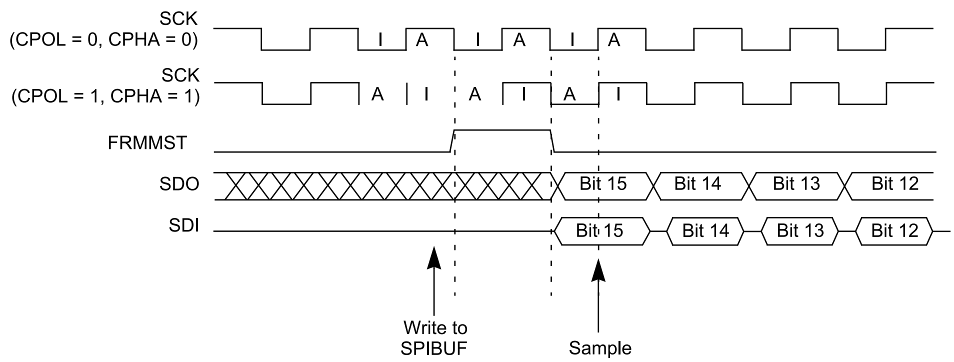


Figure 51-46. SPI Host, Frame Host (CPOL!=CPHA, FRMCOINC=0, FRMPOL=1)

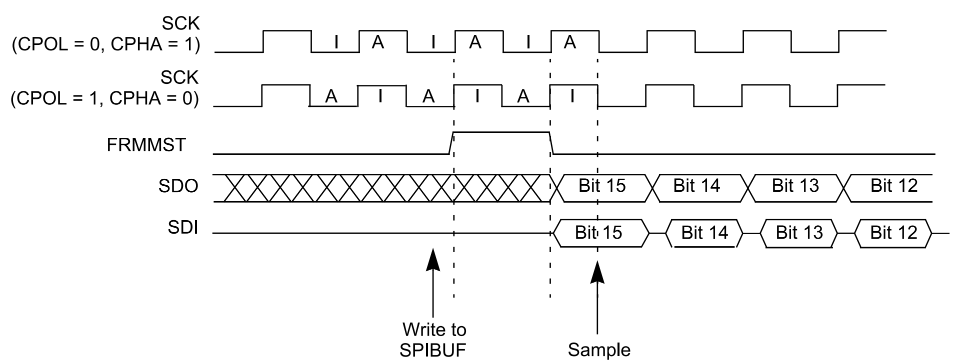


Figure 51-47. SPI Host, Frame Client (CPOL=0, CPHA=1, FRMCOINC=0, FRMPOL=1)

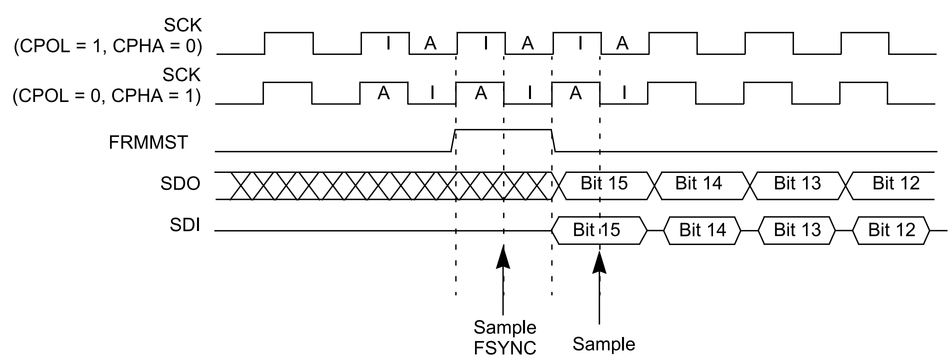


Figure 51-48. SPI Host, Frame Client (CPOL=CPHA, FRMCOINC=0, FRMPOL=1)

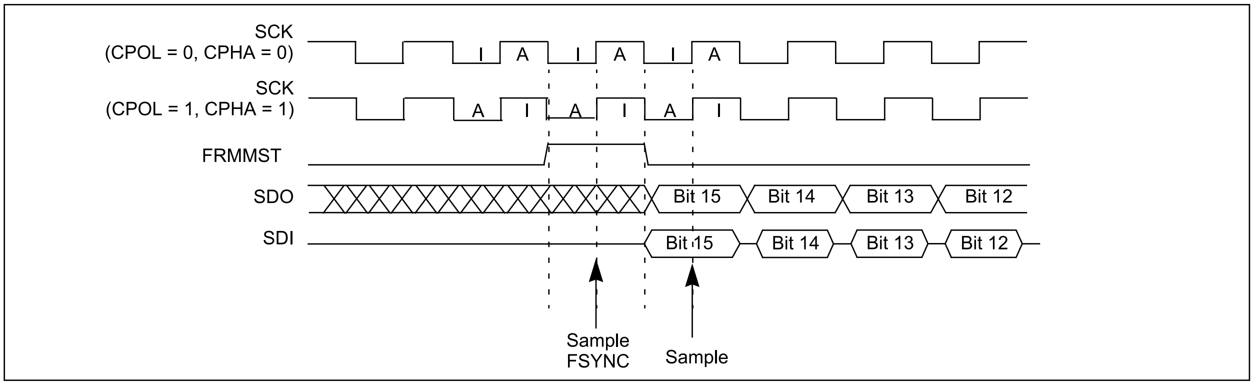


Figure 51-49. SPI Host Mode Interrupt Event Operation

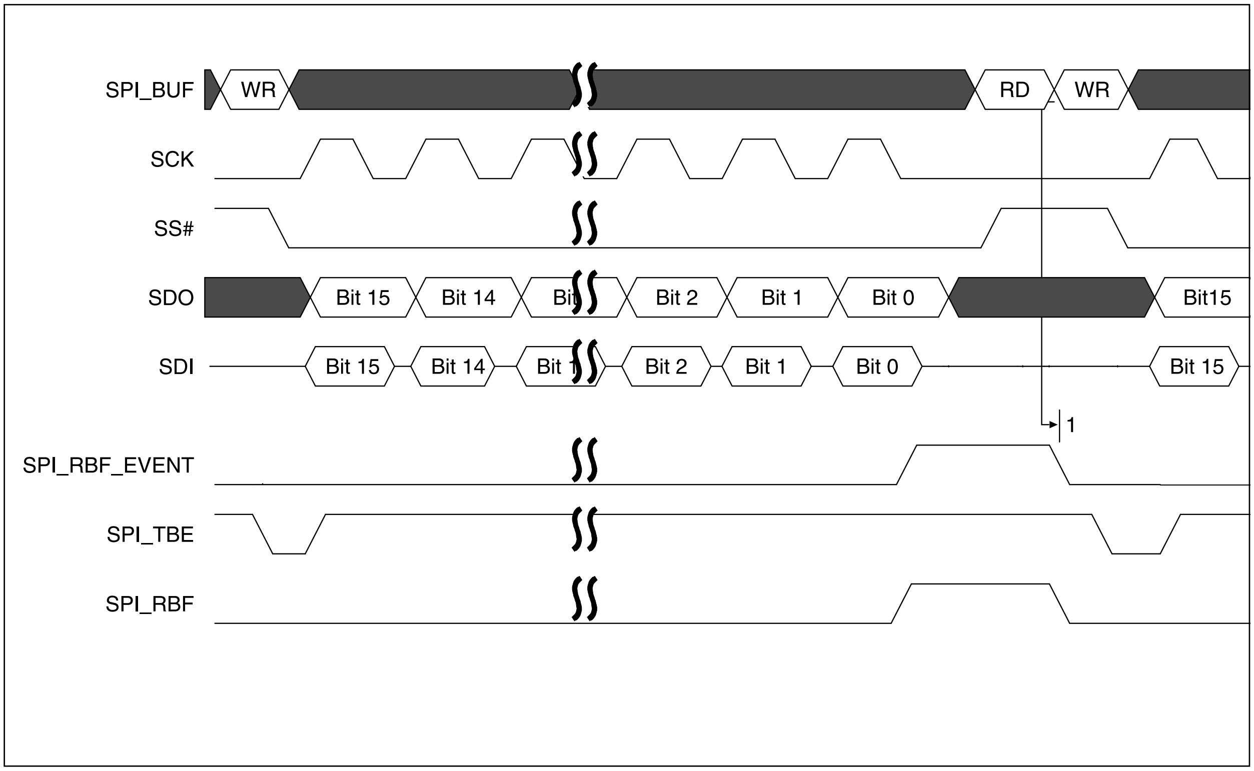
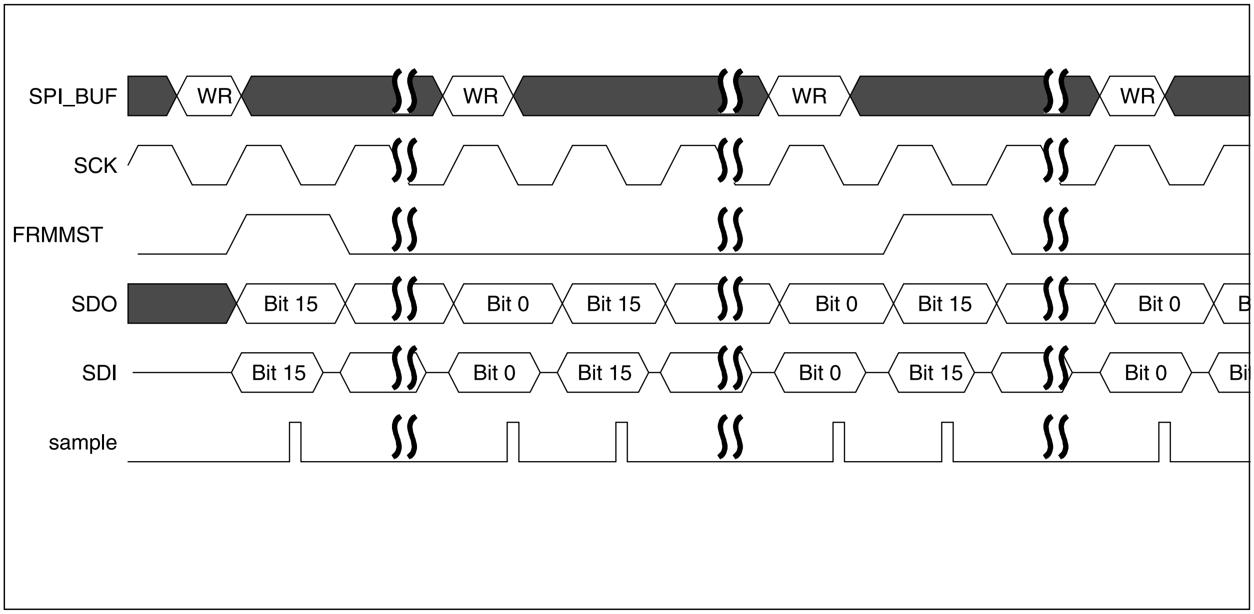


Figure 51-50. SPI Host, Frame Host (CPOL=0, CPHA=1, FRMCOINC=1, FRMPOL=1, FRMCNT=1)



51.5.16.12. SPI Operation in Power Save Modes

The Peripheral can be in any of the following power modes:

1. Operational Mode: Everything is running.
2. The peripheral continues to operate in all sleep modes that still provide its clocks.

51.6. Register Summary

For descriptions and definitions of both Register and bitfield properties, refer to Register Properties.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRLA	31:24								
		23:16								
		15:8								
		7:0		RUNSTDBY					ENABLE	SWRST
0x04	SELCTRL	31:24	MODEEN[1:0]							DATFILL
		23:16	TURSAMP							DATFMTLR
		15:8		IGNTUR	STXISEL[1:0]		CPOL	CPHA		
		7:0		IGNROV	SRXISEL[1:0]				CLKINDLY[1:0]	
0x08	SPICTRL	31:24								
		23:16								
		15:8						SMP	SPISGNEXT	
		7:0		CSEN	MSTEN	DISSDO	DISSDI		MODE32	MODE16
0x0C	FRAMECTRL	31:24	TDMWSZ[2:0]					TDMSSZ[2:0]		
		23:16				FRMCNT[4:0]				
		15:8		FRMSLV	FRMPOL					FRMCOINC
		7:0					FRMSYPW[3:0]			
0x10	AUDCTRL	31:24								
		23:16								
		15:8							AUDWDMODE[1:0]	
		7:0		AUDFMT[2:0]			AUDMONO		AUDMOD[1:0]	
0x14	TPDCTRL	31:24								
		23:16								
		15:8						SLVNUM[2:0]		
		7:0						PKFMT[2:0]		
0x18	INTENSET	31:24		SPIROVEN			SPITUREN			
		23:16								
		15:8	FRMERREN							
		7:0				SPITXBEEN				SPIRXBFEN
0x1C	INTENCLR	31:24		SPIROVEN			SPITUREN			
		23:16								
		15:8	FRMERREN							
		7:0				SPITXBEEN				SPIRXBFEN
0x20	INTFLAG	31:24		SPIROV			SPITUR			
		23:16								
		15:8	FRMERR							
		7:0				SPITXBE				SPIRXBF
0x24	STATUS	31:24	SPIRBE		SPIRBF	SPITBE		SPITBF		TXBUFELM8
		23:16	TXBUFELM7	TXBUFELM6	TXBUFELM5	TXBUFELM4	TXBUFELM3	TXBUFELM2	TXBUFELM1	TXBUFELM0
		15:8		SPIBUSY	SRMT					RXBUFELM8
		7:0	RXBUFELM7	RXBUFELM6	RXBUFELM5	RXBUFELM4	RXBUFELM3	RXBUFELM2	RXBUFELM1	RXBUFELM0
0x28	BUF	31:24	DATA[31:24]							
		23:16	DATA[23:16]							
		15:8	DATA[15:8]							
		7:0	DATA[7:0]							
0x2C	BRG	31:24								
		23:16								
		15:8				BRG[12:8]				
		7:0	BRG[7:0]							
0x30	DBGCTRL	31:24								
		23:16								
		15:8								
		7:0								DBGRUN

51.6.1. SPI Control Enable Register

Name: CTRLA
Offset: 0x00
Reset: 0x00
Property: PAC Write-Protection

Table 51-4. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

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

Bit 6 – RUNSTDBY Run in Standby Mode Enable Bit

Note: Writing a zero to this bit will disable the standby peripheral, Writing a one to this bit will enable the peripheral to be put into to standby.

Value	Description
0	Disable the macro for standby (no standby)
1	Enable macro for run standby

Bit 1 – ENABLE Enable (ON) bit

Note: Writing a zero to this bit will disable the peripheral, Writing a one to this bit will enable the peripheral.

Value	Description
0	Turn off and reset macro, disable clocks, disable interrupt event generation, allow SFR modifications.
1	Enable macro

Bit 0 – SWRST SPI Software Reset

Notes:

1. Writing a one to the SWRST bit resets the state of the module and all the registers, also the hidden registers, in the module to their initial state. The only exception is the DBGSTOP bit, which will keep its value after a SWRST. The module will be disabled after the reset. When writing a one to SWRST, no other bits in the same register will be written, as SWRST will clear all the bits in the same register. After writing a one to SWRST, SWRST will read back one until the module and the registers are reset. Any register write access during the ongoing reset will be discarded and an error will be generated. Read access can be performed without error generated and must return reset value. Writing a one to SWRST will have priority above all other actions, will always happen immediately and never stall the bus.
2. Writing a ‘0’ to SWRST has no effect.
3. During a SWRST, access to registers/bits without SWRST are disallowed until the CTRLA.SWRST is cleared by hardware.

Value	Description
0	There is no reset operation ongoing
1	The reset operation is ongoing

51.6.2. SPI Control Options Select Register

Name: SELCTRL
Offset: 0x04
Reset: 0x00
Property: PAC Write-Protection

Table 51-5. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	MODEEN[1:0]							DATFILL
Access	R/W	R/W						R/W
Reset	0	0						0

Bit	23	22	21	20	19	18	17	16
	TURSAMP							DATFMTLR
Access	R/W							R
Reset	0							0

Bit	15	14	13	12	11	10	9	8
		IGNTUR	STXISEL[1:0]		CPOL	CPHA		
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
		IGNROV	SRXISEL[1:0]				CLKINDLY[1:0]	
Access		R/W	R/W	R/W			R/W	R/W
Reset		0	0	0			0	0

Bits 31:30 – MODEEN[1:0] Macro Mode Enabled

Notes:

- MODE_EN (TPD_EN, TDM_EN, AUDEN) can only be written when Enable ='0'
- When AUDEN = 1, MSTEN controls the direction of both SCK and Frame (aka LRC)
- FRMEN = 1, When AUDEN = 1 or TDM_EN =1 or TPD_EN=1
- When AUDEN = 1, this peripheral functions as if FRMMST = ~MSTEN, regardless of its actual value
- When AUDEN = 1, this peripheral functions as if SMP = 0, regardless of its actual value

Value	Description
`11`	TPD ENABLED (TPD_EN =1)
`10`	TDM_ENABLED (TDM_EN =1)
`01`	Enable Audio CODEC Support (AUDEN =1)
`00`	DEFAULT - AUDEN =0,TDM_EN =0, TPD_EN = 0)

Bit 24 – DATFILL DATFILL undefined bits 1 or 0

Notes:

1. Can only be written when Enable ='0', and is not used when using sign extended (SPISGNEXT).
2. DATAFILL is not applicable for Received data. Only sign extension will.

Value	Description
0	fill undefined bits with“0.”
1	fill undefined bits with“1.”

Bit 23 – TURSAMP Transmit Under-run last sample sent
Note: Can only be written when Enable =’0’.

Value	Description
0	Transmit Under-run last sample send out 0’s
1	Transmit Under-run last sample, send out last sample of the channel (transmits previously received data), if there is no last sample, 0's will be sent out.

Bit 16 – DATFMTLR Packed data format - left or right justified
Note: Can only be written when Enable=’0’ in audio mode and TPD mode, only a setting of TUSSAMP = 0 should be used.

Value	Description
0	Data is Left Justified (in the upper part for the packed data)
1	Data is Right Justified (in the lower part for the packed data)

Bit 14 – IGNTUR Ignore Transmit Underrun
Note: Can only be written when Enable=’0’.

Value	Description
0	A TUR is a critical error which stop SPI operation
1	A TUR is NOT a critical error and zeros are transmitted until the SPIxTXB is not empty

Bits 13:12 – STXISEL[1:0] SPI Transmit Service Request Interrupt Select
Note: Can only be written when Enable=’0’.

The SPI generates a Transmit Service Request when:

Value	Description
11	The SPIxTXB is not full
10	The SPIxTXB is at least half empty
01	The SPIxTXB is empty
00	The SPIxTXB is empty and SPIxSR is empty (i.e. all transmit operations are complete)

Bit 11 – CPOL Clock Polarity Select bit
Note: Can only be written when Enable=’0’.

Value	Description
0	Idle state for clock is a low level; active state is a high level
1	Idle state for clock is a high level; active state is a low level

Bit 10 – CPHA SPI Clock Edge Select bit
Notes:
- Can only be written when Enable=’0’
- When AUDEN = 1, this peripheral functions as if CPHA = 1, regardless of its actual value

Value	Description
0	Transmit happens on transition from active clock state to idle clock state.
1	Transmit happens on transition from idle clock state to active clock state.

Bit 6 – IGNROV Ignore Receive Overflow (for Audio Data Transmissions)
Note: Can only be written when Enable=’0’.

Value	Description
0	A ROV is a critical error which stop SPI operation.
1	A ROV is NOT a critical error; during ROV data in the FIFO is not overwritten by receive data.

Bits 5:4 – SRXISEL[1:0] SPI Receive Service Request Interrupt Select
Note: Can only be written when Enable='0'.

The SPI generates a Receive Service Request when:

Value	Description
11	The SPIxRXB is full
10	The SPIxRXB is at least half full
01	The SPIxRXB is not empty
00	The SPIxRXB is empty

Bits 1:0 – CLKINDLY[1:0] Serial Clock Input Delay for SDI sampling
Notes:

- Can only be written when Enable='0'
- CLKINDLY[x] is used by the SPI FSM when MSTEN=1 (i.e. the SPI is a clock host). For all other case the value is ignored and the SPI FSM does NOT delay SCK for SDI sampling.
- See TscInDly parameter for tap delay resolution

Value	Description
11	3 tap delays added to clock input
10	2 tap delays added to clock input
01	1 tap delay added to clock input
00	0 tap delays added to clock input

51.6.3. SPI Control Register

Name: SPICTRL
Offset: 0x08
Reset: 0x00
Property: PAC Write-Protection

Table 51-6. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
						SMP	SPISGNEXT	
Access						R/W	R/W	
Reset						0	0	

Bit	7	6	5	4	3	2	1	0
		CSEN	MSTEN	DISSDO	DISSDI		MODE32	MODE16
Access		R/W	R/W	R/W	R/W		R/W	R/W
Reset		0	0	0	0		0	0

Bit 10 – SMP SPI Data Input Sample Phase bit
Input data is always sampled at the middle of data output time regardless of the SMP setting in Client Mode.

Value	Description
0	Input data sampled at the middle of data output time in Host Mode
1	Input data sampled at the end of data output time in Host Mode

Bit 9 – SPISGNEXT Sign Extend Read Data from the RXFIFO

Value	Description
0	Data from RX FIFO is not sign extended
1	Data from RX FIFO is sign extended

Bit 6 – CSEN Host/Client Mode Select Enable bit
Notes:
- When FRMEN = 1, MSSEN is not used
- Use is dependent on MSTEN bit
- Can only be written when Enable ='0'

Value	Description
0	**Host Mode:** Client select SPI support disabled
1	**Host Mode:** SPI Client Select support enabled with polarity determined by FRMPOL ($\overline{SS}$ pin automatically driven during transmission in Host Mode)
0	**Client Mode:** $\overline{SS}$ pin used by the macro in Client mode; ($\overline{SS}$ pin used as client select input)
1	**Client Mode:** $\overline{SS}$ pin not used by client mode

Bit 5 – MSTEN Host Mode Enable bit

Note: Can only be written when Enable ='0'.

Value	Description
0	Client Mode
1	Host Mode

Bit 4 – DISSDO Disable SDO bit

Note: Can only be written when Enable ='0'.

Value	Description
0	SDO pin is controlled by the macro.
1	SDO pin is not used by the macro. Pin controlled by PORT function.

Bit 3 – DISSDI Disable SDI bit

Note: Can only be written when Enable ='0'.

Value	Description
0	SDI pin is controlled by the macro.
1	SDI pin is not used by the macro. Pin controlled by PORT function.

Bit 1 – MODE32 Serial Word Length bits for AUDEN=0

MODE32 - For AUDEN=0	Communication
1	32-bit
0	16-bit
0	8-bit

Notes:

- Can only be written when Enable ='0'
- Not used when AUDEN=1
- Channel is not meaningful for DSP/PCM mode as LRC follows FRMSYPW

Bit 0 – MODE16 Serial Word Length bits for AUDEN=0

MODE16 - For AUDEN=0	Communication
X	32-bit
1	16-bit
0	8-bit

Notes:

- Can only be written when Enable ='0'
- Not used when AUDEN=1
- Channel is not meaningful for DSP/PCM mode as LRC follows FRMSYPW

51.6.4. SPI Control Frame Register

Name: FRAMECTRL
Offset: 0x0C
Reset: 0x00
Property: PAC Write-Protection

Table 51-7. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	TDMWSZ[2:0]					TDMSSZ[2:0]		
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
				FRMCNT[4:0]				
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
		FRMSLV	FRMPOL					FRMCOINC
Access		R/W	R/W					R/W
Reset		0	0					0

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

Bits 31:29 – TDMWSZ[2:0] TTDM Number of Bits in a Word Size
Note: TDMWSZ can only be written when ON bit ='0', and is only valid for TDM_EN = 1.

Value	Description
111	Reserved
110	word size number of bits = 32
101	word size number of bits = 28
100	word size number of bits = 24
011	word size number of bits = 20
010	word size number of bits = 16
001	word size number of bits = 12
000	word size number of bits = 8

Bits 26:24 – TDMSSZ[2:0] TTDM Number of Bits in a Slot Size
Note: TDMSSZ can only be written when ON bit ='0', and is only valid for TDM_EN = 1.

Value	Description
111	Reserved
110	slot size number of bits = 32
101	slot size number of bits = 28
100	slot size number of bits = 24

Value	Description
011	slot size number of bits = 20
010	slot size number of bits = 16
001	slot size number of bits = 12
000	slot size number of bits = 8

Bits 20:16 – FRMCNT[4:0] Frame Sync Pulse Counter

Controls the number of Slots (Serial Words) transmitted per sync pulse.

Notes:

- FRMCNT is only valid when FRMEN = 1 (i.e., Framed SPI mode/TDM mode)
- Can only be written when ON bit = ’0’

Value	Description
10011-11111	Reserved
10010	Number of slots per frame sync pulse is 32
10001	Number of slots per frame sync pulse is 30
10000	Number of slots per frame sync pulse is 28
01111	Number of slots per frame sync pulse is 26
01110	Number of slots per frame sync pulse is 24
01101	Number of slots per frame sync pulse is 22
01100	Number of slots per frame sync pulse is 20
01011	Number of slots per frame sync pulse is 18
01010	Number of slots per frame sync pulse is 16
01001	Number of slots per frame sync pulse is 14
01000	Number of slots per frame sync pulse is 12
00111	Number of slots per frame sync pulse is 10
00110	Number of slots per frame sync pulse is 8
00101	Number of slots per frame sync pulse is 6
00100	Number of slots per frame sync pulse is 5
00011	Number of slots per frame sync pulse is 4
00010	Number of slots per frame sync pulse is 3
00001	Number of slots per frame sync pulse is 2
00000	Generate a frame sync pulse on each Serial Word.

Bit 14 – FRMSLV Frame Sync Pulse Direction Control bit

Note: Can only be written when ON bit = ’0’.

Value	Description
0	Frame sync pulse output (Host)
1	Frame sync pulse input (Client)

Bit 13 – FRMPOL Frame Sync/Client Select Polarity bit

Notes:

- Can only be written when ON bit = ’0’
- Valid when FRMEN = ‘1’ or SPI Host mode and CSEN = ‘1’

Value	Description
0	Frame pulse/Client Select is active low
1	Frame pulse/Client Select is active high

Bit 8 – FRMCOINC Frame Sync Pulse Edge Select bit

Note: Can only be written when ON bit = ’0’.

Value	Description
0	Frame synchronization pulse (idle-to-active edge) precedes the first bit clock.

Value	Description
1	Frame synchronization pulse (idle-to-active edge) coincides with the first bit clock.

Bits 3:0 – FRMSYPW[3:0] Frame Sync Pulse Width in Serial Words

(As defined by AUDWDMODE[1,0], or MODE[32,16] or TDMSSZ/TDMWSZ for TDM) -- (i.e., Framed SPI Mode, I^2S, I^8S, TDM,TPD).)
Not all settings are valid for all MODES, and must be set by the user correctly for different MODES like AUDIO, TDM, TPD SPI.
Not all settings are valid for all MODES and must be set by the user correctly for different MODES like AUDIO, TDM, TPD SPI.

Notes:

- Can only be written when ON bit ='0'.
- The sync pulse is four Serial Word Length wide used for I^8S.
- In TDM mode the serial word length is defined by TDMSSZ/TDMWSZ and not (AUDWDMODE[1:0] or MODE[32,16])

Value	Description
1011 – 1111	Reserved
1010	Frame sync pulse is thirty-two Serial Word (32 slot) Length wide
1001	Frame sync pulse is sixteen Serial Word (16 slot) Length wide
1000	Frame sync pulse is eight Serial Word (8 slot) Length wide
0111	Frame sync pulse is seven Serial Word (7 slot) Length wide
0110	Frame sync pulse is six Serial Word (6 slot) Length wide
0101	Frame sync pulse is five Serial Word (5 slot) Length wide
0100	Frame sync pulse is four Serial Word (4 slot) Length wide
0011	Frame sync pulse is three Serial Word (3 slot) Length wide
0010	Frame sync pulse is two Serial Word (2 slot) Length wide
0001	Frame sync pulse is one Serial Word (1 slot) Length wide
0000	Frame sync pulse is one clock (SCK) wide

51.6.5. SPI Control Audio Register

Name: AUDCTRL
Offset: 0x10
Reset: 0x00
Property: PAC Write-Protection

Table 51-8. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
							AUDWDMODE[1:0]	
Access							R/W	R/W
Reset							0	0

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

Bits 9:8 – AUDWDMODE[1:0] Serial Word Length bits (Ignored when AUDEN=0)

For AUDEN = 1: & AUDFMT= 3'b000		
COMMUNICATION		
1	1	24-bit Data, 32-bit Channel/1/32-bit Frame pulse
1	0	32-bit Data, 32-bit Channel/1/32-bit Frame pulse
0	1	16-bit Data, 32-bit Channel/1/16/32-bit Frame pulse
0	0	16-bit Data, 16-bit Channel/1/16-bit Frame pulse
For AUDEN = 1: & AUDFMT= 3'b001		
COMMUNICATION		
1	1	Reserved
1	0	I²S -24-bit raw audio data (slot), 32-Bit Ch/1/32-bit Frame pulse
0	1	I²S -20-bit raw audio data (slot), 32-Bit Ch/1/32-bit Frame pulse
0	0	I²S -16-bit raw audio data (slot), 32-Bit Ch/1/32-bit Frame pulse
For AUDEN = 1: & AUDFMT= 3'b010		
COMMUNICATION		
1	1	I²S -16-bit x2 audio packed left down, 32-Bit Ch/Ch/1/32-bit Frame pulse
1	0	I²S -16-bit x2 audio packed left up, 32-Bit Ch/Ch/1/32-bit Frame pulse
0	1	Reserved
0	0	I²S -24-bit audio packed, 32-Bit Ch/Ch/1/32-bit Frame pulse
For AUDEN = 1: & AUDFMT= 3'b101		
COMMUNICATION		
1	1	Reserved

1	0	I^{8}S -24-bit raw audio data, 32-Bit Ch, 1/32/128-bit Frame pulse
0	1	I^{8}S -20-bit raw audio data, 32-Bit Ch, 1/32/128-bit Frame pulse
0	0	I^{8}S -16-bit raw audio data, 32-Bit Ch, 1/32/128-bit Frame pulse
For AUDEN = 1: & AUDFMT= 3'b110		
COMMUNICATION		
1	1	I^{8}S -16-bit x2 audio packed left down, 32-Bit Ch, 1/32/128-bit Frame pulse
1	0	I^{8}S -16-bit x2 audio packed left up, 32-Bit Ch, 1/32/128-bit Frame pulse
0	1	I^{8}S -32-bit data, 32-Bit Ch/128-bit Frame pulse
0	0	I^{8}S -24-bit audio packed, 32-Bit Ch, 1/32/128-bit Frame pulse
For AUDEN = 1: & AUDFMT= 3'b100		
COMMUNICATION		
1	1	Reserved
1	0	I^{8}S -24-bit MSB al, mute lower bits, 32-Bit Ch, 1/32/128-bit Frame pulse
0	1	I^{8}S -20-bit MSB al, mute lower bits, 32-Bit Ch, 1/32/128-bit Frame pulse
0	0	I^{8}S -16-bit MSB al, mute lower bits, 32-Bit Ch,1/32/128-bit Frame pulse

Notes:
- Channel is not meaningful for DSP/PCM mode as LRC follows FRMSYPW
- Can only be written when ON bit ='0', and is only valid for AUDEN = 1

Bits 6:4 – AUDFMT[2:0] Audio Protocol Format

Notes:
1. Generate a frame sync pulse on every 8 Serial Words.(Value used by Audio I8S Protocols.)
2. Generate a frame sync pulse on every 2 Serial Words. (Value used by Audio I2S Protocols.)
3. Can only be written when ON bit ='0', and is only valid for AUDEN = 1.

Value	Description
111	Reserved
110	I^{8}S Other AM824 formats Mode see note 2
101	I^{8}S AM824 Raw audio format Mode see note 2
100	I^{8}S Other formats see note 2
011	Reserved
010	I^{2}S Other AM824 formats Mode see note 1
001	I^{2}S AM824 Raw audio format Mode see note 1
000	Legacy I^{2}S modes see note 1

Bit 3 – AUDMONO Transmit audio data format

Note: Can only be written when ON bit ='0', and is only valid for AUDEN = 1.

Value	Description
0	Audio Data is Stereo
1	Audio Data is Mono (Each data word is transmitted on both left and right channels)

Bits 1:0 – AUDMOD[1:0] Audio Protocol Mode

Notes:
1. Can only be written when ON bit ='0', and is only valid for AUDEN = 1.
2. In I^{2}S Mode, this peripheral functions as if FRMCOINC=0, regardless of its actual value.
3. In Right or Left Justified Mode (DATFMT_LR), this peripheral functions as if FRMCOINC=1, regardless of its actual value.
4. When not in PCM/DSP Mode, this peripheral functions as if FRMSYPW=0001, regardless of it actual value.
5. AUDFMT is used to select between I^{2}S, I^{8}S, PCM audio modes.

Value	Description
11	PCM/DSP Mode

Value	Description
10	Reserved
01	I^2S, I^8S right/left Justified Mode -see DATFMT_LR
00	I^2S, I^8S Standard Mode

51.6.6. SPI Control TPD Register

Name: TPDCTRL
Offset: 0x14
Reset: 0x00
Property: PAC Write-Protection, Enable-Protected

Table 51-9. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
						SLVNUM[2:0]		
Access						R/W	R/W	R/W
Reset						0	0	0

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

Bits 10:8 – SLVNUM[2:0] Number of the Client designated: to be used with the PKFMT to determine the client.

Not valid for a Host and all other decodes are ignored and not allowed.

Value	Description
000	Client 0
001	Client 1
010	Client 2
011	Client 3
100	Client 4

Bits 2:0 – PKFMT[2:0] Host Client TPD mode.

Notes:

1. Can only be written when ON bit ='0', and is only valid for TPD_EN = 1.
2. The transmitted channel is as follows: 000, 001 = channel = 32 bits.
3. The transmitted channel is as follows: 010, 011 = channel = 32 bits -- 24bits and formatted by other registers (DATAFIL, DATFMTLR).
4. The transmitted channel is as follows: 100, 101, 110 = channel= 16 bits.

Value	Description
000	32-bit data in 4x32 packed format
001	32-bit data in 3x32 packed format
010	24-bit data in 4x24 packed format
011	24-bit data in 3x24 packed format
100	16-bit data in 6x16 packed format
101	16-bit data in 4x16 packed format
110	16-bit data in 2x16 packed format

51.6.7. SPI Interrupt Enable Set Register

Name: INTENSET
Offset: 0x18
Reset: 0x00
Property: PAC Write-Protection, Enable-Protected

Notes:
1. 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.
2. In TPD mode is recommended to not enable the devices dedicated as clients interrupts. The SPITUREN, SPIROUEN, and FRMERREN for the Clients should all be disabled and only enable the Host's interrupts as needed.

Table 51-10. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
		SPIROVEN			SPITUREN			
Access		R/W			R/W			
Reset		0			0			

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

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

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

Bit 30 – SPIROVEN Enable Interrupt Events via SPIROV
Writing a zero to this bit has no effect.
Writing a one to this bit will SET the Enable bit.

Value	Description
0	Disables Receive Overflow (Does Not) Generates Error Events
1	Enables Receive Overflow Generates Error Events

Bit 27 – SPITUREN Enable Interrupt Events via SPITUR
Writing a zero to this bit has no effect.
Writing a one to this bit will SET the Enable bit.

Value	Description
0	Disables Transmit Under-run (Does Not) Generates Error Events

Value	Description
1	Enables Transmit Under-run Generates Error Events

Bit 15 – FRMERREN Enable Interrupt Events via FRMERR

Writing a zero to this bit has no effect.
Writing a one to this bit will SET the Enable bit.

Value	Description
0	Disables Frame Error (Does not) Generates Error Events
1	Enables Frame Error Generates Error Events

Bit 4 – SPITXBEEN Enable Interrupt Events via SPITXBE

Writing a zero to this bit has no effect.
Writing a one to this bit will SET the Enable bit.

Value	Description
0	Disables (Does Not) Generates TXBE Error Events
1	Enables Generates TXBE Events

Bit 0 – SPIRXBFEN Enable Interrupt Events via SPIRXBF

Writing a zero to this bit has no effect.
Writing a one to this bit will SET the Enable bit.

Value	Description
0	Disables (Does Not) Generates RXBF Error Events
1	Enables Generates RXBF Events

51.6.8. SPI Interrupt Enable Clear Register

Name: INTENCLR
Offset: 0x1C
Reset: 0x00
Property: PAC Write-Protection, Enable-Protected

Notes:

1. 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.
2. In TPD mode is recommended to not enable the devices dedicated as clients interrupts. The SPITUREN, SPIROUEN, and FRMERREN for the Clients should all be disabled and only enable the Host's interrupts as needed.

Table 51-11. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
		SPIROVEN			SPITUREN			
Access		R/W			R/W			
Reset		0			0			

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

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

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

Bit 30 – SPIROVEN Enable Interrupt Events via SPIROV
Writing a zero to this bit has no effect.
Writing a one to this bit will SET the Enable bit.

Value	Description
0	Disables Receive Overflow (Does Not) Generates Error Events
1	Enables Receive Overflow Generates Error Events

Bit 27 – SPITUREN Enable Interrupt Events via SPITUR
Writing a zero to this bit has no effect.
Writing a one to this bit will SET the Enable bit.

Value	Description
0	Disables Transmit Under-run (Does Not) Generates Error Events

Value	Description
1	Enables Transmit Under-run Generates Error Events

Bit 15 – FRMERREN Enable Interrupt Events via FRMERR
Writing a zero to this bit has no effect.
Writing a one to this bit will SET the Enable bit.

Value	Description
0	Disables Frame Error (Does not) Generates Error Events
1	Enables Frame Error Generates Error Events

Bit 4 – SPITXBEEN Enable Interrupt Events via SPITXBE
Writing a zero to this bit has no effect.
Writing a one to this bit will SET the Enable bit.

Value	Description
0	Disables (Does Not) Generates TXBE Error Events
1	Enables Generates TXBE Events

Bit 0 – SPIRXBFEN Enable Interrupt Events via SPIRXBF
Writing a zero to this bit has no effect.
Writing a one to this bit will SET the Enable bit.

Value	Description
0	Disables (Does Not) Generates RXBF Error Events
1	Enables Generates RXBF Events

51.6.9. SPI Interrupt Flag Register

Name: INTFLAG
Offset: 0x20
Reset: 0x00
Property: PAC Write-Protection

Note: In TPD mode is recommended to not enable the devices dedicated as clients interrupts. The SPITUREN, SPIROUEN, and FRMERREN for the Clients should all be disabled and only enable the Host's interrupts as needed.

Note: Interrupt flags must be cleared and then read back to confirm they are cleared before exiting the ISR to avoid double interrupts.

Table 51-12. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
		SPIROV			SPITUR			
Access		R/W			R/W			
Reset		0			0			

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

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

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

Bit 30 – SPIROV Receive Overflow Status bit

Note: Cleared only by software, Writing a zero to this bit has no effect, Writing a one to this bit will CLEAR the bit.

Value	Description
0	No overflow
1	A new byte/half-word/word has been completely received when the SPIxRXB was full

Bit 27 – SPITUR Transmit Underrun Status bit

Writing a zero to this bit has no effect.
Writing a one to this bit will SET the Enable bit.
Not cleared with FIFO operation. Only Hardware cleared by the ON bit.

Notes:
1. SPITUR is only valid when FRMEN = 1.
2. SPITUR is also cleared when ON = 0.
3. When IGNTUR = 1, SPITUR provides dynamic status of the underrun condition but does not stop Rx/Tx operation and does not need to be cleared by software.

Value	Description
0	Transmit buffer has No underrun condition
1	Transmit buffer has encountered an underrun condition

Bit 15 – FRMERR SPI Frame Error status bit
Writing a zero to this bit has no effect.
Writing a one to this bit will SET the Enable bit.
Note: FRMERR is only valid when FRMEN =1.

Value	Description
0	No Frame error detected
1	Frame error detected

Bit 4 – SPITXBE SPI Transmit Buffer Empty Flag bit
SPITXBE reflects the full status of the multi-element FIFO.

Value	Description
0	TXB not full
1	TXB is full

Bit 0 – SPIRXBF RX Buffer Full Flag bit
SPIRXBF reflects the full status of the multi-element FIFO.

Value	Description
0	RX Buffer not full
1	RX Buffer full

51.6.10. SPI Status Register

Name: STATUS
Offset: 0x24
Reset: 0x90002000
Property: PAC Write-Protection

Table 51-13. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	SPIRBE		SPIRBF	SPITBE		SPITBF		TXBUFELM8
Access	R/W		R/W	R/W		R/W		R/W
Reset	1		0	1		0		0

Bit	23	22	21	20	19	18	17	16
	TXBUFELM7	TXBUFELM6	TXBUFELM5	TXBUFELM4	TXBUFELM3	TXBUFELM2	TXBUFELM1	TXBUFELM0
Access	R/W	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
		SPIBUSY	SRMT					RXBUFELM8
Access		R/W	R/W					R/W
Reset		0	1					0

Bit	7	6	5	4	3	2	1	0
	RXBUFELM7	RXBUFELM6	RXBUFELM5	RXBUFELM4	RXBUFELM3	RXBUFELM2	RXBUFELM1	RXBUFELM0
Access	R/W	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 – SPIRBE RX Buffer Empty bit
SPIRBF reflects the empty status of the multi-element FIFO.

Value	Description
0	RX Buffer not Empty
1	RX Buffer Empty

Bit 29 – SPIRBF SPI Receive Buffer Full status bit
SPIRBF reflects the full status of the multi-element FIFO.

Value	Description
0	SPIxRXB is not full
1	SPIxRXB is full

Bit 28 – SPITBE SPI Transmit Buffer Empty status bit
SPITBE reflects the empty status of the multi-element FIFO.

Value	Description
0	SPIxTXB is not empty
1	SPIxTXB is empty

Bit 26 – SPITBF SPI Transmit Buffer Full Status bit
SPITBF reflects the full status of the multi-element FIFO.

Value	Description
0	SPIxTXB not full
1	SPIxTXB is full

Bits 16, 17, 18, 19, 20, 21, 22, 23, 24 – TXBUFELM Transmit Buffer Element Count bits
Reflects the number of FIFO elements used.
8 bit data is 1 element (total of 64 elements available)
8 = 16 bit data is 2 elements (total of 32 elements available)
16 = 24 bit data is 3 elements (total of 21elements available)
24 bit data is 4 elements (total of 16 elements available)

Bit 14 – SPIBUSY SPI activity status bit

Value	Description
0	No on-going transactions (at time of read)
1	Macro currently busy with some transactions

Bit 13 – SRMT Register (SPIxSR) Empty bit

Value	Description
0	There are current or pending transactions.
1	There are no current or pending transactions. (i.e. Neither SPIxTXB or SPIxSR contain data to transmit)

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8 – RXBUFELM Receive Buffer Element Count bits
Reflects the number of FIFO elements used
8 bit data is 1 element (total of 64 elements available)
8 = 16 bit data is 2 elements (total of 32 elements available)
16 = 24 bit data is 3 elements (total of 21 elements available)
24 bit data is 4 elements (total of 16 elements available)

51.6.11. SPI Buffer Register

Name: BUF
Offset: 0x28
Reset: 0x00
Property: PAC Write-Protection

Table 51-14. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable bit	—	—

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] FIFO Data bits

Notes:

1. When MODE[32,16] or AUDWD_MODE[1:0] selects 32-bit data, the SPI uses DATA[31:0].
2. When MODE[32,16] or AUDWD_MODE[1:0] selects 24-bit data, the SPI only uses DATA[24:0].
3. When MODE[32,16] or AUDWD_MODE[1:0] selects 16-bit data, the SPI only uses DATA[15:0].
4. When MODE[32,16] or AUDWD_MODE[1:0] selects 8-bit data, the SPI only uses DATA[7:0].
5. The SPI pushes data to the SPIxTXB on the write to the highest byte defined by MODE[32,16] or AUDWD_MODE[1:0]. Likewise, the SPI pops data from the SPIxRXB on the read of the highest byte defined by MODE[32,16] or AUDWD_MODE[1:0]. For instance, if writing data 8-bits at a time in 32-bit mode, the SPI commits data to the SPIxTXB on a write to DATA[31:24]. The SPI does not use or track any other byte write location to determine commitment to the buffer.

51.6.12. SPI Baud Rate Register

Name: BRG
Offset: 0x2C
Reset: 0x00
Property: PAC Write-Protection

Table 51-15. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
				BRG[12:8]				
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	BRG[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 12:0 – BRG[12:0] Baud Rate Divisor bits

51.6.13. SPI Debug Control Register

Name: DBGCTRL
Offset: 0x30
Reset: 0x00
Property: PAC Write-Protection

Table 51-16. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	**HC**	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	**HS**	Set by Hardware	**X**	Bit is unknown at Reset
K	Write to clear	**S**	Software settable 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
Access								
Reset								

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

Bit 0 – DBGRUN Debug Running State
This bit is not affected by software reset and should not be changed by software while the SPIx is enabled.

Value	Description
0	Macro is halted when the device is halted in debug mode
1	Macro continues normal operation when the device is halted in debug mode

52. Electrical Characteristics (85°C)

52.1. Absolute Maximum Electrical Characteristics

Absolute maximum ratings for 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 52-1. MCU Absolute Maximum Electrical Characteristics

Ambient temperature under bias	-40°C to +85°C
Storage temperature	-65°C to +150°C
Voltage on VDD / VDDIO / VDDREG with respect to VSS	-0.3V to +4.0V
Voltage on any pin that is not 5V tolerant, with respect to VSS [3, 4]	-0.4V to (VDDIO+0.4V)
Voltage on any 5V tolerant pin with respect to VSS when VDDIO ≥ 1.9V [3, 4]	-0.4V to 5.5V
Voltage on any 5V tolerant pin with respect to VSS when VDDIO < 1.9V [3, 4]	-0.4V to (VDDIO+0.4V)
Voltage on USBHS PHY D+ or D- pin with respect to VSS	-0.4V to 5.3V
Voltage on USBHS PHY VBUS with respect to VSS	-0.3V to +5.25V
Voltage on USBFS D+ or D- pin with respect to VSS	-0.4V to (VDDIO+0.4V)
Voltage on VREFx with respect to VSS	-0.4V to (AVDD+0.4V)
Maximum current out of VSS pin(s)	190 mA
Maximum current into VDDIOx pin(s) [2]	190 mA
Maximum DC output current sourced/sunk by any Low-Current Mode I/O pin (4x Drive Strength) [5, 6]	38 mA
Maximum DC output current sourced/sunk by any High-Current Mode I/O pin (8x Drive Strength) [5, 6]	64 mA
Maximum DC current sourced/sunk by any port cluster [7] .	190 mA
Maximum Junction Temperature	+105°C
ESD qualification:	
Human Body Model (HBM) per JESD22-A114	±4000 V
Charged Device Model (CDM) (ANSI/ESD STM 5.3.1)	±1000 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. Maximum allowable current is a function of device maximum power dissipation.
3. See Package and Pinout for the 5V tolerant pins.
4. When applying higher or lower voltage than those specified on IO pins, please refer to DI_19 / DI_21 to respect the maximum injection current specification.
5. See the I/O Drive Strength table for the I/O pin Drive Strength.
6. This maximum value is specific to continuous DC current. For IO pin AC current characteristics, refer to the I/O Pin Electrical Specifications.
7. A cluster is a group of GPIO’s bounded by power/ground pins. Refer to the following table *I/O Clusters*.

Table 52-2. I/O Clusters

I/O Cluster	Port Pads
1	PA00, PA01, PA02, PA03
2	PA20, PA21, PA22, PA23, PA13, PA14, PA15, PA16, PA17, PA24
3	PA25, PA26, PA27, PA04, PA05, PA06, PA07, PA08, PA18, PA28
4	PA19, PA09, PA10, PA11, PA12, PB00, PB01, PB11, PB12, PB18, PB19, PB20, PB02, PB03

Table 52-2. I/O Clusters (continued)

I/O Cluster	Port Pads
5	PB04, PB05, PB13, PB06, PB14, PB15
6	PB21, PB22, PB23, PB24, PB25, PB26, PB16
7	PB17, PB07, PB08, PB09, PB10, PC00, PC01, PC02, PC03, PC04
8	PC21, PC22, PC14, PC23, PC05, PC06, PC07, PC24, PC25
9	PC26, PC27, PC09, PC15, PC28, PC29, PC16, PC17, PC18, PC10
10	PC11, PC12, PC13, PC19, PC20
11	PD00, PD01, PD02, PD13, PD14, PD15, PD16
12	PD17, PD18, PD19, PD20, PD21
13	PD12, PD05, PD06, PD07, PD08, PD09, PD10, PD11

52.2. CPU Electrical Characteristics

Table 52-3. Operating Frequency Versus Voltage

Param. No.	VDDIOx, AVDD, VDDREG Range	Temp. Range (in °C)	Max CPU Frequency	Comments
DC_5	1.71V to 3.63V	-40°C to +85°C	FCLK_1 MHz	Industrial

Table 52-4. CPU Thermal Operating Conditions

Rating	Symbol	Min.	Typ.	Max.	Unit
Industrial Temperature Devices Operating Ambient Temperature Range Operating Junction Temperature Range	TA TJ	-40 —	—	85 105	°C
Power Dissipation: **Internal Chip Power Dissipation:** PINT = (VDDIOx x (IDD –∑ IOHVDDIOx)) + (VUSB3V3 x IDDUSB) + (AVDD x (IDDANA - ∑ IOHAVDD)) + (VDDREG x IDDREG) **I/O Pin Power Dissipation:** PI/O = ∑ ((VDDIOx – VOHVDDIOx) x IOHVDDIOx) + ∑ (VOL x IOLVDDIOx) + ∑ ((AVDD – VOHAVDD) x IOHAVDD) + ∑ (VOL x IOLAVDD)	PD	PINT + PI/O			W
Maximum Allowed Power Dissipation	PDMAX	(TJ – TA)/θJA			W

Table 52-5. Thermal Packaging Characteristics (1)

Characteristics	Symbol	Typ.	Max.	Unit
Thermal Resistance, 144-pin TQFP (16x16x1 mm) Package	θJA	17.1	—	°C/W
Thermal Resistance, 144-pin TFBGA (10x10x1.2 mm) Package	θJA	28	—	°C/W
Thermal Resistance, 100-pin eTQFP (14x14x1 mm) Package	θJA	18.5	—	°C/W
Thermal Resistance, 64-pin QFN (9x9x0.9 mm) Package	θJA	18.6	—	°C/W
Thermal Resistance, 64-pin TQFP (10x10x1 mm) Package	θJA	21.3	—	°C/W

Note:
1. Junction to ambient thermal resistance, Theta-JA (θJA) numbers are achieved by package simulations.

52.3. Power Supply

Table 52-6. Power Supply DC Electrical Specifications

DC CHARACTERISTICS			Standard Operating Conditions: VDDREG=VDDIO=AVDD 1.71V to 3.63V (unless otherwise stated) **Operating temperature:** **-40°C ≤ TA ≤ +85°C for Industrial**				
Param. No.	**Symbol**	**Characteristics**	**Min.**	**Typ.**	**Max.**	**Units**	**Conditions**
REG_5	VDDIOx_CIN (4)	VDDIOx Input Bypass parallel Capacitor pair	33	—	—	μF	Bulk Ceramic or solid Tantalum with ESR <0.5Ω
			100	—	—	nF	Ceramic X7R with ESR <0.5Ω on all VDDIOx pins
REG_6	VDDREG_CIN (4)	VDDREGx Input Bypass parallel Capacitor pair	33	—	—	μF	Bulk Ceramic or solid Tantalum with ESR <0.5Ω
			100	—	—	nF	Ceramic X7R with ESR <0.5Ω on all VDDIOx pins
REG_10	VDDUSB_CIN (4)	USB Power pin bypass capacitance	4.7	—	—	μF	Required VDDUSB power pin paralell bypass capacitors
			0.1	—	—	μF	
REG_17	AVDD_CIN (4)	AVDD Input Bypass parallel Capacitor pair	10	—	—	μF	Bulk Ceramic or solid Tantalum with ESR <0.5Ω
			100	—	—	nF	Ceramic X7R with ESR <0.5Ω
REG_23	AVDD_LEXT (1)	AVDD series Ferrite Bead DCR (DC Resistance)	—	—	0.15	Ω	≥1k Ω @ 100 MHz
REG_25		Ferrite Bead current Rating	500	—	—	mA	—
REG_37	VDDIOx (2)	VDDIO Input Voltage Range	1.71	3.3	3.63	V	—
REG_39	AVDD (2)	AVDD Input Voltage Range	1.71	3.3	3.63	V	—
REG_40	VDDREG (3)	VDDREG Input Voltage Range	1.71	3.3	3.63	V	—
REG_42	VDDUSB	VDDUSB Input Voltage Range	3	—	3.6	V	—
REG_42A	IDDUSB	VUSB3V3 max current	—	—	8	mA	—
REG_43	SVDD_R	VDD Rise Ramp Rate to Ensure Internal Power-on Reset Signal	0.00000033	—	0.18	V/μs	Failure to meet this specification may lead to start-up or unexpected behaviors
REG_45	VPOR	Power-on Reset	1.43	—	—	V	VDD Power-down

Table 52-6. Power Supply DC Electrical Specifications (continued)

DC CHARACTERISTICS			Standard Operating Conditions: VDDREG=VDDIO=AVDD 1.71V to 3.63V (unless otherwise stated) **Operating temperature:** **-40°C ≤ TA ≤ +85°C for Industrial**				
Param. No.	Symbol	Characteristics	Min.	Typ.	Max.	Units	Conditions
REG_47	VDDIO / AVDD BOR [5]	VDDIO / AVDD Brown-Out Reset Thresholds	1.63	—	1.7	V	BOR_TRIP_VDDx = 0x0 [6] HYST_BOR_VDDx = 0x0
			2.11	—	2.24	V	BOR_TRIP_VDDx = 0x1 HYST_BOR_VDDx = 0x0
			2.49	—	2.68	V	BOR_TRIP_VDDx = 0x2 HYST_BOR_VDDx = 0x0
			2.75	—	2.97	V	BOR_TRIP_VDDx = 0x3 HYST_BOR_VDDx = 0x0
			1.61	—	1.7	V	BOR_TRIP_VDDx = 0x0 (6) HYST_BOR_VDDx = 0x1
			2.05	—	2.24	V	BOR_TRIP_VDDx = 0x1 HYST_BOR_VDDx = 0x1
			2.41	—	2.68	V	BOR_TRIP_VDDx = 0x2 HYST_BOR_VDDx = 0x1
			2.62	—	2.97	V	BOR_TRIP_VDDx = 0x3 HYST_BOR_VDDx = 0x1
REG_49	VDDREG BOR [5]	VDDREG Brown-Out Reset Thresholds	1.63	—	1.7	V	HYST_BOR_VDDREG = 0x0 [6]
			1.61	—	1.7	V	HYST_BOR_VDDREG = 0x1 [6]
REG_50	VDDUSB BOR [5]	VDDUSB Brown-Out Reset Thresholds	2.75	—	2.97	V	—
REG_51	VDDIO / AVDD / VDDREG DCBOR [7]	VDDIO / AVDD / VDDREG Duty Cycled BOR Thresholds	1.44	—	—	V	BOR_TRIP = 0x0 BOR_HYS = 0x0
			1.88	—	—	V	BOR_TRIP = 0x1 BOR_HYS = 0x0
			2.23	—	—	V	BOR_TRIP = 0x2 BOR_HYS = 0x0
			2.43	—	—	V	BOR_TRIP = 0x3 BOR_HYS = 0x0
			1.41	—	—	V	BOR_TRIP = 0x0 BOR_HYS = 0x1
			1.81	—	—	V	BOR_TRIP = 0x1 BOR_HYS = 0x1
			2.14	—	—	V	BOR_TRIP = 0x2 BOR_HYS = 0x1
			2.27	—	—	V	BOR_TRIP = 0x3 BOR_HYS = 0x1

Table 52-6. Power Supply DC Electrical Specifications (continued)

DC CHARACTERISTICS			Standard Operating Conditions: VDDREG=VDDIO=AVDD 1.71V to 3.63V (unless otherwise stated) **Operating temperature:** **-40°C ≤ TA ≤ +85°C for Industrial**				
Param. No.	Symbol	Characteristics	Min.	Typ.	Max.	Units	Conditions
REG_52	LVD [8]	VDDIO Low Voltage Detector Thresholds	1.66	—	1.76	V	LVD.LEVEL = 0x0
			1.74	—	1.84	V	LVD.LEVEL = 0x1
			1.82	—	1.92	V	LVD.LEVEL = 0x2
			1.96	—	2.07	V	LVD.LEVEL = 0x3
			2.06	—	2.18	V	LVD.LEVEL = 0x4
			2.11	—	2.24	V	LVD.LEVEL = 0x5
			2.17	—	2.30	V	LVD.LEVEL = 0x6
			2.25	—	2.37	V	LVD.LEVEL = 0x7
			2.32	—	2.44	V	LVD.LEVEL = 0x8
			2.43	—	2.55	V	LVD.LEVEL = 0x9
			2.63	—	2.77	V	LVD.LEVEL = 0xA
			2.72	—	2.87	V	LVD.LEVEL = 0xB
			2.82	—	2.97	V	LVD.LEVEL = 0xC
			2.93	—	3.08	V	LVD.LEVEL = 0xD
			3.24	—	3.40	V	LVD.LEVEL = 0xE
			3.54	—	3.71	V	LVD.LEVEL = 0xF
REG_53	TRST	External RESET valid active pulse width	2	—	—	μs	Minimum reset active time to guarantee MCU reset

Notes:
1. Ferrite Bead ISAT(min) ≥ (IDDANA(max) * 1.15).
2. VDDIOx and AVDD must be at the same voltage level.
3. VDDREG voltage must be equal or lower than VDDIOx.
4. All bypass caps should be located immediately adjacent to pin(s) and on the same side of the PCB as the MCU, or in the case of BGA packages, directly below the power pads and direct adjacent to the fan-out vias. Each primary power supply group VDDIO and AVDD should have one bulk capacitor and all power pins everywhere a 100 nF bypass cap.
5. Voltages below the Min BOR threshold will result in a device reset, except for the BOR_VDDUSB that can cause an interrupt. Voltages above the Max BOR threshold will allow the device starting-up.
6. Overall functional device operation at VBORMIN < VDD < VDDMIN is guaranteed, but not characterized. Device will function with degraded performances below VDDMIN.
7. Voltages below the Min DCBOR threshold will result in a device reset.
8. Voltages below the Min LVD threshold will result in a falling detection. Voltages above the Max LVD threshold will result in a rising detection. Enabling the LVD when VDDIO is between Min / Max thresholds may result in unexpected behavior.

52.4. MCU Active Power

Table 52-7. CPU Active Current Consumption DC Electrical Specifications

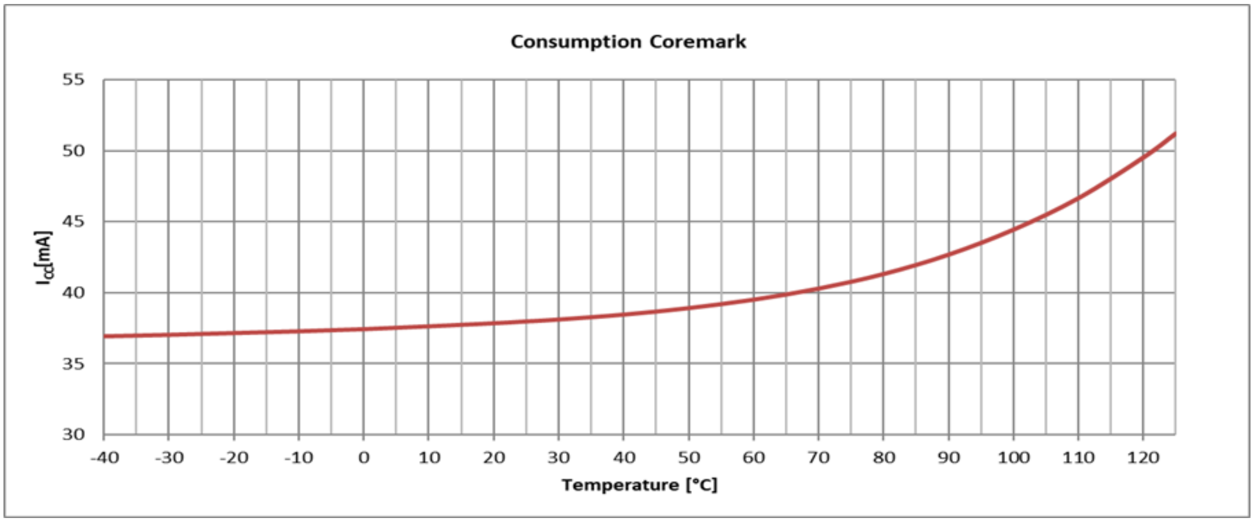
DC CHARACTERISTICS					Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.6V (unless otherwise stated) **Operating temperature:** **-40°C ≤ T_A ≤ +85°C for Industrial**			
Param. No.	Symbol	Characteristics	Clock/Freq	Power Pin	Typ. [1]	Max.	Units	Conditions
APWR_1	I_{DD_ACTIVE} [2,3]	MCU IDD in active mode	PLL_120_MHz	VDDIO = AVDD = 3.3V	5	5	mA	PLL runs at 120 MHz, uses XOSC with 48 MHz crystal as reference clock
APWR_3				VDDREG = 3.3V	34	78	mA	
APWR_5			DFLL _48_MHz	VDDIO = AVDD = 3.3V	3	3	mA	DFLL in open loop
APWR_7				VDDREG = 3.3V	15	57	mA	
APWR_9			XOSC _48_MHz	VDDIO = AVDD = 3.3V	4	4	mA	48 MHz crystal
APWR_11				VDDREG = 3.3V	15	57	mA	

Notes:
1. Typical values at 25°C only.
2. Conditions:
   - RTC running on ULP32K for Coremark® ticks count
   - No other peripheral modules are operating (i.e., all peripherals except RTC inactive)
   - APB Peripheral bus clocks: default settings after reset
   - CMCC Cache disabled
   - MCU is running on Flash with automatic wait state
   - VREG_USB disabled
   - I/Os are inactive input mode with input trigger disabled
   - All clock generation sources disabled unless otherwise specified
   - WDT, CFD Clock Fail Detect disabled
3. MCU Running CoreMark Test Suite.

Operating Conditions:

- VDDIO = 3.3V
- PLL_120_MHz

Figure 52-1. Power Consumption over Temperature in Active Mode (Typical values for guidance only, not tested)



52.5. MCU Idle Power

Table 52-8. MCU Idle Current Consumption DC Electrical Specifications

DC CHARACTERISTICS					Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.6V (unless otherwise stated) **Operating temperature:** **-40°C ≤ T_A ≤ +85°C for Industrial**			
Param. No.	Symbol	Characteristics	Clock/Freq	Power Pin	Typ. [(1)]	Max.	Units	Conditions
IPWR_1	I_{DD_IDLE} [(2)]	MCU IDD in Idle mode	PLL_120_MHz	VDDIO = AVDD = 3.3V	2	2	mA	PLL runs at 120 MHz, uses XOSC with 48 MHz crystal as reference clock
IPWR_3				VDDREG = 3.3V	18	60	mA	
IPWR_5			DFLL _48_MHz	VDDIO = AVDD = 3.3V	2	2	mA	DFLL in open loop
IPWR_7				VDDREG = 3.3V	7	48	mA	
IPWR_9			XOSC _48_MHz	VDDIO = AVDD = 3.3V	2	2	mA	48 MHz crystal
IPWR_11				VDDREG = 3.3V	7	49	mA	

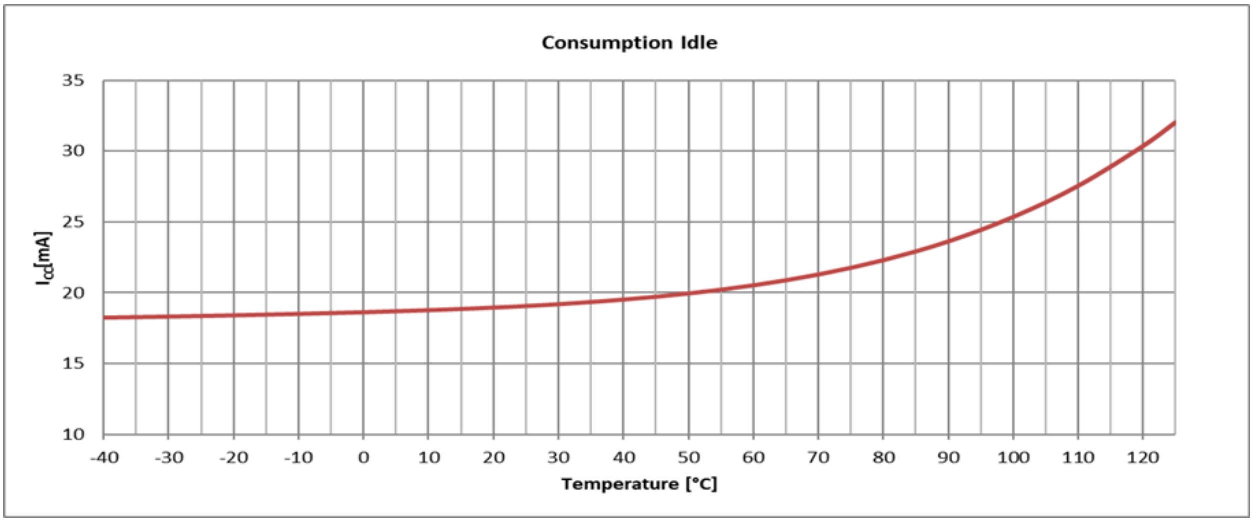
Notes:

1. Typical values at 25°C only.
2. Conditions:
   - No peripheral modules are operating (i.e., all peripherals are inactive)
   - APB peripheral bus clocks: default settings after reset
   - CMCC Cache disabled
   - MCU is running on Flash with automatic wait state
   - VREG_USB disabled
   - I/Os are inactive input mode with input trigger disabled
   - All clock generation sources disabled unless otherwise specified
   - WDT, CFD Clock Fail Detect disabled

Operating Conditions

- VDDIO = 3.3V
- PLL_120_MHz

Figure 52-2. Power Consumption over Temperature in Idle0 Mode (Typical values for guidance only, not characterized over Process or voltage)



52.6. MCU Standby Power

Table 52-9. CPU Standby Current Consumption DC Electrical Specifications

DC CHARACTERISTICS				Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated) **Operating temperature:** **-40°C ≤ TA ≤ +85°C for Industrial**			
Param. No.	Symbol	Characteristics	Power Pin	Typ. [1]	Max.	Units	Conditions
SPWR_1	IDD_STANDBY [2]	MCU IDD in Standby mode w/VDDCORE_x = 0.8V	VDDIO = AVDD = 3.3V	0.5	0.6	mA	Full System RAM retained (PM.STDBYCFG.RAMCFG = 0x0), RAM Low- Power mode disabled (PM.STDBYCFG.LPRAM = 0x0), Voltage Regulators Low Voltage enabled (SUPC.VREGCTRL.LVSTDBY = 0x1)
SPWR_3			VDDREG = 3.3V	0.9	15	mA	
SPWR_5			VDDIO = AVDD = 3.3V	0.5	0.6	mA	32 Kb System RAM retained (PM.STDBYCFG.RAMCFG = 0x1), RAM Low- Power mode disabled (PM.STDBYCFG.LPRAM = 0x0), Voltage Regulators Low Voltage enabled (SUPC.VREGCTRL.LVSTDBY = 0x1)
SPWR_7			VDDREG = 3.3V	0.8	12.7	mA	
SPWR_9			VDDIO = AVDD = 3.3V	0.5	0.6	mA	Full System RAM retained (PM.STDBYCFG.RAMCFG = 0x0), RAM Low- Power mode enabled (PM.STDBYCFG.LPRAM = 0x1), Voltage Regulators Low Voltage enabled (SUPC.VREGCTRL.LVSTDBY = 0x1)
SPWR_11			VDDREG = 3.3V	0.8	12.9	mA	
SPWR_13		MCU IDD in Standby mode w/VDDCORE_x = 1.2V	VDDIO = AVDD = 3.3V	0.5	0.7	mA	Full System RAM retained (PM.STDBYCFG.RAMCFG = 0x0), RAM Low- Power mode disabled (PM.STDBYCFG.LPRAM = 0x0), Voltage Regulators Low Voltage disabled (SUPC.VREGCTRL.LVSTDBY = 0x0)
SPWR_15			VDDREG = 3.3V	1.4	41.7	mA	
SPWR_17			VDDIO = AVDD = 3.3V	0.5	0.7	mA	32 Kb System RAM retained (PM.STDBYCFG.RAMCFG = 0x1), RAM Low- Power mode disabled (PM.STDBYCFG.LPRAM = 0x0), Voltage Regulators Low Voltage disabled (SUPC.VREGCTRL.LVSTDBY = 0x0)
SPWR_19			VDDREG = 3.3V	1.3	34.7	mA	
SPWR_21			VDDIO = AVDD = 3.3V	0.5	0.7	mA	Full System RAM retained (PM.STDBYCFG.RAMCFG = 0x0), RAM Low- Power mode enabled (PM.STDBYCFG.LPRAM = 0x1), Voltage Regulators Low Voltage disabled (SUPC.VREGCTRL.LVSTDBY = 0x0)
SPWR_23			VDDREG = 3.3V	1.4	35.2	mA	

Table 52-9. CPU Standby Current Consumption DC Electrical Specifications (continued)

DC CHARACTERISTICS				Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated) **Operating temperature:** **-40°C ≤ TA ≤ +85°C for Industrial**			
Param. No.	Symbol	Characteristics	Power Pin	Typ. (1)	Max.	Units	Conditions

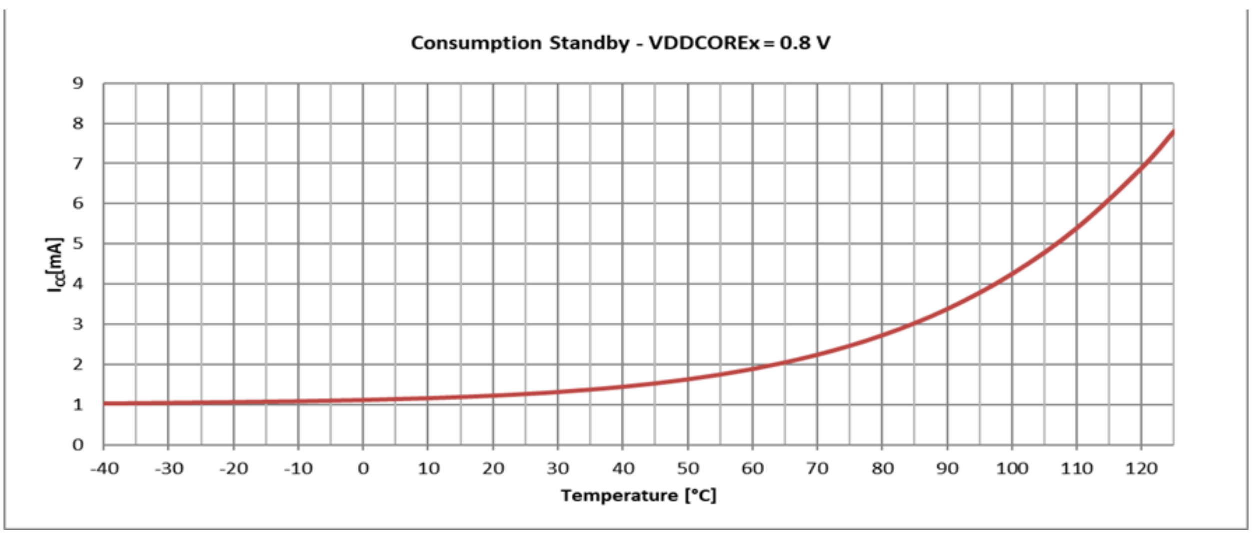
Notes:

1. Typical values at 25°C only.
2. Conditions:
   - No peripheral modules are operating, unless specified (i.e., all peripherals inactive)
   - APB Peripheral bus clocks: default settings after reset
   - CMCC Cache disabled
   - MCU is running on Flash with automatic wait state
   - VREG_USB disabled
   - I/Os are inactive input mode with input trigger disabled
   - All clock generation sources disabled unless otherwise specified
   - WDT, CFD Clock Fail Detect disabled

Operating Conditions:

- VDDIO = 3.3V
- VDDCORE = 0.8V

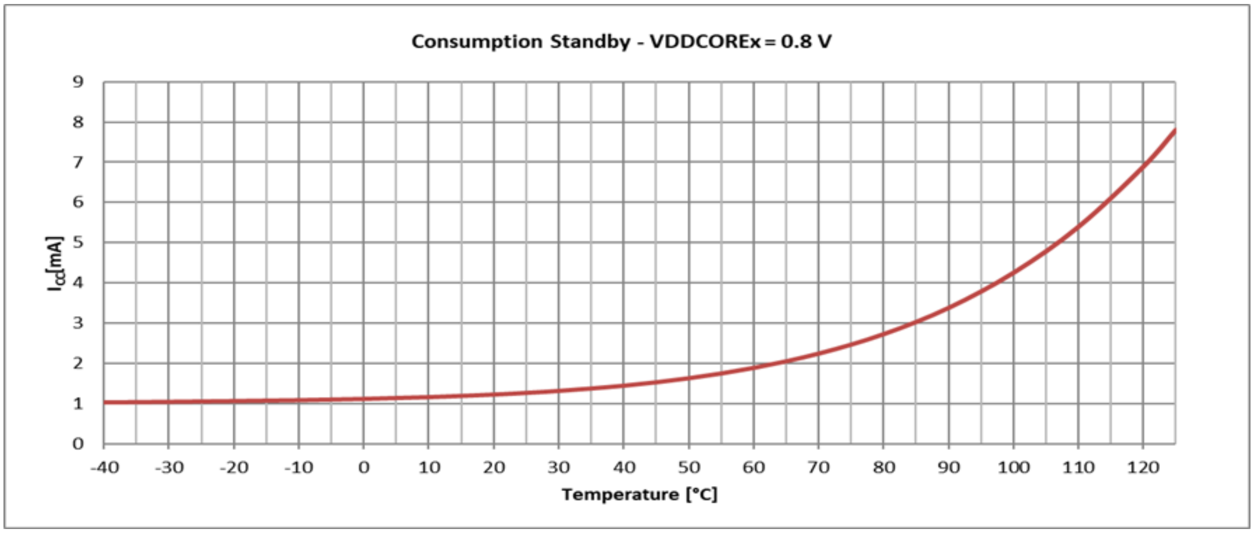
Figure 52-3. Power Consumption over Temperature in Standby Mode, VDDCORE = 0.8V (Typical values for guidance only, not characterized over Process /Voltage)



Operating Conditions:

- VDDIO = 3.3V
- VDDCORE = 1.2V

Figure 52-4. Power Consumption over Temperature in Standby Mode, VDDCORE = 1.2V (Typical values for guidance only, not characterized over Process /Voltage)



52.7. MCU Hibernate Power

Table 52-10. CPU Hibernate Current Consumption DC Electrical Specifications

DC CHARACTERISTICS				Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated) **Operating temperature:** **-40°C ≤ TA ≤ +85°C for Industrial**			
Param. No.	Symbol	Characteristics	Power Pin	Typ. [(1)]	Max.	Units	Conditions
HPWR_1	IDD_HIBERNATE [(2)]	MCU IDD in Hibernate mode	VDDIO = AVDD = 3.3V	108	150	µA	Full System RAM retained (PM.HIBCFG.RAMCFG = 0x0), RAM Low- Power mode disabled (PM.HIBCFG.LPRAM = 0x0), Voltage Regulators Low Voltage enabled (SUPC.VREGCTRL.LVHIB = 0x1)
HPWR_3			VDDREG = 3.3V	0.3	2.9	mA	
HPWR_5			VDDIO = AVDD = 3.3V	108	150	µA	32 Kb System RAM retained (PM.HIBCFG.RAMCFG = 0x1), RAM Low- Power mode disabled (PM.HIBCFG.LPRAM = 0x0), Voltage Regulators Low- Voltage enabled (SUPC.VREGCTRL.LVHIB = 0x1)
HPWR_7			VDDREG = 3.3V	0.2	0.9	mA	
HPWR_9			VDDIO = AVDD = 3.3V	108	150	µA	Full System RAM retained (PM.HIBCFG.RAMCFG = 0x0), RAM Low- Power mode enabled (PM.HIBCFG.LPRAM = 0x1), Voltage Regulators Low Voltage enabled (SUPC.VREGCTRL.LVHIB = 0x1)
HPWR_11			VDDREG = 3.3V	0.2	1.1	mA	

Table 52-10. CPU Hibernate Current Consumption DC Electrical Specifications (continued)

DC CHARACTERISTICS				Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated) **Operating temperature:** **-40°C ≤ TA ≤ +85°C for Industrial**			
Param. No.	Symbol	Characteristics	Power Pin	Typ. [1]	Max.	Units	Conditions

Notes:

1. Typical values at 25°C only.
2. Conditions:
   - No peripheral modules are operating, unless specified (i.e., all peripherals are inactive)
   - APB peripheral bus clocks: default settings after reset
   - CMCC cache disabled
   - MCU is running on Flash with automatic wait state
   - VREG_USB disabled
   - I/Os are inactive input mode with input trigger disabled
   - All clock generation sources disabled unless otherwise specified
   - WDT, CFD Clock Fail Detect disabled

52.8. MCU Backup Power

Table 52-11. CPU Backup Current Consumption DC Electrical Specifications

DC CHARACTERISTICS				Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated) **Operating temperature:** **-40°C ≤ TA ≤ +85°C for Industrial**			
Param. No.	Symbol	Characteristics	Power Pin	Typ (1)	Max.	Units	Conditions
BPWR_1	IDD_BACKUP [2]	MCU IDD in Backup mode powered from VDDIOx	VDDIO = AVDD = 3.3V	10	122	µA	RTC running on XOSC32K, TRAM enabled
BPWR_3			VDDREG = 3.3V	0.5	1.1	µA	

Notes:

1. Typical values at 25°C only.
2. Conditions:
   - No peripheral modules are operating (i.e., all peripherals inactive)
   - APB peripheral bus clocks: default settings after reset
   - CMCC cache disabled
   - MCU is running on Flash with automatic wait state
   - VREG_USB disabled
   - I/Os are inactive input mode with input trigger disabled
   - All clock generation sources disabled unless otherwise specified
   - WDT, CFD Clock Fail Detect disabled

52.9. MCU Off Power

Table 52-12. MCU Off Current Consumption DC Electrical Specifications

DC CHARACTERISTICS				Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.6V (unless otherwise stated) **Operating temperature:** **-40°C ≤ T_A ≤ +85°C for Industrial**			
Param. No.	Symbol	Characteristics	Power Pin	Typ. (1)	Max.	Units	Conditions
OPWR_1	I_{DD_OFF} (2)	MCU IDD in Off mode powered from VDDIOx	VDDIO = AVDD = 3.3V	1.4	8.4	µA	In Off mode, the device is entirely powered-off. (SLEEPCFG.SLEEPMODE = OFF), and subsequent execution of the WFI instruction. **Note:** This mode is left by pulling the $\overline{\text{Reset}}$ pin low, or when a power Reset is done.
OPWR_3			VDDREG = 3.3V	0	0.5	µA	

Notes:
1. Typical values at 25°C only.
2. Conditions:
   - All peripherals inactive unless otherwise specified
   - All clock generation sources disabled unless otherwise specified
   - All I/O pins configured as input pins pulled down or tied to GND

52.10. Wake-Up Timing

Table 52-13. Wake-Up Timing from Low-Power Modes AC Electrical Specifications

AC CHARACTERISTICS			Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.6V (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	—	0.334	—	µs	—
WKUP_7	WKUP_STDBY	Wake from Standby mode	—	22.5	—	µs	—
WKUP_11	WKUP_HIBER	Wake from Hibernate mode	—	612	—	µs	—
WKUP_13	WKUP_BCKUP	Wake from Backup mode	—	750	—	µs	—
WKUP_15	WKUP_OFF	Wake from Off mode	—	533	—	µs	—

Note:
1. VDD = 3.3V, CPU clock = 48 MHz, automatic wait states, Cache disabled, Temperature = 25°C.

52.11. Peripheral Active Power

Table 52-14. Peripheral Active Current DC Electrical Specifications

DC CHARACTERISTICS (1)			Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated) **Operating temperature:** **-40°C ≤ TA ≤ +85°C for Industrial**
Param. No.	Symbol	Characteristics	Conditions
MODULES/PERIPHERALS ACTIVE CURRENTS ≤ 1 mA			
PAI_51	AC	Analog Comparator Current	2 Comparators Enabled

Table 52-14. Peripheral Active Current DC Electrical Specifications (continued)

DC CHARACTERISTICS (1)			Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated) **Operating temperature:** **-40°C ≤ TA ≤ +85°C for Industrial**
Param. No.	Symbol	Characteristics	Conditions
PAI_52	BOR	BOR Current	—
PAI_53	CCL	CCL Current	—
PAI_54	EIC	EIC Current	—
PAI_55	EVSYS	EVSYS Current	—
PAI_56	FREQM	FREQM Current	—
PAI_57	SERCOM	SERCOM Current	SPI mode, baudrate = 20 MHz, internal loopback
PAI_58	TRNG	TRNG Current	—
PAI_59	WDT	WDT Current	—
MODULES/PERIPHERALS ACTIVE CURRENTS ≤ 3 mA			
PAI_100	ADC	Analog-to-Digital Converter Current	12-bit, Core 0 = 3.125 Msps. DMA enabled for data transfers
PAI_101	CAN-FD	CAN-FD Current	8 Mbps
PAI_102	I2S	I²S Current	I²S mode, 2 channels, Sampling Frequency = 192 kHz. DMA enabled for data transfers
PAI_103	PTC	PTC Current	Single Mutual Capacitance node, acquisition time = 10 ms
PAI_104	SQI	SQI Current	SDR mode, SQI frequency = 84 MHz, Load = 30 pF. DMA enabled for data transfers
PAI_105	TCC	TCC Current	—
PAI_106	XOSC	XOSC Current	48 MHz crystal, AGC Enabled
PAI_107	PCC	PCC Current	DMA enabled for data transfers
PAI_108	PDEC	PDEC Current	—
PAI_109	USB FS	Full-Speed USB	—
MODULES/PERIPHERALS ACTIVE CURRENTS ≤ 5 mA			
PAI_200	ETH	Ethernet Current	MII 100 Mbit/s. DMA enabled for data transfers
MODULES/PERIPHERALS ACTIVE CURRENTS ≤ 10 mA			
PAI_400	DMA	DMA Current	—
PAI_401	Flash	Flash Erase/Write	—
PAI_402	PLL	PLL Active Current	PLL output = 120 MHz
MODULES/PERIPHERALS ACTIVE CURRENTS ≤ 15 mA			
PAI_800	EBI	EBI Current	—
PAI_801	HSM	Hardware Security Module Current	DMA enabled for data transfers
PAI_802	SDMMC	SDMMC Active Current	SD frequency = 52 MHz, Load = 30 pF. DMA enabled for data transfers
MODULES/PERIPHERALS ACTIVE CURRENTS ≤ 50 mA			
PAI_1000	USB HS	High-Speed USB Current	DMA enabled for data transfers

Note:

1. These values are for worst case guidance only unless otherwise specified, actual values depend on user peripheral configuration and operating frequency.

52.12. I/O Pin Electrical Specifications

Table 52-15. I/O Pin AC/DC Electrical Specifications

AC - DC CHARACTERISTICS			Standard Operating Conditions: VDDREG=VDDIO=AVDD 1.71V to 3.63V (unless otherwise stated) **Operating temperature:** **-40°C ≤ TA ≤ +85°C for Industrial**				
Param. No.	**Symbol**	**Characteristics**	**Min.**	**Typ.**	**Max.**	**Units**	**Conditions**
DI_1	VIL	Input Low-Voltage I/O Pins	GND	—	0.2*VDDIO	V	VDDIO(min) -to- VDDIO(max)
DI_3	VIH	Input High-Voltage Non-5V tolerant I/O Pins	0.7*VDDIO	—	VDDIO	V	VDDIO(min) -to- VDDIO(max)
		Input High-Voltage I/O Pins 5V-tolerant	0.7*VDDIO	—	5.5V	V	VDDIO(min) -to- 5.5V
DI_5	VOL	4x Drive Strength I/O pins (Output Low)	—	—	0.4	V	VDDIO < 3.0V @ IOL = 3 mA [6]
							VDDIO ≥ 3.0V @ IOL = 7 mA [6]
		8x Drive Strength I/O pins (Output Low)					VDDIO < 3.0V @ IOL = 6 mA [6]
							VDDIO ≥ 3.0V @ IOL = 13 mA [6]
DI_9	VOH	4x Drive Strength I/O pins (Output High)	VDDIO-0.6V	—	—	V	VDDIO < 3.0V @ IOH = 5 mA [6]
							VDDIO ≥ 3.0V @ IOH = 11 mA [6]
		8x Drive Strength I/O pins (Output High)					VDDIO < 3.0V @ IOH = 9 mA [6]
							VDDIO ≥ 3.0V @ IOH = 19 mA [6]
DI_13	IIL	Input pin leakage current	-1	—	1	µA	GND ≤ VPIN ≤ VDDIO(max) (VPIN = Voltage present on Pin)
DI_14	IIL_5V	Input 5.5V tolerant pin leakage current	-1.1	—	3.5	µA	For VDDIO < 1.9V: GND ≤ VPIN ≤ VDDIO+3.6 For VDDIO ≥ 1.9V: 1.9V ≤ VPIN ≤ 5.5V (VPIN = Voltage present on Pin)
DI_15	RPDWN	Internal Pull-Down (DIR = OUT = 0, PULLEN = 1)	25	—	400	kΩ	VDDIOx(min) - VDDIO(max)
DI_17	RPUP	Internal Pull-Up (DIR = 0, OUT = PULLEN = 1)	-400	—	-30	kΩ	
DI_19	IICL	Input Low Injection Current	-5	—	—	mA	This parameter applies to all I/O pins.[1,4,5]
DI_21	IICH	Input High Injection Current	—	—	5	mA	This parameter applies to all pins, with the exception of 5V tolerant I/O pins[2,3,4,5]

Table 52-15. I/O Pin AC/DC Electrical Specifications (continued)

AC - DC CHARACTERISTICS			Standard Operating Conditions: VDDREG=VDDIO=AVDD 1.71V to 3.63V (unless otherwise stated) **Operating temperature:** **-40°C ≤ TA ≤ +85°C for Industrial**				
Param. No.	**Symbol**	**Characteristics**	**Min.**	**Typ.**	**Max.**	**Units**	**Conditions**
DI_23	∑IICT	Total Input Injection Current (sum of all I/O and control pins) Absolute value of | ∑IICT |	-20	—	20	mA	Absolute instantaneous sum of all ± input injection currents from all I/O pins. (| IICL | + | IICH |) ≤ ∑IICT
DI_25	TRISE	I/O pin Rise Time (DRVSTR = 4x)	—	—	11	ns	VDDIOx(min), CLOAD = 30 pF(max)
DI_27	TFALL	I/O pin Fall Time (DRVSTR = 4x)	—	—	10.1	ns	
DI_29	TRISE	I/O pin Rise Time (DRVSTR = 8x, SLEW = FAST)	—	—	6.3	ns	
		I/O pin Rise Time (DRVSTR = 8x, SLEW = SLOW4)	—	—	20.5	ns	
		I/O pin Rise Time (DRVSTR = 8x, SLEW = SLOW8)	—	—	36.7	ns	
		I/O pin Rise Time (DRVSTR = 8x, SLEW = SLOW12)	—	—	51.8	ns	
DI_31	TFALL	I/O pin Fall Time (DRVSTR = 8x, SLEW = FAST)	—	—	5.8	ns	
		I/O pin Fall Time (DRVSTR = 8x, SLEW = SLOW4)	—	—	23.3	ns	
		I/O pin Fall Time (DRVSTR = 8x, SLEW = SLOW8)	—	—	39.8	ns	
		I/O pin Fall Time (DRVSTR = 8x, SLEW = SLOW12)	—	—	56.7	ns	

Notes:

1. V_{IL} source < (GND - 0.3). Characterized but not tested.
2. V_{IH} source > (VDDIO + 0.3) for non-5V tolerant pins only.
3. Digital 5V tolerant pins do not have an internal high side Diode-to-VDDIO, and therefore, cannot tolerate any “positive” input injection current.
4. If the sum of all injection currents are > | $\sum_{IICT}$ | it can affect the ADC results by approximately 4 to 6 counts (i.e., V_{IH} Source > (VDDIO + 0.3) or V_{IL} source < (GND - 0.3)).
5. Any number and/or combination of I/O pins not excluded under I_{ICL} or I_{ICH} conditions are permitted provided the “absolute instantaneous” sum of the input injection currents from all pins do not exceed the specified $\sum_{IICT}$ limit. To limit the injection current the user must insert a resistor in series R_{SERIES}, (i.e. R_S), between the input source voltage and device pin. The resistor value is calculated according to the following:
   - For negative Input voltages less than (GND-0.3): R_S ≥ absolute value of | ((V_{IL} source - (GND - 0.3)) / I_{ICL}) |
   - For positive input voltages greater than (VDDIO +0.3): R_S ≥ ((V_{IH} source - (VDDIO +0.3))/ I_{ICH})
   - For Vpin voltages >VDDIO +0.3 and <GND-0.3 then R_S = the larger of the values calculated above

Table 52-16. I/O Drive Strength

Drive Strength	IO Port/Pad
4x	PA20, PA21, PA22, PA23, PA15, PA16, PA17, PA24, PA25, PA26, PA27, PA04, PA05, PA06, PA07, PA08, PA18, PA28, PA19, PA09, PA10, PA11, PA12, PB00, PB01, PB11, PB12, PB18, PB19, PB20, PB02, PB03, PB04, PB05, PB13, PB06, PB14, PB15, PB21, PB22, PB23, PB24, PB25, PB26, PB16, PB17, PB08, PC03, PC04, PC21, PC22, PC14, PC23, PC24, PC25, PC26, PC27, PC09, PC15, PC28, PC29, PC17, PC18, PD00, PD01, PD15, PD18, PD19, PD20, PD21
8x	PA00, PA01, PA02, PA03, PA13, PA14, PB07, PB09, PB10, PC00, PC01, PC02, PC05, PC06, PC07, PC16, PC10, PC11, PC12, PC13, PC19, PC20, PD02, PD13, PD14, PD16, PD17, PD12, PD05, PD06, PD07, PD08, PD09, PD10, PD11

Figure 52-5. IOL vs VOL, 4x Drive Strength I/O on Typical parts with VDDIO = 1.71 V

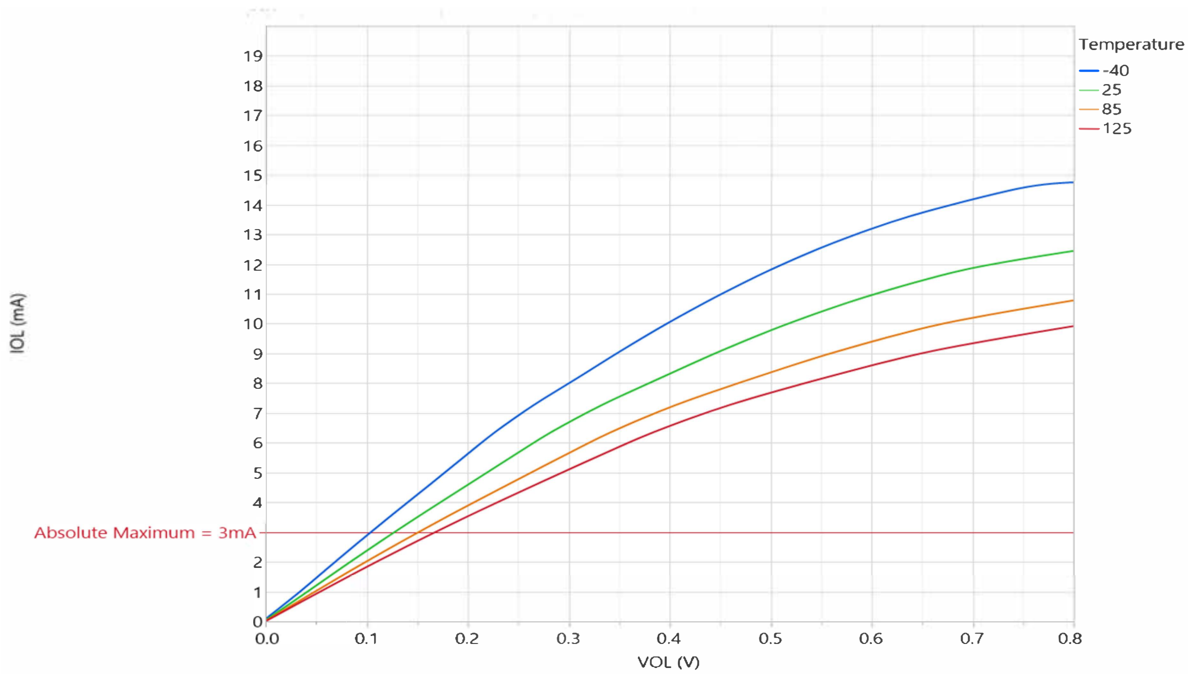


Figure 52-6. IOL vs VOL, 4x Drive Strength I/O on Typical parts with VDDIO = 3.0 V

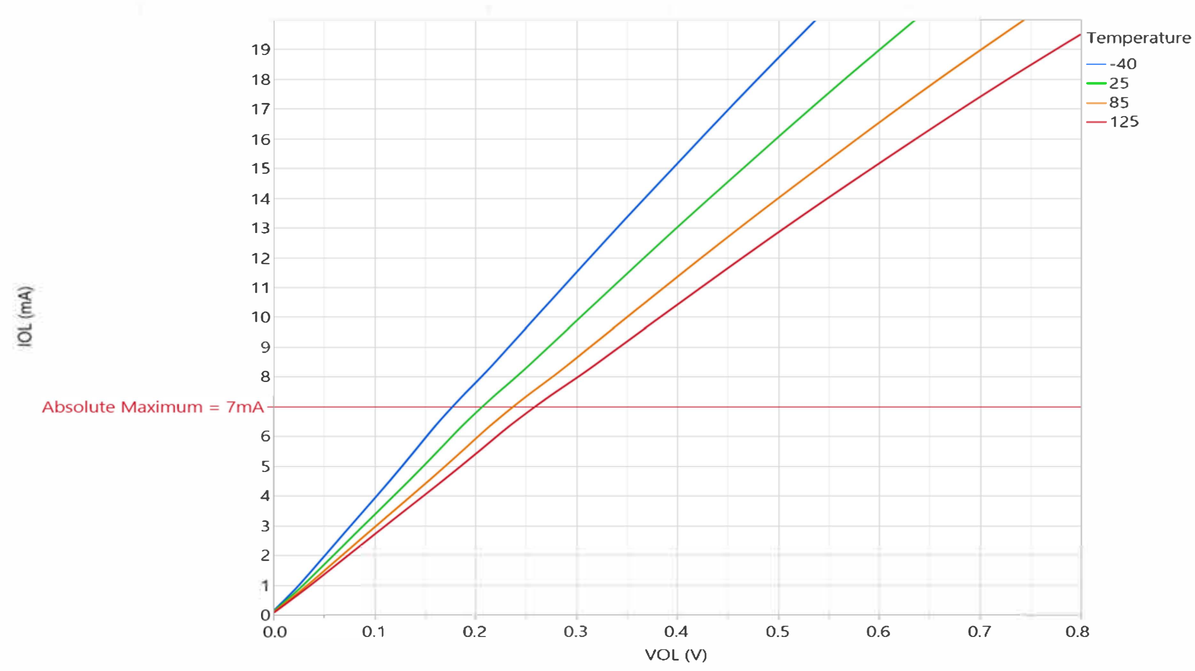


Figure 52-7. IOL vs VOL, 8x Drive Strength I/O on Typical parts with VDDIO = 1.71 V

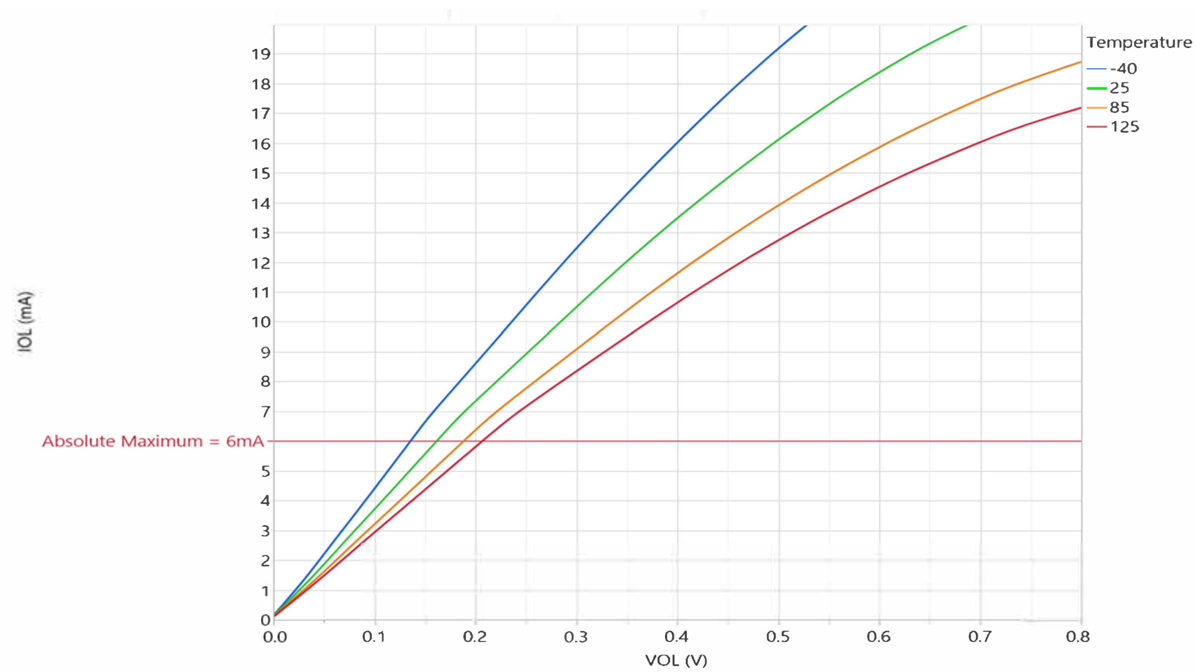


Figure 52-8. IOL vs VOL, 8x Drive Strength I/O on Typical parts with VDDIO = 3.0 V

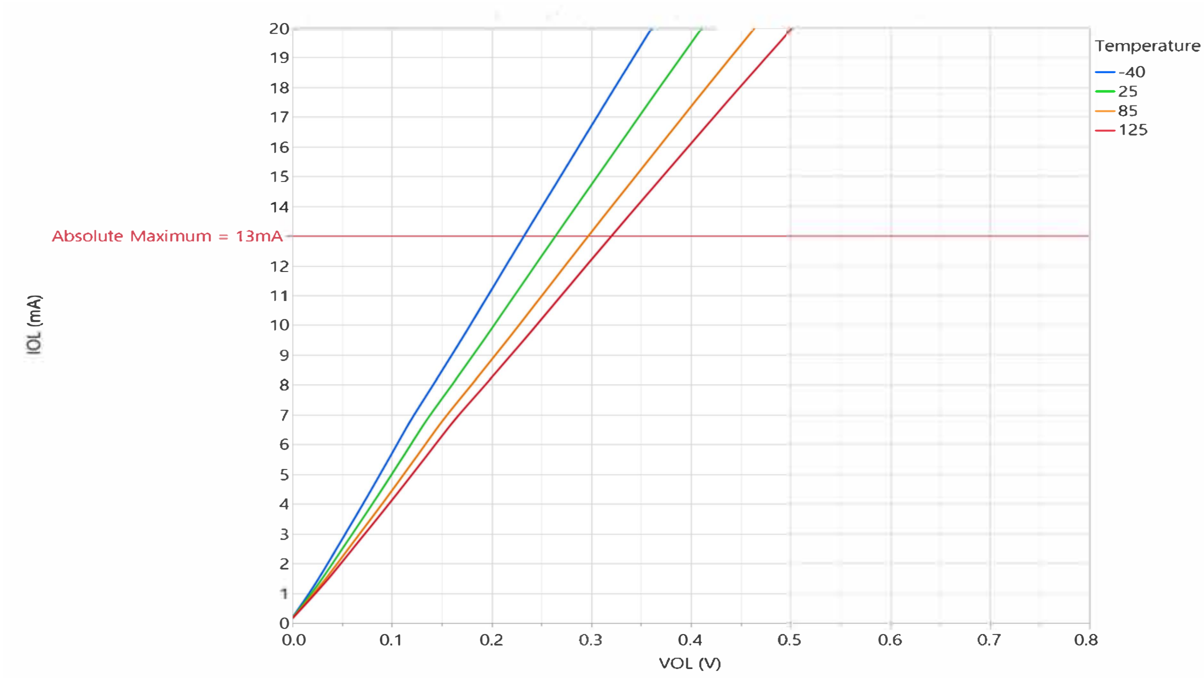


Figure 52-9. IOH vs VOH, 4x Drive Strength I/O on Typical parts with VDDIO = 1.71 V

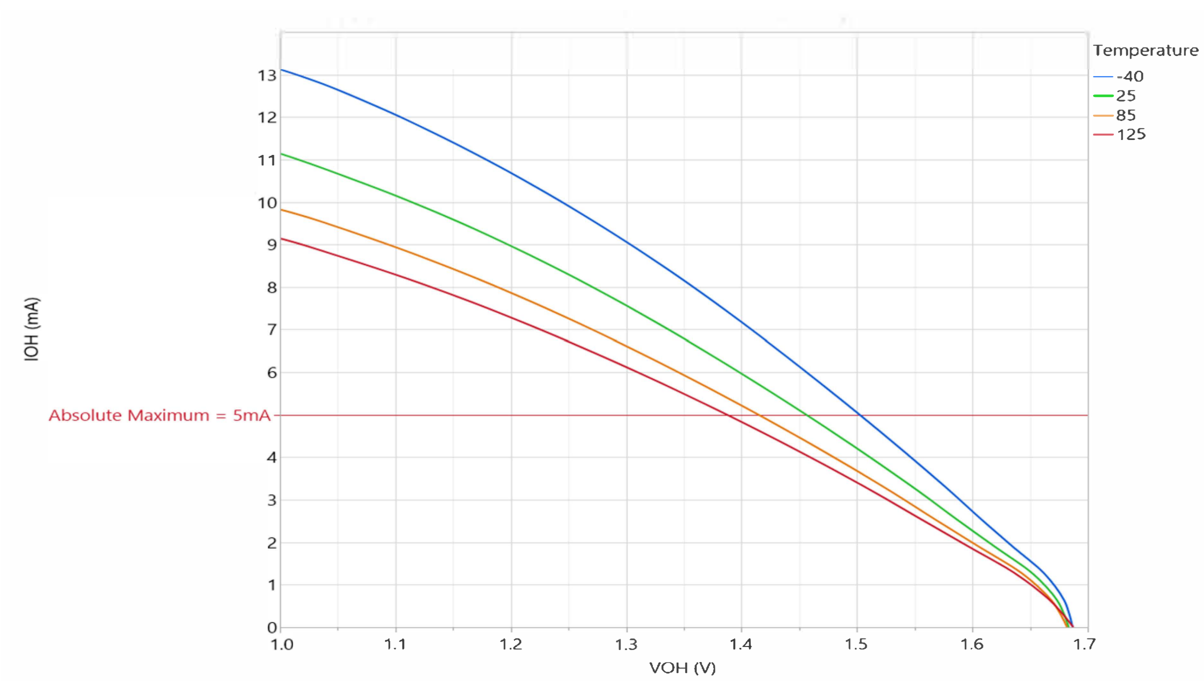


Figure 52-10. IOH vs VOH, 4x Drive Strength I/O on Typical parts with VDDIO = 3.0 V

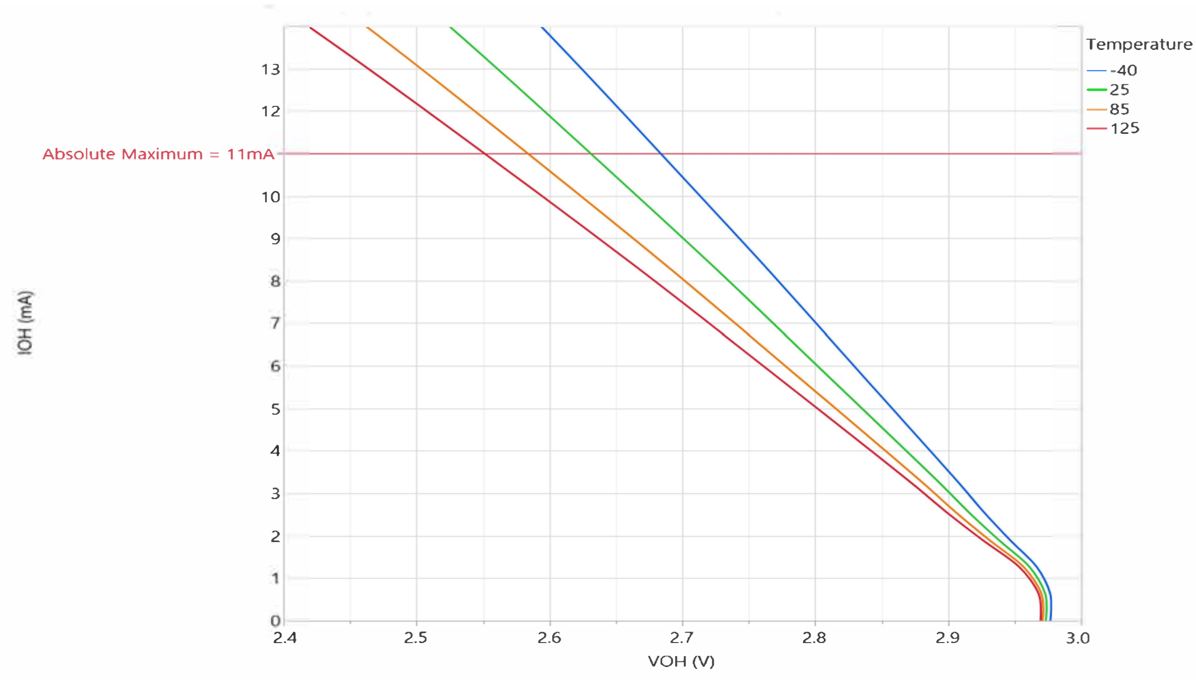


Figure 52-11. IOH vs VOH, 8x Drive Strength I/O on Typical parts with VDDIO = 1.71 V

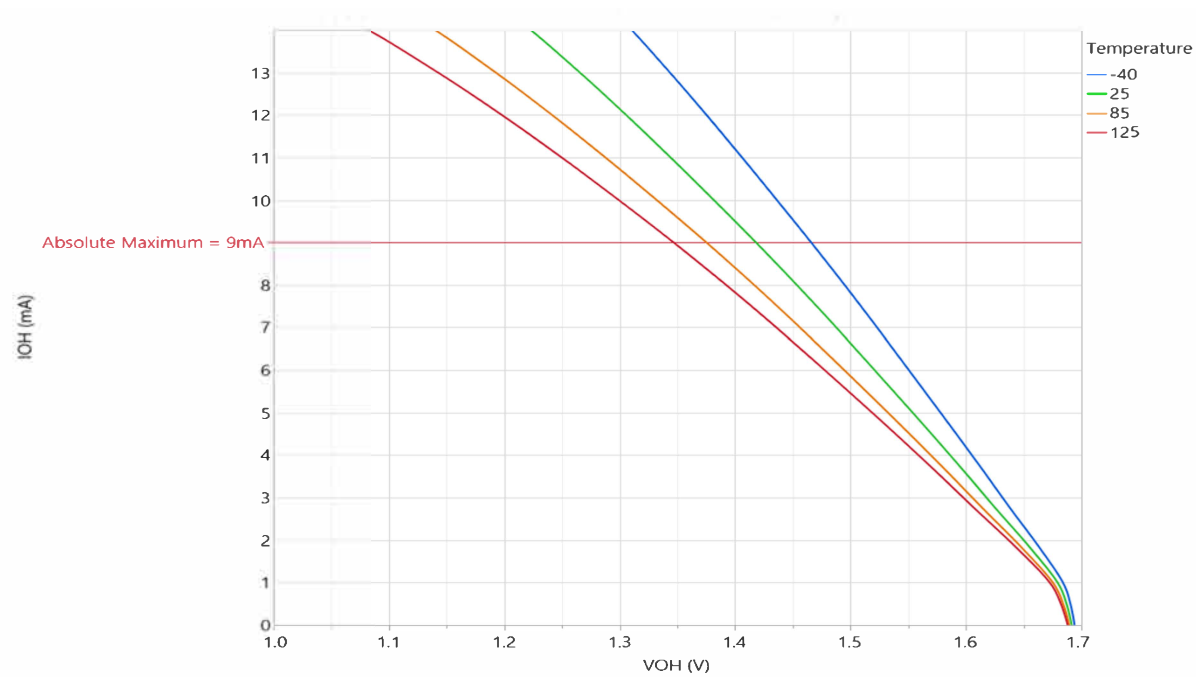
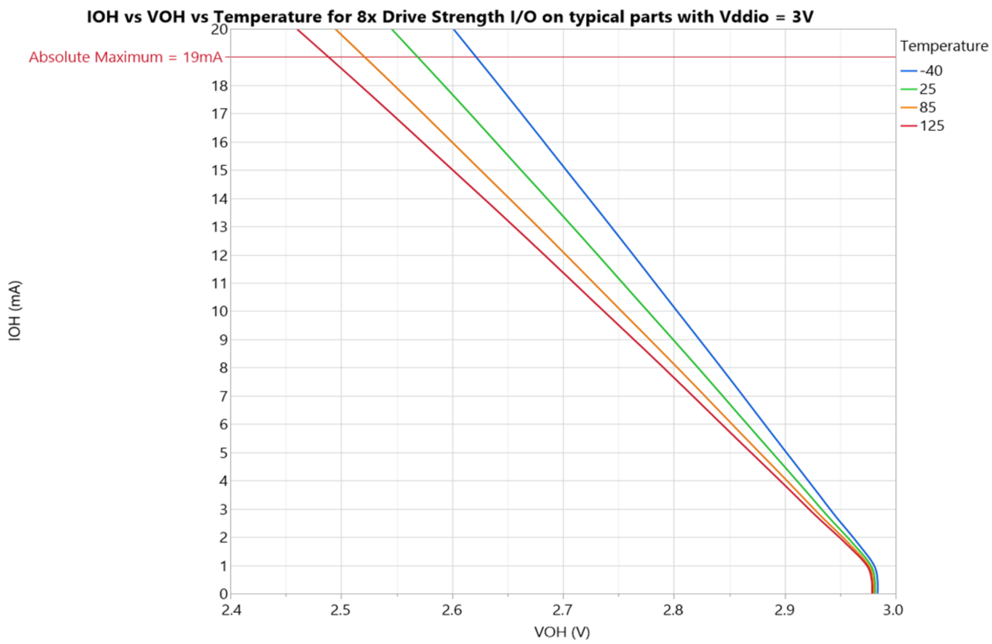


Figure 52-12. IOH vs VOH, 8x Drive Strength I/O on Typical parts with VDDIO = 3.0 V



52.13. Internal Voltage Reference Electrical Specifications

Table 52-17. Internal Voltage Reference DC Electrical Specifications

DC CHARACTERISTICS			Standard Operating Conditions: VDDREG=VDDIO=AVDD 1.71V to 3.63V (unless otherwise stated) **Operating temperature:** **-40°C ≤ TA ≤ +85°C for Industrial**				
Param. No.	Symbol	Characteristics	Min.	Typ. [1]	Max.	Units	Conditions
VR_1	IVREF	ADC Internal Analog Channel Voltage	1.168	1.200	1.223	V	ADC.CTRLB.ADCHSEL = 1.2V IVREF
VR_9	ACIREF [2]	Internal Comparator Voltage Reference	0.787	0.8	0.84	V	AC.COMPCTRLn.MUXNEG = Bandgap
VR_25	TDRIFT	Internal Voltage Reference Temperature Drift	—	—	0.34	%/°C	Over full operating temperature range
VR_27	VDRIFT	Internal Voltage Reference Voltage Drift	—	—	2.328	%/V	Over full operating voltage range

Notes:

1. Typical values at 25°C.
2. See parameter CMP_23 for Comparator Ref voltage range information.

52.14. Maximum Clock Frequencies

Table 52-18. Maximum Clock Frequencies AC Electrical Specifications

AC CHARACTERISTICS		Standard Operating Conditions: VDDREG=VDDIO=AVDD 1.71V to 3.63V (unless otherwise stated) **Operating temperature:** **-40°C ≤ TA ≤ +85°C for Industrial**		
Param. No.	Symbol	Characteristics	Max	Units
FCLK_1	FCY	MCU clock freq	120	MHz
FCLK_3	fAHB	AHB clock freq	120	MHz
FCLK_5	fAPBn	APBA, APBB, APBC clock freq	120	MHz
FCLK_6	fGCLKGEN[0]	GCLK clock freq output	120	MHz
	fGCLKGEN[1:7]		200	MHz
	fGCLKGEN[8:11]		100	MHz
FCLK_7	fGCLK_PLL	PLL reference clock freq	48	MHz
FCLK_11	fGCLK_DFLL48M_REF	DFLL 48M reference clock freq	1	MHz
FCLK_13	fGCLK_EIC	EIC input clock freq	100	MHz
FCLK_15	fGCLK_FREQM_MSR	FREQM Measure Clock Frequency	200	MHz
FCLK_17	fGCLK_FREQM_REF	FREQM Reference Clock Frequency	100	MHz
FCLK_19	fGCLK_EVSYS_CHANNELx	EVSYS channel x input clock freq	100	MHz
FCLK_21	fGCLK_SERCOMx_SLOW	Common SERCOM slow input clock freq	32	kHz
FCLK_23	fGCLK_SERCOMx_CORE, x = 1 to 7	SERCOMx input clock freq	100	MHz
	fGCLK_SERCOM0_CORE	SERCOM0 input clock freq	160	MHz
FCLK_25	fGCLK_CANx	CAN input clock freq	100	MHz
FCLK_27	fGCLK_USBFS	USB FS input clock freq	48	MHz
FCLK_29	fGCLK_IxS	IxS input clock freq	100	MHz
FCLK_31	fGCLK_SDHCx_SLOW	Common SDHC slow input clock freq	32	kHz
FCLK_33	fGCLK_SDHCx_CORE	SDHCx input clock freq	106	MHz
FCLK_35	fGCLK_TCCx	TCCx input clock freq	200	MHz
FCLK_42	fGCLK_PDEC	PDEC input clock freq	200	MHz
FCLK_43	fGCLK_CCL	CCL input clock freq	100	MHz
FCLK_45	fGCLK_GCLKINx	External GCLKx input clock freq	50	MHz
FCLK_47	fGCLK_CM33_TRACE	CM33 Trace input clock freq	50	MHz
FCLK_49	fGCLK_AC	Analog comparator peripheral module clock freq	100	MHz
FCLK_51	fGCLK_ADCx	ADCx input clock freq	100	MHz
FCLK_55	fGCLK_PTC	PTC input clock freq	50	MHz
FCLK_61	fGCLK_ETH	Ethernet input clock freq	25	MHz
FCLK_62	fGCLK_ETH_TSU (1)	Ethernet TSU input clock freq	< fAHB	MHz
FCLK_73	fGCLK_QSPI	QSPI internal GCLK freq	160	MHz

Note:

1. fGCK_ETH_TSU must be lower than fAHB (FCLK_3) for reliable operation.

52.15. External Oscillator (XOSC) Electrical Specifications

Table 52-19. External XTAL and Clock AC Electrical Specifications (1)

AC CHARACTERISTICS			Standard Operating Conditions: VDDREG=VDDIO=AVDD 1.71V to 3.63V (unless otherwise stated) **Operating temperature:** **-40°C ≤ TA ≤ +85°C for Industrial**				
Param. No.	Symbol	Characteristics	Min.	Typ.	Max.	Units	Conditions [1]
XOSC_1	FOSC_XOSC	XOSC Crystal Frequency	4	—	48	MHz	XOSCCTRLn.XTALEN=1 XIN, XOUT Primary Osc
XOSC_1A	TOSC	TOSC = 1/ FOSC_XOSC	20.83	—	250	ns	See parameter XOSC1 for FOSC_XOSC value
XOSC_2	XOSC_ST (2)	XOSC Crystal Start-up Time	—	—	1300000 (4)	TOSC	Crystal stabilization time only not Oscillator Ready XOSCCTRLA.AGC = 1 FOSC = 48MHz (2)
XOSC_3	CXIN	XOSC XIN parasitic pin capacitance	—	5.1	—	pF	—
XOSC_5	CXOUT	XOSC XOUT parasitic pin capacitance	—	5.7	—	pF	—
XOSC_11		Crystal load capacitance FOSC = 4MHz	—	—		pF	XOSCCTRLA.AGC = 1 Crystal ESR ≤ 500Ω
XOSC_13		Crystal load capacitance FOSC = 8MHz	—	—	20	pF	XOSCCTRLA.AGC = 1 Crystal ESR ≤ 500Ω
XOSC_15		Crystal load capacitance FOSC = 12MHz	—	—		pF	XOSCCTRLA.AGC = 1 Crystal ESR ≤ 250Ω
XOSC_17	CLOAD (3)	Crystal load capacitance FOSC = 16MHz	—	—	18	pF	XOSCCTRLA.AGC = 1 Crystal ESR ≤ 170Ω
XOSC_19		Crystal load capacitance FOSC = 24MHz	—	—		pF	XOSCCTRLA.AGC = 1 Crystal ESR ≤ 80Ω
XOSC_21		Crystal load capacitance FOSC = 32MHz	—	—	12	pF	XOSCCTRLA.AGC = 1 Crystal ESR ≤ 90Ω
XOSC_23		Crystal load capacitance FOSC = 48MHz	—	—	8	pF	XOSCCTRLA.AGC = 1 Crystal ESR ≤ 90Ω
XOSC_33	DLEVEL	MCU Crystal Osc Power Drive Level	—	—	100	µW	XOSCCTRLA.AGC = 1
XOSC_35	FOSC_XCLK	Ext Clock Oscillator Input Freq (XIN pin)	4	—	48	MHz	XOSCCTRLA.XTALEN = 0
XOSC_37	XCLK_DC	Ext Clock Oscillator (XIN) Duty Cycle	40	50	60	%	XOSCCTRLA.XTALEN = 0
XOSC_39	XCLK_FST	Primary XIN Clock Fail Safe Time-out Period	—	4*1/ (DFLL_1/2^XOSCCTRLA.CFDPRESC)	—	µs	—

Table 52-19. External XTAL and Clock AC Electrical Specifications [1] (continued)

AC CHARACTERISTICS			Standard Operating Conditions: VDDREG=VDDIO=AVDD 1.71V to 3.63V (unless otherwise stated) **Operating temperature:** -40°C ≤ TA ≤ +85°C for Industrial				
Param. No.	Symbol	Characteristics	Min.	Typ.	Max.	Units	Conditions [1]

Notes:

1. VDDIOx = AVDD = 3.3V.
2. This is for guidance only. A major component of crystal start-up time is based on the 2nd party crystal MFG parasitics that are outside the scope of this specification. If this is a major concern the customer would need to characterize this based on their design choices.
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)
   - X_{TAL} PCB capacitance typical therefore ~= 2.5 pF for a tight PCB X_{TAL} layout
   - For C_{XIN} and C_{XOUT} within 4 pF of each other, Assume C_{XTAL_EFF} = ((C_{XIN}+C_{XOUT}) / 2)
     Note: Averaging C_{XIN} and C_{XOUT} will effect final calculated C_{LOAD} value by less than 0.25 pF.

   Equation 1: MFG CLOAD Spec = {([C_{XIN} + C1] * [C_{XOUT} + C2]) / [C_{XIN} + C1 + C2 + C_{XOUT}] } + estimated oscillator PCB stray capacitance

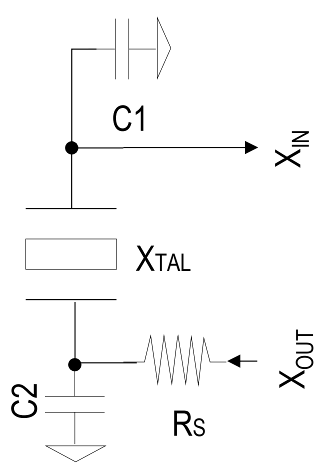
   - Assuming C1 = C2 and C_{XIN} ~= C_{XOUT}, the formula can be further simplified and restated to solve for C1 and C2 by:

   Equation 2: (i.e. 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 = 12 pF
   - PCB X_{TAL} trace Capacitance = 2.5 pF
   - C_{XIN} pin = 6.5 pF, C_{XOUT} pin = 4.5 pF therefore C_{XTAL_EFF}= ((C_{XIN}+$C_{XOUT)}$ / 2) C_{XTAL_EFF} = ((6.5 + 4.5)/2) = 5.5 pF

   C1 = C2 = ((2 * MFG C_{LOAD} spec) - C_{XTAL_EFF} - (2 * PCB capacitance))
   C1 = C2 = (24 - 5.5 - (2 * 2.5))
   C1 = C2 = 13.5 pF (Always rounded down)
   C1 = C2 = 13 pF (i.e. for hypothetical example crystal external load capacitors)
   User C1 = C2 = 13 pF C_{LOAD}(max) spec
4. Start up time selected in XOSCCTRL.STARTUP should be ≥ to this spec.

Figure 52-13. X_{TAL}



52.16. External 32.768 kHz Oscillator (XOSC32) Electrical Specifications

Table 52-20. XOSC32K AC Electrical Specifications

AC CHARACTERISTICS			Standard Operating Conditions: VDDREG=VDDIO=AVDD 1.71V to 3.63V (unless otherwise stated) **Operating temperature:** **-40°C ≤ TA ≤ +85°C for Industrial**				
Param. No.	**Symbol**	**Characteristics**	**Min.**	**Typ.**	**Max.**	**Units**	**Conditions** (1)
XOSC32_1	FOSC_XOSC32	XOSC32 Oscillator Crystal Frequency	—	32.768	—	kHz	XIN32, XOUT32 Secondary Osc
XOSC32_3	CXIN32	XOSC32 XIN32 parasitic pin capacitance	—	8	—	pF	—
XOSC32_5	CXOUT32	XOSC32 XOUT32 parasitic pin capacitance	—	8.5	—	pF	—
XOSC32_11	CLOAD_X32 (3)	32.768kz Crystal Load Capacitance	—	—	12.5	pF	XOSC32K.CGM = 15 XOSC32K.XTALEN = 1 XOSC32K.ENABLE = 1
XOSC32_13	ESR_X32	32.768kz Crystal ESR	—	—	70	KΩ	XOSC32K.CGM = 15 XOSC32K.XTALEN = 1 XOSC32K.ENABLE = 1 Cload = 12.5 pF
XOSC32_15	TOSC32	TOSC32 = 1/ FOSC_XOSC32	—	30.5176	—	µs	See parameter XOSC32_1 for FOSC_XOSC32 value
XOSC32_17	XOSC32_ST (2)	XOSC32 Crystal Start-up Time	—	—	14000 (4)	TOSC	XOSC32K.CGM = 15 XOSC32K.XTALEN = 1 XOSC32K.ENABLE = 1 Crystal ESR = 70 KΩ Cload = 12.5 pF (2) Crystal stabilization time only not Oscillator Ready
XOSC32_19	FOSC_XCLK32	Ext Clock Oscillator Input Freq (XIN32 pin)	31.13	32.768	34.406	kHz	XOSC32K.XTALEN = 0 XOSC32K.ENABLE = 1
XOSC32_21	XCLK32_DC	Ext Clock Oscillator Duty Cycle	25	50	75	%	XOSC32K.XTALEN = 0 XOSC32K.ENABLE = 1
XOSC32_23	XCLK32_FST	XIN32 Clock Fail Safe Time-out Period	—	4*1/ (LP32K_1/2^CFDCTRL.CFDPRESC)	—	µs	—

Table 52-20. XOSC32K AC Electrical Specifications (continued)

AC CHARACTERISTICS			Standard Operating Conditions: VDDREG=VDDIO=AVDD 1.71V to 3.63V (unless otherwise stated) **Operating temperature:** **-40°C ≤ TA ≤ +85°C for Industrial**				
Param. No.	Symbol	Characteristics	Min.	Typ.	Max.	Units	Conditions [1]

Notes:

1. This is for guidance only. A major component of crystal start-up time is based on the 2nd party crystal MFG parasitics that are outside the scope of this specification. If this is a major concern the customer would need to characterize this based on their design choices.
2. 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)
   - X_{TAL} PCB capacitance typical therefore ~= 2.5pF for a tight PCB X_{TAL} layout
   - For C_{XIN} and C_{XOUT} within 4pF of each other, Assume C_{XTAL_EFF} = ((C_{XIN}+C_{XOUT}) / 2)
     Note: Averaging C_{XIN} and C_{XOUT} will effect final calculated CLOAD value by less than the tolerance of the capacitor selection.

   EQUATION 1:

   MFG CLOAD 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: (i.e. Simplified Equation #1)

   C1 = C2 = ((2 * MFG CLOAD spec) - C_{XTAL_EFF} - (2 * PCB capacitance))

   EXAMPLE ONLY:

   - X_{TAL} Mfg CLOAD Data Sheet Spec = 12 pF
   - PCB X_{TAL} trace Capacitance = 2.5 pF
   - C_{XIN} pin = 6.5 pF, C_{XOUT} pin = 4.5 pF therefore C_{XTAL_EFF} = ((C_{XIN}+C_{XOUT}) / 2) C_{XTAL_EFF} = ((6.5 + 4.5)/2) = 5.5 pF

   C1 = C2 = ((2 * MFG CLOAD spec) - C_{XTAL_EFF} - (2 * PCB capacitance))

   C1 = C2 = (24 - 5.5 - (2 * 2.5))

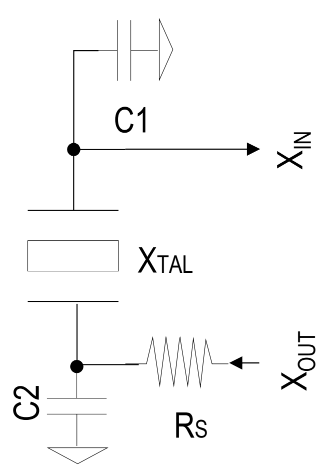
   C1 = C2 = (24 - 5.5 - 5)

   C1 = C2 = 13.5 pF (Always rounded down)

   C1 = C2 = 13 pF (i.e., for hypothetical example crystal external load capacitors)

   User C1=C2=13pF ≤ C_{LOAD_X32}(max) spec
3. User Selectable in XOSC32K.STARTUP.

Figure 52-14. X_{TAL}



52.17. Low Power Internal 32kHz RC Oscillator (OSCULP32K) Electrical Specifications

Table 52-21. OSCULP32K AC Electrical Specifications

AC CHARACTERISTICS			Standard Operating Conditions: VDDREG=VDDIO=AVDD 1.71V to 3.6V (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	30.1	32.768	36.8	kHz	-40°C ≤ T_A ≤ +85°C (i.e. Factory Default Calibration)
LP32K_9	RC32K_Duty	LPRC32K OSC Duty Cycle	—	50	—	%	—

52.18. DFLL/PLL Electrical Specifications

Table 52-22. DFLL (Digital Frequency Locked Loop) AC Electrical Specifications

AC CHARACTERISTICS			Standard Operating Conditions: VDDREG=VDDIO=AVDD 1.71V to 3.63V (unless otherwise stated) **Operating temperature:** **-40°C ≤ TA ≤ +85°C for Industrial**				
Param. No.	Symbol	Characteristics	Min.	Typ.	Max.	Units	Conditions
DFLL48MHz (Open Loop) [1,2]							
DFLL_1	DFLL_OL_FOUT	DFLL Open Loop Clock Frequency	45.60	48.00	49.12	MHz	Normal Mode (DFLLCTRLA.LOWFREQ=0)
			6.76	8.00	9.20		Low Frequency Mode (DFLLCTRLA.LOWFREQ=1)
DFLL_9	DFLL_OL_SRT	Start-Up (Ready bit valid)	—	13	16	µs	Normal Mode (DFLLCTRLA.LOWFREQ=0)
			—	68	75		Low Frequency Mode (DFLLCTRLA.LOWFREQ=1)
DFLL48MHz (Closed Loop) [3,4]							
DFLL_11	DFLL_CL_FIN [4]	DFLL Closed loop Input Frequency Range	1000	32768	1000000	Hz	—
DFLL_13	DFLL_CL_FOUT [6]	DFLL Closed Loop Clock Frequency	47.88	48.00	48.12	MHz	Normal Mode (DFLLCTRLA.LOWFREQ=0) XOSC32 32.768 kHz PPM ≤ 100, DFLLMUL = 1465
			7.98	8.00	8.02		Low Frequency Mode (DFLLCTRLA.LOWFREQ=1) XOSC32 32.768 kHz PPM ≤ 100, DFLLMUL = 244
DFLL_15	DFLL_CL_Jitter	DFLL Period Jitter Pk-to-Pk	—	0.92	2.700	%	Normal Mode (DFLLCTRLA.LOWFREQ=0) XOSC32 32.768 kHz PPM ≤ 100, DFLLMUL = 1465
DFLL_17			—	1.0	2.000	%	Low Frequency Mode (DFLLCTRLA.LOWFREQ=1) XOSC32 32.768 kHz PPM ≤ 100, DFLLMUL = 244
DFLL_21	DFLL_CL_SRT [5, 6]	DFLL Closed Loop Mode / Lock Time	—	0.39	0.900	ms	Normal Mode (DFLLCTRLA.LOWFREQ=0) XOSC32 32.768 kHz PPM ≤ 100, DFLLMUL = 1465
			—	0.33	1.000		Low Frequency Mode (DFLLCTRLA.LOWFREQ=1) XOSC32 32.768 kHz PPM ≤ 100, DFLLMUL = 244

Table 52-22. DFLL (Digital Frequency Locked Loop) AC Electrical Specifications (continued)

AC CHARACTERISTICS			Standard Operating Conditions: VDDREG=VDDIO=AVDD 1.71V to 3.63V (unless otherwise stated) **Operating temperature:** **-40°C ≤ TA ≤ +85°C for Industrial**				
Param. No.	Symbol	Characteristics	Min.	Typ.	Max.	Units	Conditions

Notes:
1. In Open Loop mode the DFLL uses a self contained internal RC oscillator clock source who's course calibrated value is loaded at out of reset. In addition, there is a fine tune trim register (DFLLTUNE) the user software can assess.
2. Not recommended for functional USB operation, SOF sync start-up only.
3. In Closed loop mode the DFLL can use a variety of clock sources. The DFLL can be trimmed using register DFLLMUL.
4. To insure that the DFLL stays within the +/-0.25% of its clock frequency, any reference clock for DFLL in close loop must be within a 8% maximum error accuracy.
5. REFCLK for DFLL or XOSC32K,PLL is XOSC.
6. DFLLCTRLB.QLDIS = 0: quick lock enabled (default), DFLLCTRLB.CCDIS = 0: chill cycles enabled (default). DFLLMUL.STEP = 8 : Max fine step size, divided or dividing into two parts, search 8 is optimum value.

   During a maximum of 30 cycles of the reference clock period (250 cycles for 1Mhz), between lock flag asserted and frequency stabilization, DFLL frequency accuracy will be limited to +/-1.5% at 48Mhz (+/-3% at 8Mhz); after this duration, the frequency accuracy is within +/-0.25%.

   At 1khz reference clock and to maintain the frequency accuracy within +/-0.25% for 48Mhz, a reduced STEP value at 4 (instead of the optimum 8) eliminates this period of inaccuracy, at the expense of maximum 15% of lock time.

Table 52-23. PLL (Frequency Digital Phase Locked Loop) AC Electrical Specifications

AC CHARACTERISTICS			Standard Operating Conditions: VDDREG=VDDIO=AVDD 1.71V to 3.63V (unless otherwise stated) **Operating temperature:** **-40°C ≤ TA ≤ +85°C for Industrial**				
Param. No.	Symbol	Characteristics	Min.	Typ.	Max.	Units	Conditions
PLLxxMHz (Fractional Digital Phase Locked Loop)							
PLL_1	PLL_FIN	PLL Input Frequency Range	4	—	48	MHz	Over full voltage and temperature operating ranges
PLL_3	PLL_FOUT	PLL Output Clock Frequency	12.7	—	200	MHz	
PLL_11	PLL_SRT(5)	PLL Lock Time	—	25	—	µs	—

Notes:
1. In Open Loop mode the DFLL uses a self contained internal RC oscillator clock source who's course calibrated value is loaded at out of reset. In addition, there is a fine tune trim register (DFLLTUNE) the user software can assess.
2. Not recommended for functional USB operation, SOF sync start-up only.
3. In Closed loop mode the DFLL can use a variety of clock sources. The DFLL can be trimmed using register DFLLMUL.
4. To insure that the DFLL stays within the +/-0.25% of its clock frequency, any reference clock for DFLL in close loop must be within a 8% maximum error accuracy.
5. REFCLK for DFLL or XOSC32K,PLL is XOSC.
6. DFLLCTRLB.QLDIS = 0: quick lock enabled (default), DFLLCTRLB.CCDIS = 0: chill cycles enabled (default). DFLLMUL.STEP = 8: Max fine step size, divided or dividing into two parts, search 8 is optimum value.

   During a maximum of 30 cycles of the reference clock period (250 cycles for 1Mhz), between lock flag asserted and frequency stabilization, DFLL frequency accuracy will be limited to +/-1.5% at 48Mhz (+/-3% at 8Mhz); after this duration, the frequency accuracy is within +/-0.25%.

   At 1khz reference clock and to maintain the frequency accuracy within +/-0.25% for 48Mhz, a reduced STEP value at 4 (instead of the optimum 8) eliminates this period of inaccuracy, at the expense of maximum 15% of lock time.

52.19. Analog-to-Digital Converter (ADC) Electrical Specifications

Table 52-24. ADC AC Electrical Specifications

AC CHARACTERISTICS			Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated) **Operating temperature:** **-40°C ≤ TA ≤ +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	VDDIOx = AVDD
Reference Inputs							
ADC_3	VREF[4]	ADC Reference Voltage [4]	The greater of ≥ AVDD(min) or 2.4V [4]	—	AVDD	V	VREF ≤ AVDD
Analog Input Range							
ADC_7	AFS	Full-Scale Analog Input Signal Range (Single-Ended)	AVSS	—	VREF	V	VREF = AVDD(max)
ADC_9		Full-Scale Analog Input Signal Range (Differential)	-VREF	—	VREF	V	

Notes:
1. Characterized with an analog input sine wave = (FTP(max)/100). For example: FTP(max) = 1 Msps /100 = 10 kHz sine wave.
2. Sine wave peak amplitude = 96% ADC_ Full Scale amplitude input with 12bit resolution.
3. ADC is configured in 12-bits mode, All registers are at the reset default value unless otherwise stated.
4. ADC functional device operation with either internal or external VREF < 2.4V is functional, but not characterized. ADC will function, but with degraded accuracy of approximately ~ ((0,006 * 2^n)/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.
5. Value taken over 7 harmonics.
6. Value coming from simulation.

Table 52-25. ADC Single Ended Mode AC Electrical Specifications

AC CHARACTERISTICS			Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated) **Operating temperature:** **-40°C ≤ TA ≤ +85°C for Industrial**				
Param. No.	Symbol	Characteristics	Min.	Typ.	Max.	Units	Conditions
SINGLE ENDED MODE ADC Accuracy							
SADC_11	Res	Resolution	8	—	12	bits	Selectable 8, 10, 12 bit Resolution Ranges
SADC_13	INL [3]	Integral Nonlinearity	-2	±1.25	2	LSB	3.125 Msps, Internal VREF = AVDD = VDDIO = 3.3V
SADC_19	DNL [3]	Differential Nonlinearity	-1	-0.75 / +1	2	LSB	3.125 Msps, Internal VREF = AVDD = VDDIO = 3.3V
SADC_25	GERR [3,6]	Gain Error	-5	—	-1	LSB	3.125 Msps, Internal VREF = AVDD = VDDIO = 3.3V
SADC_31	EOFF [3,6]	0ffset Error	1	—	4	LSB	3.125 Msps Internal VREF = AVDD = VDDIO = 3.3V
SINGLE ENDED MODE ADC Dynamic Performance [1,2]							

Table 52-25. ADC Single Ended Mode AC Electrical Specifications (continued)

AC CHARACTERISTICS			Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated) **Operating temperature:** **-40°C ≤ TA ≤ +85°C for Industrial**				
Param. No.	Symbol	Characteristics	Min.	Typ.	Max.	Units	Conditions
SADC_43	ENOB [3]	Effective Number of bits	10.6	11.2	—	bits	VREF = AVDD = VDDIO = 3.3V @ 12 bit at 3.125 Msps
SADC_45	SINAD [1,2,3]	Signal to Noise and Distortion	65	70	—	dB	
SADC_47	SNR [1,2,3]	Signal to Noise ratio	65	70	—		
SADC_51	THD [1,2,3,5]	Total Harmonic Distortion	—	-80	-75		

Notes:

1. Characterized with an analog input sine wave = (FTP(max) /100). For example: FTP(max) = 1 Msps/100 = 10 kHz sine wave.
2. Sine wave peak amplitude = 96% ADC_ Full Scale amplitude input with 12-bit resolution.
3. ADC is configured in 12 bits mode, All registers are at the reset default value unless otherwise stated.
4. ADC functional device operation with either internal or external VREF < 2.4V is functional, but not characterized. ADC will function, but with degraded accuracy of approximately ~ ((0,006 * 2^n) / 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 and AVSS.
5. Value taken over 7 harmonics.
6. Value coming from simulation.

Table 52-26. ADC Differential Mode AC Electrical Specifications

AC CHARACTERISTICS			Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated) **Operating temperature:** **-40°C ≤ TA ≤ +85°C for Industrial**				
Param. No.	Symbol	Characteristics	Min.	Typ.	Max.	Units	Conditions
DIFFERENTIAL MODE ADC Accuracy							
DADC_11	Res	Resolution	8	—	12	bits	Selectable 8, 10, 12 bit Resolution Ranges
DADC_13	INL [3]	Integral Nonlinearity	-2	±1.25	2	LSB	3.125 Msps, Internal VREF = AVDD = VDDIO = 3.3V
DADC_19	DNL [3]	Differential Nonlinearity	-1	-0.75 / +1	2	LSB	3.125 Msps, Internal VREF = AVDD = VDDIO = 3.3V
DADC_25	GERR [3,6]	Gain Error	-5	—	-1	LSB	3.125 Msps, Internal VREF = AVDD = VDDIO = 3.3V
DADC_31	EOFF [3,6]	Offset Error	1	—	2	LSB	3.125 Msps, Internal VREF = AVDD = VDDIO = 3.3V
DIFFERENTIAL MODE ADC Dynamic Performance [1,2]							
DADC_43	ENOB [3]	Effective Number of bits	11.2	11.4	—	bits	VREF = AVDD = VDDIO = 3.3V @ 12 bit at 3.125 Msps
DADC_45	SINAD [1,2,3]	Signal to Noise and Distortion	68	70	—	dB	
DADC_47	SNR [1,2,3]	Signal to Noise ratio	68	70	—		
DADC_51	THD [1,2,3,5]	Total Harmonic Distortion	—	-84	-80		

Table 52-26. ADC Differential Mode AC Electrical Specifications (continued)

AC CHARACTERISTICS			Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated) **Operating temperature:** **-40°C ≤ TA ≤ +85°C for Industrial**				
Param. No.	Symbol	Characteristics	Min.	Typ.	Max.	Units	Conditions

Notes:
1. Characterized with an analog input sine wave = (FTP(max) /100). Example: FTP(max) = 1 Msps/100 = 10 kHz sine wave.
2. Sine wave peak amplitude = 96% ADC_ Full Scale amplitude input with 12bit resolution.
3. ADC functional device operation with either internal or external VREF < 2.4V is functional, but not characterized. ADC will function, but with degraded accuracy of approximately ~ ((0,006 * 2^n) / 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 and AVSS.
4. Value taken over 7 harmonics.
5. Value coming from simulation.

Table 52-27. ADC Conversion and Sample AC Electrical Specifications

AC CHARACTERISTICS			Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated) **Operating temperature:** **-40°C ≤ TA ≤ +85°C for Industrial**				
Param. No.	Symbol	Characteristics	Min.	Typ.	Max.	Units	Conditions
ADC Clock Requirements							
ADC_53	TAD	ADC Clock Period	20	—	7145	ns	VREF = AVDD = 3.3V and Res = 6, 8, 10 bit
			20	—	1250	ns	VREF = AVDD = 3.3V and Res = 12 bit
ADC_55	fGCLK_ADCx	ADCx Module GCLK max input freq	—	—	FCLK_51	MHz	VREF = AVDD = 3.3V
ADC Throughput Rates							
ADC_57	FTPR [1]	Sample-Rate for ADC with SAMC=1 (min)	—	—	3.125	Msps	12-bit resolution, Rsource Impedance ≤ 200 Ω
			—	—	3.571429		10-bit resolution, Rsource Impedance ≤ 250 Ω
			—	—	4.166667		8-bit resolution, Rsource Impedance ≤ 300 Ω
			—	—	5		6-bit resolution, Rsource Impedance ≤ 400 Ω
ADC Conversion and Sample Time							

Table 52-27. ADC Conversion and Sample AC Electrical Specifications (continued)

AC CHARACTERISTICS			Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated) **Operating temperature:** **-40°C ≤ TA ≤ +85°C for Industrial**				
Param. No.	Symbol	Characteristics	Min.	Typ.	Max.	Units	Conditions
ADC_59	TSAMP	Sample-Time for ADC	3	—	—	TAD	12 bit TAD(min), Ext Analog Input Rsource ≤ 200 Ω, Max ADC Clock
			3				10 bit TAD(min), Ext Analog Input Rsource ≤ 250 Ω, Max ADC Clock
			8	—	—		12 bit TAD(min), Ext Analog Input Rsource ≤ 500 Ω, Max ADC Clock
			8				10 bit TAD(min), Ext Analog Input Rsource ≤ 700 Ω, Max ADC Clock
			14	—	—		12 bit TAD(min), Ext Analog Input Rsource ≤ 1 kΩ, Max ADC Clock
			14				10 bit TAD(min), Ext Analog Input Rsource ≤ 1.25 kΩ, Max ADC Clock
			64	—	—		12 bit TAD(min), Ext Analog Input Rsource ≤ 5 kΩ, Max ADC Clock
			64				10 bit TAD(min), Ext Analog Input Rsource ≤ 5.5 kΩ, Max ADC Clock
ADC_61	TCNV	Conversion Time (after sample time is complete)	13			TAD	12-bit resolution
			11				10-bit resolution
			9				8-bit resolution
			7				6-bit resolution
ADC_63	Twarm-up	Warm Up Time after CTRLA.ANAEN = 1 and CTRLA.ENABLE = 1	—	—	500 TAD or 20 µs, which ever is bigger	µs	

Note:

1. ADC Throughput Rate FTP = ((1 / ((TSAMP + TCNV) * TAD)) / (Number of user active analog inputs in use on specific target ADC module)).
   Specification values assume only one AINx channel in use.

52.20. Comparator Electrical Specifications

Table 52-28. Comparator AC Electrical Specifications

AC CHARACTERISTICS			Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated) **Operating temperature:** **-40°C ≤ TA ≤ +85°C for Industrial**				
Param. No.	Symbol	Characteristics	Min.	Typ.(1)	Max.	Units	Conditions
CMP_1	VIOFF	Input Offset Voltage	-8	—	8	mV	COMPCTRLn.HYST = 0x3. Comparator ref voltage = AVDD/2
CMP_4	VIN	Input Voltage Range	AVSS	—	AVDD	V	With respect to GND and AVDD

Table 52-28. Comparator AC Electrical Specifications (continued)

AC CHARACTERISTICS			Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated) **Operating temperature:** **-40°C ≤ TA ≤ +85°C for Industrial**				
Param. No.	Symbol	Characteristics	Min.	Typ.(1)	Max.	Units	Conditions
CMP_5	VHYST_00	Input Hysteresis Voltage	2	—	14	mV	COMPCTRLn.HYST = 0x0. Comparator ref voltage = AVDD/2
	VHYST_01	Input Hysteresis Voltage	6	—	26	mV	COMPCTRLn.HYST = 0x1. Comparator ref voltage = AVDD/2
	VHYST_02	Input Hysteresis Voltage	20	—	56	mV	COMPCTRLn.HYST = 0x2. Comparator ref voltage = AVDD/2
	VHYST_03	Input Hysteresis Voltage	34	—	85	mV	COMPCTRLn.HYST = 0x3. Comparator ref voltage = AVDD/2
CMP_15	TRESPSS ain<0:1>	Small Signal Response Time - High speed	—		64	ns	COMPCTRLn.SPEED = 0x0 COMPCTRLn.MUXPOS = 0x0 COMPCTRLn.MUXNEG = 0x1 COMPCTRLn.HYST = 0x3 Comparator ref voltage = AVDD/2. Input overdrive = +/- 180 mV
		Small Signal Response Time - Low Speed	—		260	ns	COMPCTRLn.SPEED = 0x1 COMPCTRLn.MUXPOS = 0x0 COMPCTRLn.MUXNEG = 0x1 COMPCTRLn.HYST = 0x3 Comparator ref voltage = AVDD/2. Input overdrive = +/- 180mV
	TRESPSS ain<2:3>	Small Signal Response Time - High speed	—		100	ns	COMPCTRLn.SPEED = 0x0 COMPCTRLn.MUXPOS = 0x2 COMPCTRLn.MUXNEG = 0x3 COMPCTRLn.HYST = 0x3 Comparator ref voltage = AVDD/2. Input overdrive = +/- 180mV
		Small Signal Response Time - Low Speed	—		260	ns	COMPCTRLn.SPEED = 0x1 COMPCTRLn.MUXPOS = 0x2 COMPCTRLn.MUXNEG = 0x3 COMPCTRLn.HYST = 0x3 Comparator ref voltage = AVDD/2. Input overdrive = +/- 180 mV
CMP_19	COUTVAL	Comparator Enabled to Output Valid - High Speed	—	—	2	ms	Comparator module is configured before enabling it COMPCTRLn.SPEED = 0x0
		Comparator Enabled to Output Valid - Low Speed	—	—	20	ms	Comparator module is configured before enabling it COMPCTRLn.SPEED = 0x1
CMP_23	CVREFRNG	Comparator Voltage Reference Input Range	—	—	—	V	(1)
CMP_25	FGCLK_AC	Analog comparator peripheral module clock freq	—	—	FCLK_49	MHz	See FCLK_49 in Maximum Clock Frequencies table

Table 52-28. Comparator AC Electrical Specifications (continued)

AC CHARACTERISTICS			Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated) **Operating temperature:** **-40°C ≤ TA ≤ +85°C for Industrial**				
Param. No.	Symbol	Characteristics	Min.	Typ.[1]	Max.	Units	Conditions

Note:

1. Comparator Ref voltage cannot exceed: VIN(max) - VIOFF(max) - CMP_5(max) - 100mV) ≥CMP_23 ≥ (VIN(min) + abs(VIOFF(min)) + CMP_5(max) + 100mV).

Table 52-29. DAC Module AC Electrical Specifications

AC CHARACTERISTICS			Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated) **Operating temperature:** **-40°C ≤ TA ≤ +85°C for Industrial**				
Param. No.	Symbol	Characteristics	Min.	Typ.[1]	Max.	Units	Conditions
DAC_1	DRES	DAC Resolution	—	—	7	Bits	—
DAC_7	VOUT	Output Voltage Range	AVSS	—	AVDD - 1LSB	V	—
DAC_17	Tset_FS	DAC Full Scale Settling Time	—	—	1.5	µs	+/-4 LSB of final value for step size from 10% to 90%
DAC_18A	Thold	DAC Holding Time	0.29	—	—	ms	This parameter applies only for Low-Power mode: DACCTRLn.SHENm = 0x1 Time for 1 LSB voltage loss of the DAC output
DAC_18B			0.65	—	—	ms	This parameter applies only for Low-Power mode: DACCTRLn.SHENm = 0x1 Time for 10 LSB voltage loss of the DAC output
SDAC_19	INL	Integral Non- Linearity	-0.4	—	0.4	LSB	—
SDAC_21	DNL	Differential Non-Linearity	0.05	—	0.4	LSB	—
SDAC_23	GERR	Gain Error	-1.2	—	-0.2	LSB	—
SDAC_25	EOFF	Offset Error	-0.7	—	0.7	LSB	Normal mode: DACCTRLn.SHENm = 0x0
			-5.7	—	5.7	LSB	Low-Power mode: DACCTRLn.SHENm = 0x1

52.21. Peripheral Touch Controller (PTC) Electrical Specifications

Table 52-30. Peripheral Touch Controller (PTC) AC Electrical Specifications

AC CHARACTERISTICS			Standard Operating Conditions: VDDREG=VDDIO=AVDD 1.71V to 3.6V (unless otherwise stated) **Operating temperature:** **-40°C ≤ T_A ≤ +85°C for Industrial**				
Param. No.	Symbol	Characteristics	Min.	Typ.	Max.	Units	Conditions
PTC_1	CLOAD SC [1]	Self Capacitance Mode (PTC Channel Y0 - Y31) Yn	—	—	70	pF	Maximum sensor load capacitance[1]
PTC_3	CLOAD MC[1]	Mutual Capacitance Mode (PTC Channel Y0 - Y31)	—	—	35	pF	

Table 52-30. Peripheral Touch Controller (PTC) AC Electrical Specifications (continued)

AC CHARACTERISTICS			Standard Operating Conditions: VDDREG=VDDIO=AVDD 1.71V to 3.6V (unless otherwise stated) **Operating temperature:** **-40°C ≤ T_A ≤ +85°C for Industrial**				
Param. No.	Symbol	Characteristics	Min.	Typ.	Max.	Units	Conditions
PTC_4A	fPTC [2]	PTC Frequency	—	—	4	MHz	—
PTC_4B	Twarm	Warm-up time	—	—	500 PTC clock cycles	µs	—

Notes:

1. Maximum capacitive load that the PTC circuitry can compensate for each channel.
2. fPTC = GCLK_PTC / PTC prescaler.

52.22. Serial Peripheral Interface (SPI) Electrical Specifications

Figure 52-15. SPIx Host Module CPHA = 0 Timing Diagrams

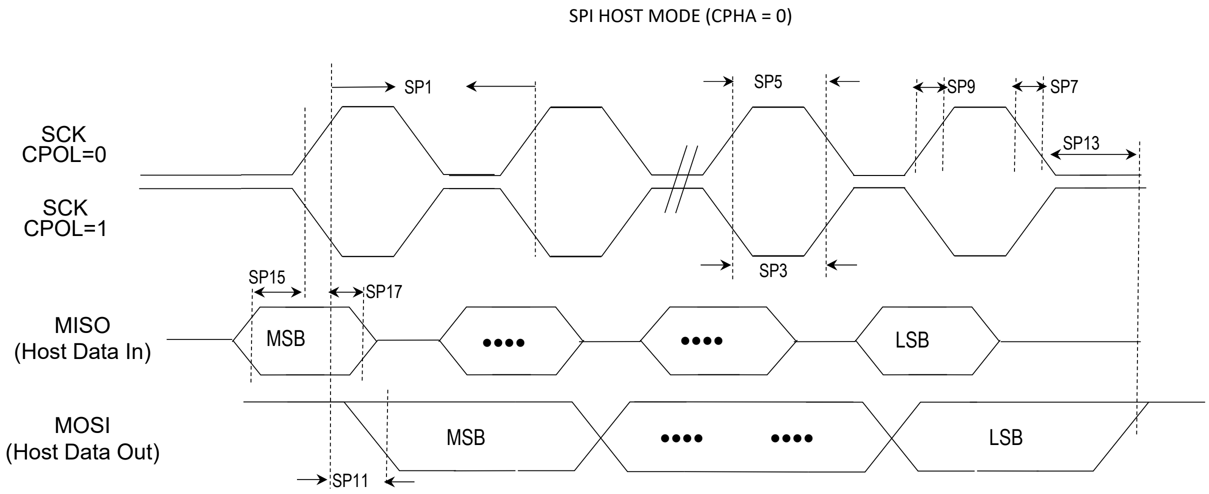


Figure 52-16. SPIx Host Module CPHA = 1 Timing Diagrams

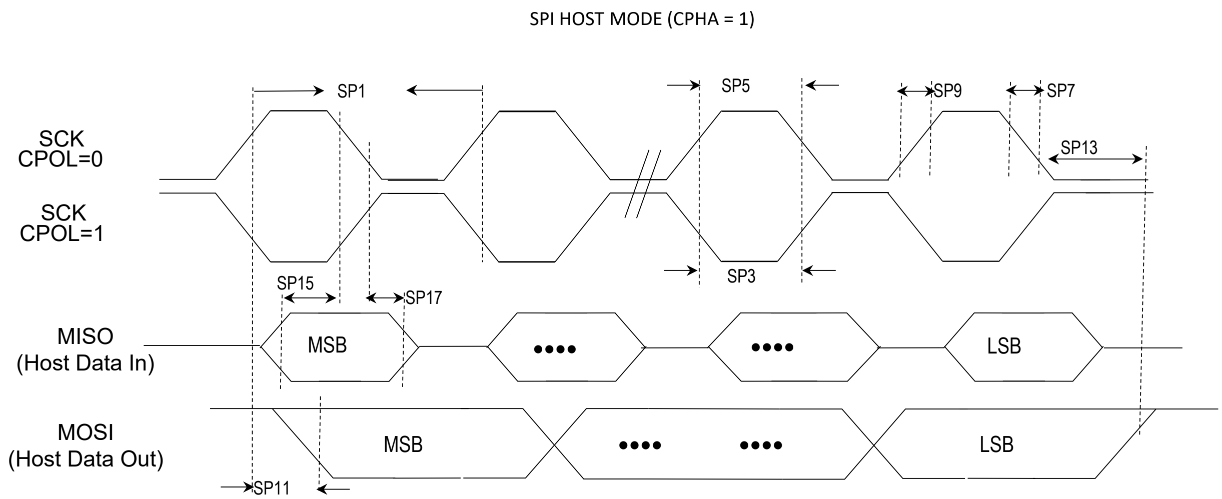


Table 52-31. SPIx Module Host Mode Electrical Specifications [1]

AC CHARACTERISTICS				Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated) **Operating temperature:** **-40°C ≤ TA ≤ +85°C for Industrial**				
Param. No.	Symbol	Characteristics [1]		Min.	Typ.	Max.	Units	Conditions
MSP_1	FSCK	SCK Frequency	Normal Speed SERCOM	—	—	21.3	MHz	Transmitter mode, CTRB.RXEN = 0. VDDIOx = 1.8V or 3.3 V, CLOAD = 30 pF(max)
				—	—	21.3		Full Duplex Transmit and Receive modes, Loop-Back mode with one SPI communicating to another SPI on the same MCU. VDDIOx = 1.8V or 3.3V, CLOAD = 30 pF(max)
				—	—	1/(2*(TMIS+NOTE2_TV)) [2]		Full Duplex Transmit and Receive modes, CLOAD = 30pF(max). The maximum SPI speed of the MCU is partially dependent on the external SPI device performance characteristics. Faster speeds than the Loop-Back mode above may therefore be possible using the formula.
			High-Speed SERCOM [3]	—	—	48.1	MHz	Transmitter mode, CTRB.RXEN = 0. VDDIOx = 1.8V or 3.3 V, CLOAD = 20 pF(max)
				—	—	1/(2*(TMIS_HS+NOTE2_TV)) [2]		Full-Duplex Transmit and Receive modes, CLOAD = 20pF(max). The maximum SPI speed of the MCU is partially dependent on the external SPI device performance characteristics. Faster speeds than the Loop-Back mode above may therefore be possible using the formula.
MSP_3	TSCL	SCK Output Low Time		1/(2*FSCK)	—	—	μs	—
MSP_5	TSCH	SCK Output High Time		1/(2*FSCK)	—	—	μs	—
MSP_7	TSCF	SCK and MOSI Output Fall Time		—	—	DI_27 or DI_31	ns	See parameter in I/O Specifications
MSP_9	TSCR	SCK and MOSI Output Rise Time		—	—	DI_25 or DI_29	ns	See parameter in I/O Specifications

Table 52-31. SPIx Module Host Mode Electrical Specifications [1] (continued)

AC CHARACTERISTICS				Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated) **Operating temperature:** **-40°C ≤ TA ≤ +85°C for Industrial**				
Param. No.	Symbol	Characteristics [1]		Min.	Typ.	Max.	Units	Conditions
MSP_11	TMOV	MOSI Data Output Valid after SCK	Normal Speed SERCOM	—	—	10	ns	—
			High Speed SERCOM [3]	—	—	4	ns	—
MSP_13	TMOH	MOSI hold after SCK		0	—	—	ns	—
MSP_15A	TMIS	MISO Setup Time of Data Input to SCK	Normal Speed SERCOM	15	—	—	ns	—
			High Speed SERCOM [3]	1	—	—	ns	—
MSP_17	TMIH	MISO Hold Time of Data Input to SCK	Normal Speed SERCOM	0	—	—	ns	—
			High Speed SERCOM [3]	2	—	—	ns	—
MSP_19	SPI_GCLK	SERCOM SPI input clock frequencies, GCLK_SPI		—	—	FCLK_23	MHz	—

Notes:
1. Assuming 30 pF external load on all SPIx pins unless otherwise noted.
2. NOTE2_TV is the client external device data output valid time from clock edge specification.
3. SERCOM0 pins: PC10, PC11, PC12, PC13.

Figure 52-17. SPIx Client Mode CPHA = 0 Timing Diagrams

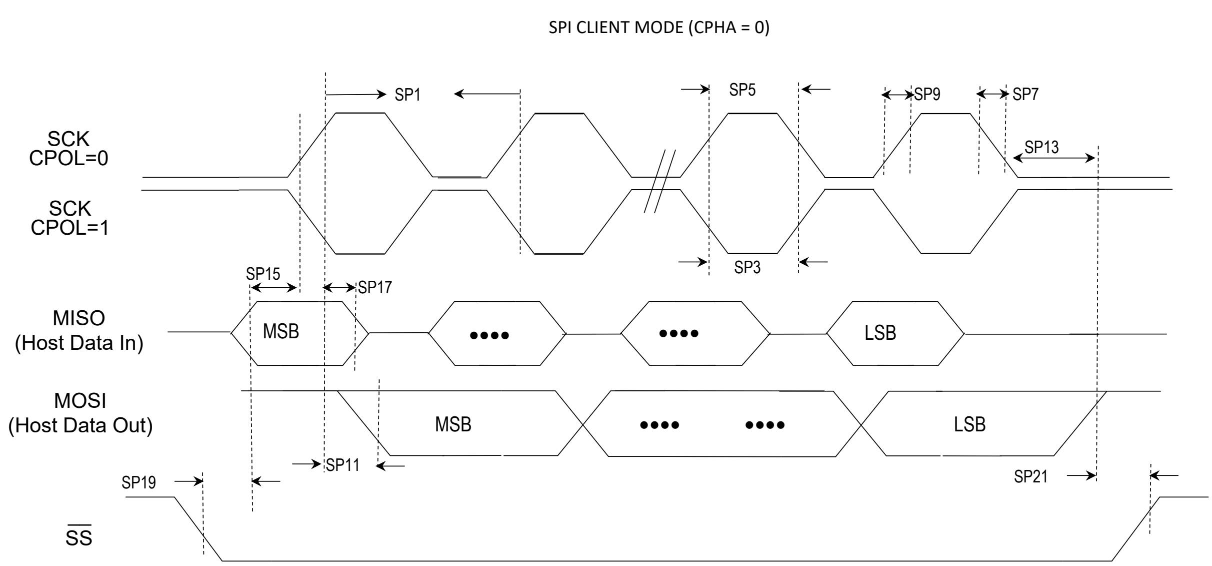


Figure 52-18. SPIx Client Mode CPHA = 1 Timing Diagrams

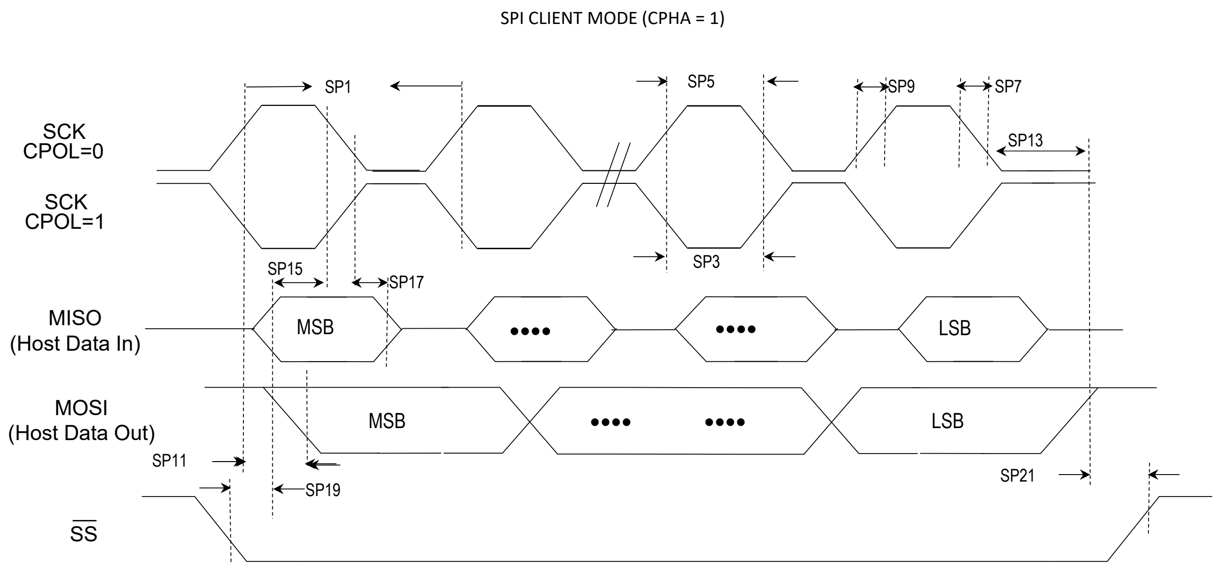


Table 52-32. SPIx Module Client Mode Electrical Specifications

AC CHARACTERISTICS				Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated) **Operating temperature:** **-40°C ≤ TA ≤ +85°C for Industrial**				
Param. No.	Symbol	Characteristics[1]		Min.	Typical	Max.	Units	Conditions
SSP_1	FSCK	SCK Frequency	Normal Speed SERCOM	—	—	21.3	MHz	Receiver mode. VDDIOx = 1.8V or 3.3V, CLOAD = 30 pF(max)
				—	—	21.3		Full-Duplex Transmit and Receive modes, Loop-Back mode with one SPI communicating to another SPI on the same MCU. VDDIOx = 1.8V or 3.3V, CLOAD = 30 pF(max)
				—	—	1/(2*(TSOV+NOTE2_TMIS)) [2]		Full- Duplex Transmit and Receive modes, CLOAD = 30pf(max). The max SPI speed of the MCU is partially dependent on the external SPI device performance characteristics. Faster speeds than the Loop-Back mode above may therefore be possible using the formula.
			High-Speed SERCOM [3]	—	—	48.1	MHz	Receiver mode. VDDIOx = 1.8V or 3.3V, CLOAD = 20 pF(max)
				—	—	1/(2*(TSOV_HS+NOTE2_TMIS)) [2]		Full-Duplex Transmit and Receive modes, CLOAD = 20pF(max). The maximum SPI speed of the MCU is partially dependent on the external SPI device performance characteristics. Faster speeds than the Loop-Back mode above may therefore be possible using the formula.
SSP_3	TSCL	SCK Output Low Time		1/(2*FSCK)	—	—	μs	—
SSP_5	TSCH	SCK Output High Time		1/(2*FSCK)	—	—	μs	—

Table 52-32. SPIx Module Client Mode Electrical Specifications (continued)

AC CHARACTERISTICS				Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated) **Operating temperature:** **-40°C ≤ TA ≤ +85°C for Industrial**				
Param. No.	Symbol	Characteristics[1]		Min.	Typical	Max.	Units	Conditions
SSP_7	TSCF	SCK and MOSI Output Fall Time		—	—	DI_27 or DI_31	ns	See parameter in *I/O spec*
SSP_9	TSCR	SCK and MOSI Output Rise Time		—	—	DI_25 or DI_29	ns	See parameter in *I/O spec*
SSP_11A	TSOV	MISO Data Output Valid after SCK	Normal Speed SERCOM	—	—	25	ns	—
			High Speed SERCOM [3]	—	—	10	ns	—
SSP_13	TSOH	MISO hold after SCK		4	—	—	ns	—
SSP_15	TSIS	MOSI Setup Time of Data Input to SCK	Normal Speed SERCOM	6	—	—	ns	—
			High Speed SERCOM [3]	2	—	—	ns	—
SSP_17	TSIH	MOSI Hold Time of Data Input to SCK	Normal Speed SERCOM	0	—	—	ns	—
			High Speed SERCOM [3]	1	—	—	ns	—
SSP_19	TSSS	$\overline{SS}$ setup to SCK (PRELOADEN=0)	Normal Speed SERCOM	5	—	—	ns	—
			High Speed SERCOM [3]	2	—	—	ns	—
SSP_21	TSSH	$\overline{SS}$ hold after SCK Client	Normal Speed SERCOM	0	—	—	ns	—
			High Speed SERCOM [3]	1	—	—	ns	—
SSP_23	SPI_GCLK	SERCOM SPI input clk freq, GCLK_SPI		—	—	FCLK_23	MHz	—

Notes:
1. Assuming and 30 pF external load on all SPIx pins unless otherwise noted.
2. NOTE2_TMIS is the Host external device setup time.
3. SERCOM0 pins: PC10, PC11, PC12, PC13.

52.23. UART Electrical Specifications

Table 52-33. UART AC Electrical Specifications

AC CHARACTERISTICS				Standard Operating Conditions: VDDREG=VDDIO=AVDD 1.71V to 3.63V (unless otherwise stated) **Operating temperature:** **-40°C ≤ TA ≤ +85°C for Industrial**				
Param. No.	Symbol	Characteristics		Min.	Typ.	Max.	Units	Conditions
UT_1A	FBRATE	Baud Rate - Normal Speed SERCOM	Asynchronous SAMPR = 16x mode	—	—	6.25	Mbps	VDDIO = 3.3V, CLOAD = 30 pF(MAX)
				—	—	6.25	Mbps	VDDIO = 1.8V, CLOAD = 30 pF(MAX)
UT_3A			Asynchronous SAMPR = 8x mode	—	—	12.5	Mbps	VDDIO = 3.3V, CLOAD = 30 pF(MAX)
				—	—	12.5	Mbps	VDDIO = 1.8V, CLOAD = 30 pF(MAX)
UT_5A			Asynchronous SAMPR = 3x mode	—	—	15	Mbps	VDDIO = 3.3V, CLOAD = 30 pF(MAX)
				—	—	15	Mbps	VDDIO = 1.8V, CLOAD = 30 pF(MAX)
UT_19A			Synchronous Mode X2 mode	—	—	25	Mbps	VDDIO = 3.3V, CLOAD = 30 pF(MAX)
				—	—	24	Mbps	VDDIO = 1.8V, CLOAD = 30 pF(MAX)
UT_21A			Synchronous Mode X1 mode	—	—	25	Mbps	VDDIO = 3.3V, CLOAD = 30 pF(MAX)
				—	—	24	Mbps	VDDIO = 1.8V, CLOAD = 30 pF(MAX)
UT_1B	FBRATE_HS	Baud Rate - High Speed SERCOM	Asynchronous SAMPR = 16x mode	—	—	10	Mbps	VDDIO = 3.3V, CLOAD = 30 pF(MAX)
				—	—	10	Mbps	VDDIO = 1.8V, CLOAD = 30 pF(MAX)
UT_3B			Asynchronous SAMPR = 8x mode	—	—	20	Mbps	VDDIO = 3.3V, CLOAD = 30 pF(MAX)
				—	—	20	Mbps	VDDIO = 1.8V, CLOAD = 30 pF(MAX)
UT_5B			Asynchronous SAMPR = 3x mode	—	—	25	Mbps	VDDIO = 3.3V, CLOAD = 30 pF(MAX)
				—	—	25	Mbps	VDDIO = 1.8V, CLOAD = 30 pF(MAX)
UT_19B			Synchronous Mode X2 mode	—	—	25	Mbps	VDDIO = 3.3V, CLOAD = 30 pF(MAX)
				—	—	24	Mbps	VDDIO = 1.8V, CLOAD = 30 pF(MAX)
UT_21B			Synchronous Mode X1 mode	—	—	25	Mbps	VDDIO = 3.3V, CLOAD = 30 pF(MAX)
				—	—	24	Mbps	VDDIO = 1.8V, CLOAD = 30 pF(MAX)
UT_23	FUSART	USART max GCLK_SERCOM		—	—	FCLK_23	MHz	—
UT_25	FXCK	USART External Clock Input		—	—	25	MHz	—

52.24. I²S Electrical Specifications

Figure 52-19. I²S Host Mode AC Timing Diagram

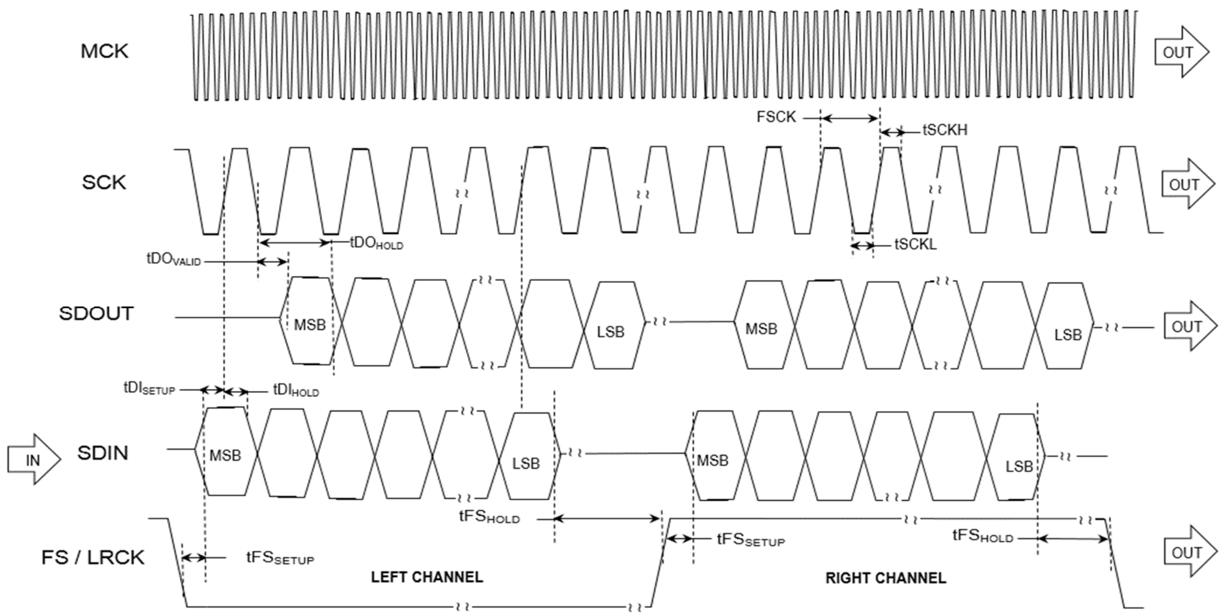


Table 52-34. I²S Host Mode AC Electrical Specifications

AC CHARACTERISTICS			Standard Operating Conditions: VDDREG=VDDIO=AVDD 1.71V to 3.63V (unless otherwise stated) **Operating temperature:** **-40°C ≤ TA ≤ +85°C for Industrial**				
Param. No.	Symbol	Characteristics	Min.	Typ.	Max.	Units	Conditions
I2S_11	fM_SCK	Host SCK Frequency	—	—	24.576	MHz	VDDIOx = 1.8V, CLOAD = 30 pF(MAX)
			—	—	24.576		VDDIOx = 3.3V, CLOAD = 30 pF(MAX)
I2S_13	tSCKL	SCK Output Low Time	1/(2*fM_SCK)	—	—	μs	—
I2S_15	tSCKH	SCK Output High Time	1/(2*fM_SCK)	—	—	μs	—
I2S_17	tSCKR	SCK Rise Time	See I/O Specifications DI_25 or DI_29			ns	—
I2S_19	tSCKF	SCK Fall Time	See I/O Specificiations DI_27 or DI_31			ns	—
I2S_21	tM_FSVALID	Host Frame Sync Valid	—	—	7	ns	VDDIOx = 1.8V, CLOAD = 30 pF(MAX)
			—	—	5	ns	VDDIOx = 3.3V, CLOAD = 30 pF(MAX)
I2S_23	tM_FSHOLD	Host Frame Sync Hold	0	—	—	ns	VDDIOx = 1.8V, CLOAD = 30 pF(MAX)
			0	—	—	ns	VDDIOx = 3.3V, CLOAD = 30 pF(MAX)
I2S_25	tM_DISETUP	Host Data Input Setup	16	—	—	ns	VDDIOx = 1.8V, CLOAD = 30 pf(MAX)
			10	—	—	ns	VDDIOx = 3.3V, CLOAD = 30 pF(MAX)

Table 52-34. I²S Host Mode AC Electrical Specifications (continued)

AC CHARACTERISTICS			Standard Operating Conditions: VDDREG=VDDIO=AVDD 1.71V to 3.63V (unless otherwise stated) **Operating temperature:** **-40°C ≤ TA ≤ +85°C for Industrial**				
Param. No.	Symbol	Characteristics	Min.	Typ.	Max.	Units	Conditions
I2S_27	tM_DIHOLD	Host Data Input Hold	0	—	—	ns	VDDIOx = 1.8V, CLOAD = 30 pF(MAX)
			0	—	—	ns	VDDIOx = 3.3V, CLOAD = 30 pF(MAX)
I2S_29	tM_DOVALID	Host Data Output Valid	—	—	7	ns	VDDIOx = 1.8V, CLOAD = 30 pF(MAX)
			—	—	5	ns	VDDIOx = 3.3V, CLOAD = 30 pF(MAX)
I2S_31	tM_DOHOLD	Host Data Output Hold	0	—	—	ns	VDDIOx = 1.8V, CLOAD = 30 pF(MAX)
			0	—	—	ns	VDDIOx = 3.3V, CLOAD = 30 pF(MAX)
I2S_33	fGCLK_I2S	I²S Max GLCK Input Clock Frequency	—	—	FCLK_29	MHz	—

Figure 52-20. I²S Client Mode AC Timing Diagram

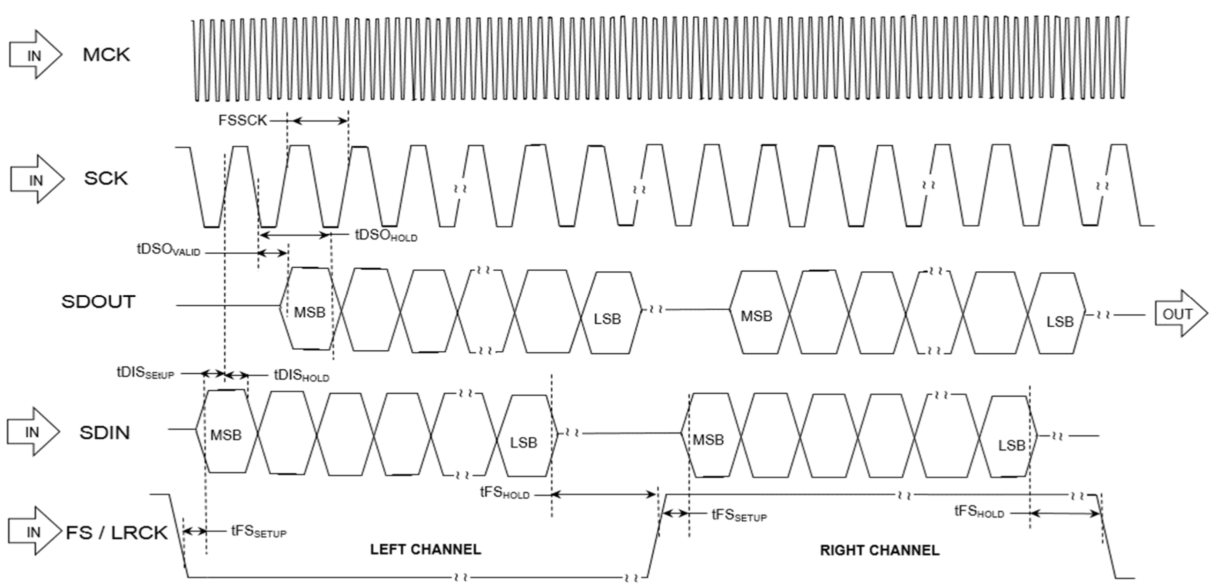


Table 52-35. I²S Client Mode AC Electrical Specifications

AC CHARACTERISTICS			Standard Operating Conditions: VDDREG=VDDIO=AVDD 1.71V to 3.63V (unless otherwise stated) **Operating temperature:** **-40°C ≤ TA ≤ +85°C for Industrial**				
Param. No.	Symbol	Characteristics	Min.	Typ.	Max.	Units	Conditions
I2S_41	fS_SCK	SCK Client Frequency	—	—	24.576	MHz	VDDIOx = 1.8V, CLOAD = 30 pF(MIN)
			—	—	24.576		VDDIOx = 3.3v, CLOAD = 30 pF(MIN)
I2S_43	tS_FSVALID	Frame Sync Valid	20	—	—	ns	VDDIOx = 1.8V, CLOAD = 30 pF(MIN)
			13	—	—	ns	VDDIOx = 3.3v, CLOAD = 30 pF(MIN)
I2S_45	tS_FSHOLD	Frame Sync Hold	2	—	—	ns	VDDIOx = 1.8V, CLOAD = 30 pF(MIN)
			2	—	—	ns	VDDIOx = 3.3v, CLOAD = 30 pF(MIN)
I2S_47	tS_DISSETUP	Data Input Client Setup	2	—	—	ns	VDDIOx = 1.8V, CLOAD = 30 pF(MIN)
			1	—	—	ns	VDDIOx = 3.3v, CLOAD = 30 pF(MIN)
I2S_49	tS_DISHOLD	Data Input Client Hold	0	—	—	ns	VDDIOx = 1.8V, CLOAD = 30 pF(MIN)
			0	—	—	ns	VDDIOx = 3.3v, CLOAD = 30 pF(MIN)
I2S_51	tS_DSOVALID	Data Output Client Valid	—	—	20	ns	VDDIOx = 1.8V, CLOAD = 30 pF(MIN)
			—	—	13	ns	VDDIOx = 3.3v, CLOAD = 30 pF(MIN)
I2S_53	tS_DSOHOLD	Data Output Client Hold	5	—	—	ns	VDDIOx = 1.8V, CLOAD = 30 pF(MIN)
			4	—	—	ns	VDDIOx = 3.3v, CLOAD = 30 pF(MIN)

52.25. I²C Electrical Specifications

Figure 52-21. I²C Start/Stop Bits Host Mode AC Timing Diagrams

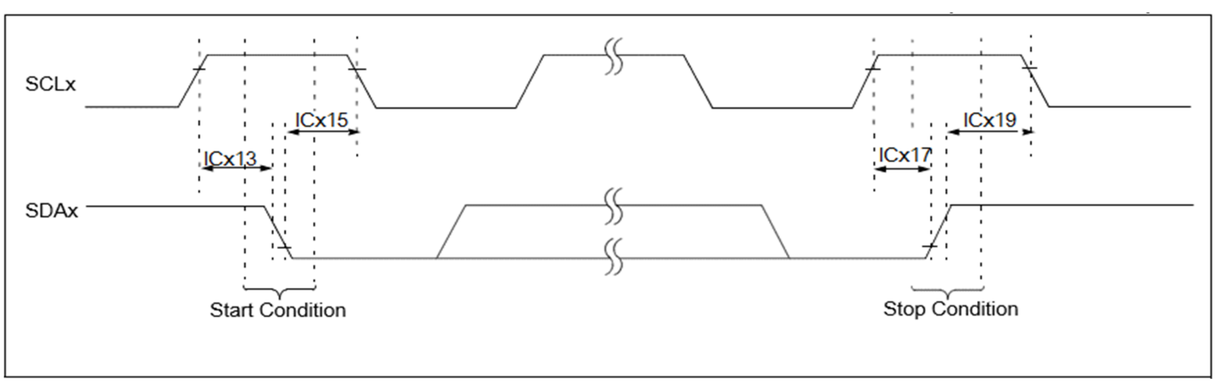


Figure 52-22. I²C Bus Data Host Mode AC Timing Diagrams

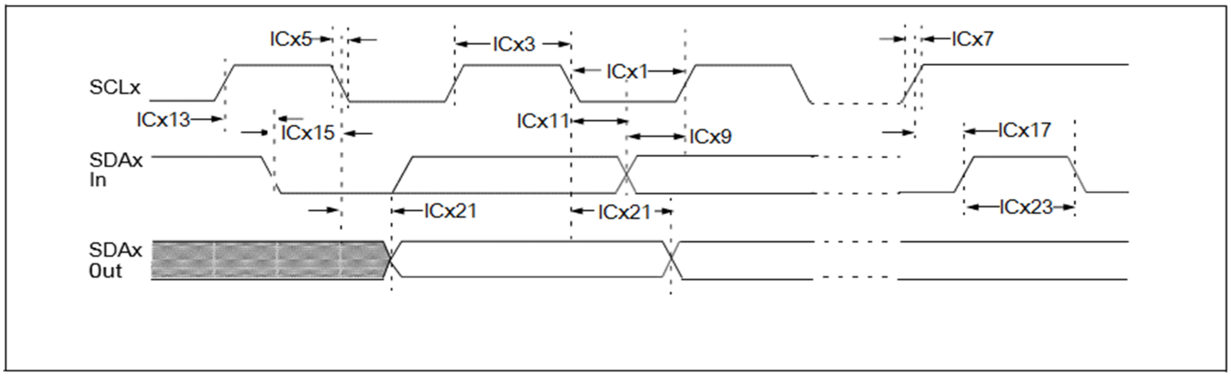


Table 52-36. I²C Host Mode AC Electrical Specifications

AC CHARACTERISTICS				Standard Operating Conditions: VDDREG=VDDIO=AVDD 1.71V to 3.63V (unless otherwise stated) **Operating temperature:** **-40°C ≤ TA ≤ +85°C for Industrial**			
Param. No.	Symbol	Characteristics		Min.	Max.	Units	Conditions
I2CM_1	TL0:SCL	Host Clock Low Time	100 kHz mode	4.7	—	µs	VDDIOx = 3.3V, IPULL-UP = 3mA, CLOAD = 400 pF
			400 kHz mode	1.3	—	µs	
			1 MHz mode	0.5	—	µs	VDDIOx = 3.3V, IPULL-UP = 20 mA, CLOAD = 550 pF
			3.4 MHz mode	160	—	ns	VDDIOx = 3.3V, IPULL-UP = 12 mA, CLOAD = 100 pF
I2CM_3	THI:SCL	Host Clock High Time	100 kHz mode	4	—	µs	VDDIOx = 3.3V, IPULL-UP = 3mA, CLOAD = 400 pF
			400 kHz mode	0.6	—	µs	
			1 MHz mode	0.26	—	µs	VDDIOx = 3.3V, IPULL-UP = 20 mA, CLOAD = 550 pF
			3.4 MHz mode	60	—	ns	VDDIOx = 3.3V, IPULL-UP = 12 mA, CLOAD = 100 pF
I2CM_5	TF:SCL	SDAx and SCLx Fall Time	100 kHz mode	—	300	ns	VDDIOx = 3.3V, IPULL-UP = 3mA, CLOAD = 400 pF
			400 kHz mode	—	300	ns	
			1 MHz mode	—	120	ns	VDDIOx = 3.3V, IPULL-UP = 20 mA, CLOAD = 550 pF
			3.4 MHz mode	—	40	ns	VDDIOx = 3.3V, IPULL-UP = 12 mA, CLOAD = 100 pF

Table 52-36. I²C Host Mode AC Electrical Specifications (continued)

AC CHARACTERISTICS				Standard Operating Conditions: VDDREG=VDDIO=AVDD 1.71V to 3.63V (unless otherwise stated) **Operating temperature:** **-40°C ≤ TA ≤ +85°C for Industrial**			
Param. No.	Symbol	Characteristics		Min.	Max.	Units	Conditions
I2CM_7	TR:SCL	SDAx and SCLx Rise Time	100 kHz mode	—	1000	ns	VDDIOx = 3.3V, IPULL-UP = 3mA, CLOAD = 400 pF
			400 kHz mode	—	300	ns	
			1 MHz mode	—	120	ns	VDDIOx = 3.3V, IPULL-UP = 20 mA, CLOAD = 550 pF
			3.4 MHz mode	—	40	ns	VDDIOx = 3.3V, IPULL-UP = 12 mA, CLOAD = 100 pF
I2CM_9	TSU:DAT	Data Setup Time	100 kHz mode	250	—	ns	VDDIOx = 3.3V, IPULL-UP = 3mA, CLOAD = 400 pF
			400 kHz mode	100	—	ns	
			1 MHz mode	50	—	ns	VDDIOx = 3.3V, IPULL-UP = 20 mA, CLOAD = 550 pF
			3.4 MHz mode	10	—	ns	VDDIOx = 3.3V, IPULL-UP = 12 mA, CLOAD = 100 pF
I2CM_11	THD:DAT [(1)]	Data Hold Time	100 kHz mode	300	—	ns	VDDIOx = 3.3V, IPULL-UP = 3mA, CLOAD = 400 pF
			400 kHz mode	300	—	ns	
			1 MHz mode	300	—	ns	VDDIOx = 3.3V, IPULL-UP = 20 mA, CLOAD = 550 pF
			3.4 MHz mode	5	—	ns	VDDIOx = 3.3V, IPULL-UP = 12 mA, CLOAD = 100 pF
I2CM_13	TSU:STA	Start Condition Setup Time	100 kHz mode	4.7	—	μs	VDDIOx = 3.3V, IPULL-UP = 3mA, CLOAD = 400 pF
			400 kHz mode	0.6	—	μs	
			1 MHz mode	0.26	—	μs	VDDIOx = 3.3V, IPULL-UP = 20 mA, CLOAD = 550 pF
			3.4 MHz mode	160	—	ns	VDDIOx = 3.3V, IPULL-UP = 12 mA, CLOAD = 100 pF

Table 52-36. I²C Host Mode AC Electrical Specifications (continued)

AC CHARACTERISTICS				Standard Operating Conditions: VDDREG=VDDIO=AVDD 1.71V to 3.63V (unless otherwise stated) **Operating temperature:** **-40°C ≤ TA ≤ +85°C for Industrial**			
Param. No.	Symbol	Characteristics		Min.	Max.	Units	Conditions
I2CM_15	THD:STA	Start Condition Hold Time	100 kHz mode	4	—	µs	VDDIOx = 3.3V, IPULL-UP = 3mA, CLOAD = 400 pF
			400 kHz mode	0.6	—	µs	
			1 MHz mode	0.26	—	µs	VDDIOx = 3.3V, IPULL-UP = 20 mA, CLOAD = 550 pF
			3.4 MHz mode	160	—	ns	VDDIOx = 3.3V, IPULL-UP = 12 mA, CLOAD = 100 pF
I2CM_17	TSU:STO	Stop Condition Setup Time	100 kHz mode	4	—	µs	VDDIOx = 3.3V, IPULL-UP = 3mA, CLOAD = 400 pF
			400 kHz mode	0.6	—	µs	
			1 MHz mode	0.26	—	µs	VDDIOx = 3.3V, IPULL-UP = 20 mA, CLOAD = 550 pF
			3.4 MHz mode	160	—	ns	VDDIOx = 3.3V, IPULL-UP = 12 mA, CLOAD = 100 pF
I2CM_21	TAA:SCL	Output Valid from Clock	100 kHz mode	—	3.45	µs	VDDIOx = 3.3V, IPULL-UP = 3mA, CLOAD = 400 pF
			400 kHz mode	—	0.9	µs	
			1 MHz mode	—	0.45	µs	VDDIOx = 3.3V, IPULL-UP = 20 mA, CLOAD = 550 pF
			3.4 MHz mode	—	100	ns	VDDIOx = 3.3V, IPULL-UP = 12 mA, CLOAD = 100 pF
I2CM_23	TBF:SDA [(2)]	Bus Free Time	100 kHz mode	4.7	—	µs	VDDIOx = 3.3V, IPULL-UP = 3mA, CLOAD = 400 pF
			400 kHz mode	1.3	—	µs	
			1 MHz mode	0.5	—	µs	VDDIOx = 3.3V, IPULL-UP = 20 mA, CLOAD = 550 pF
			3.4 MHz mode	160	—	ns	VDDIOx = 3.3V, IPULL-UP = 12 mA, CLOAD = 100 pF

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 52-23. I²C Start/Stop Bits Client Mode AC Timing Diagrams

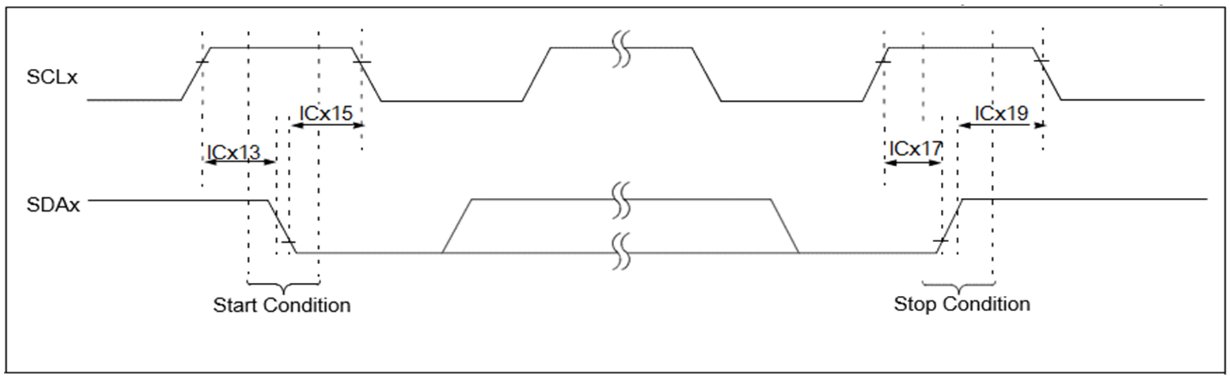


Figure 52-24. I²C Bus Data Client Mode AC Timing Diagrams

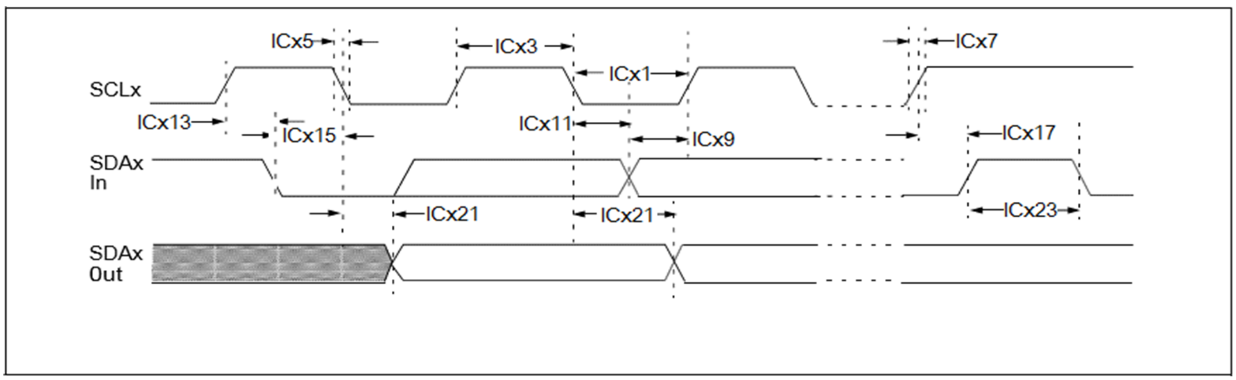


Table 52-37. I²C Client Mode AC Electrical Specifications

AC CHARACTERISTICS_				Standard Operating Conditions: VDDREG=VDDIO=AVDD 1.71V to 3.63V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial			
Param. No.	Symbol	Characteristics		Min.	Max.	Units	Conditions
I2CS_1	TL0:SCL	Client Clock Low Time	100 kHz mode	4.7	—	µs	VDDIOx = 3.3V, IPULL-UP = 3mA, CLOAD = 400 pF
			400 kHz mode	1.3	—	µs	
			1 MHz mode	0.5	—	µs	VDDIOx = 3.3V, IPULL-UP = 20 mA, CLOAD = 550 pF
			3.4 MHz mode	160	—	ns	VDDIOx = 3.3V, IPULL-UP = 12 mA, CLOAD = 100 pF
I2CS_3	THI:SCL	Client Clock High Time	100 kHz mode	4	—	µs	VDDIOx = 3.3V, IPULL-UP = 3mA, CLOAD = 400 pF
			400 kHz mode	0.6	—	µs	
			1 MHz mode	0.26	—	µs	VDDIOx = 3.3V, IPULL-UP = 20 mA, CLOAD = 550 pF
			3.4 MHz mode	60	—	ns	VDDIOx = 3.3V, IPULL-UP = 12 mA, CLOAD = 100 pF

Table 52-37. I²C Client Mode AC Electrical Specifications (continued)

AC CHARACTERISTICS_				Standard Operating Conditions: VDDREG=VDDIO=AVDD 1.71V to 3.63V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial			
Param. No.	Symbol	Characteristics		Min.	Max.	Units	Conditions
I2CS_5	TF:SCL	SDAx and SCLx Fall Time	100 kHz mode	—	300	ns	VDDIOx = 3.3V, IPULL-UP = 3mA, CLOAD = 400 pF
			400 kHz mode	—	300	ns	
			1 MHz mode	—	120	ns	VDDIOx = 3.3V, IPULL-UP = 20 mA, CLOAD = 550 pF
			3.4 MHz mode	—	40	ns	VDDIOx = 3.3V, IPULL-UP = 12 mA, CLOAD = 100 pF
I2CS_7	TR:SCL	SDAx and SCLx Rise Time	100 kHz mode	—	1000	ns	VDDIOx = 3.3V, IPULL-UP = 3mA, CLOAD = 400 pF
			400 kHz mode	—	300	ns	
			1 MHz mode	—	120	ns	VDDIOx = 3.3V, IPULL-UP = 20 mA, CLOAD = 550 pF
			3.4 MHz mode	—	40	ns	VDDIOx = 3.3V, IPULL-UP = 12 mA, CLOAD = 100 pF
I2CS_9	TSU:DAT	Data Setup Time	100 kHz mode	250	—	ns	VDDIOx = 3.3V, IPULL-UP = 3mA, CLOAD = 400 pF
			400 kHz mode	100	—	ns	
			1 MHz mode	50	—	ns	VDDIOx = 3.3V, IPULL-UP = 20 mA, CLOAD = 550 pF
			3.4 MHz mode	10	—	ns	VDDIOx = 3.3V, IPULL-UP = 12 mA, CLOAD = 100 pF
I2CS_11	THD:DAT [(1)]	Data Hold Time	100 kHz mode	300	—	ns	VDDIOx = 3.3V, IPULL-UP = 3mA, CLOAD = 400 pF
			400 kHz mode	300	—	ns	
			1 MHz mode	300	—	ns	VDDIOx = 3.3V, IPULL-UP = 20 mA, CLOAD = 550 pF
			3.4 MHz mode	5	—	ns	VDDIOx = 3.3V, IPULL-UP = 12 mA, CLOAD = 100 pF
I2CS_13	TSU:STA	Start Condition Setup Time	100 kHz mode	4.7	—	μs	VDDIOx = 3.3V, IPULL-UP = 3mA, CLOAD = 400 pF
			400 kHz mode	0.6	—	μs	
			1 MHz mode	0.26	—	μs	VDDIOx = 3.3V, IPULL-UP = 20 mA, CLOAD = 550 pF
			3.4 MHz mode	160	—	ns	VDDIOx = 3.3V, IPULL-UP = 12 mA, CLOAD = 100 pF

Table 52-37. I²C Client Mode AC Electrical Specifications (continued)

AC CHARACTERISTICS_				Standard Operating Conditions: VDDREG=VDDIO=AVDD 1.71V to 3.63V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial			
Param. No.	Symbol	Characteristics		Min.	Max.	Units	Conditions
I2CS_15	THD:STA	Start Condition Hold Time	100 kHz mode	4	—	µs	VDDIOx = 3.3V, IPULL-UP = 3mA, CLOAD = 400 pF
			400 kHz mode	0.6	—	µs	
			1 MHz mode	0.26	—	µs	VDDIOx = 3.3V, IPULL-UP = 20 mA, CLOAD = 550 pF
			3.4 MHz mode	160	—	ns	VDDIOx = 3.3V, IPULL-UP = 12 mA, CLOAD = 100 pF
I2CS_17	TSU:ST0	Stop Condition Setup Time	100 kHz mode	4	—	µs	VDDIOx = 3.3V, IPULL-UP = 3mA, CLOAD = 400 pF
			400 kHz mode	0.6	—	µs	
			1 MHz mode	0.26	—	µs	VDDIOx = 3.3V, IPULL-UP = 20 mA, CLOAD = 550 pF
			3.4 MHz mode	160	—	ns	VDDIOx = 3.3V, IPULL-UP = 12 mA, CLOAD = 100 pF
I2CS_21	TAA:SCL	Output Valid from Clock	100 kHz mode	—	3.45	µs	VDDIOx = 3.3V, IPULL-UP = 3mA, CLOAD = 400 pF
			400 kHz mode	—	0.9	µs	
			1 MHz mode	—	0.45	µs	VDDIOx = 3.3V, IPULL-UP = 20 mA, CLOAD = 550 pF
			3.4 MHz mode	—	100	ns	VDDIOx = 3.3V, IPULL-UP = 12 mA, CLOAD = 100 pF
I2CS_23	TBF:SDA [(2)]	Bus Free Time	100 kHz mode	4.7	—	µs	VDDIOx = 3.3V, IPULL-UP = 3mA, CLOAD = 400 pF
			400 kHz mode	1.3	—	µs	
			1 MHz mode	0.5	—	µs	VDDIOx = 3.3V, IPULL-UP = 20 mA, CLOAD = 550 pF
			3.4 MHz mode	160	—	ns	VDDIOx = 3.3V, IPULL-UP = 12 mA, CLOAD = 100 pF

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).

52.26. SQI/QSPI Electrical Specifications

Figure 52-25. QSPI SDR Host Mode 0,1,2,3 Module Timing Diagram

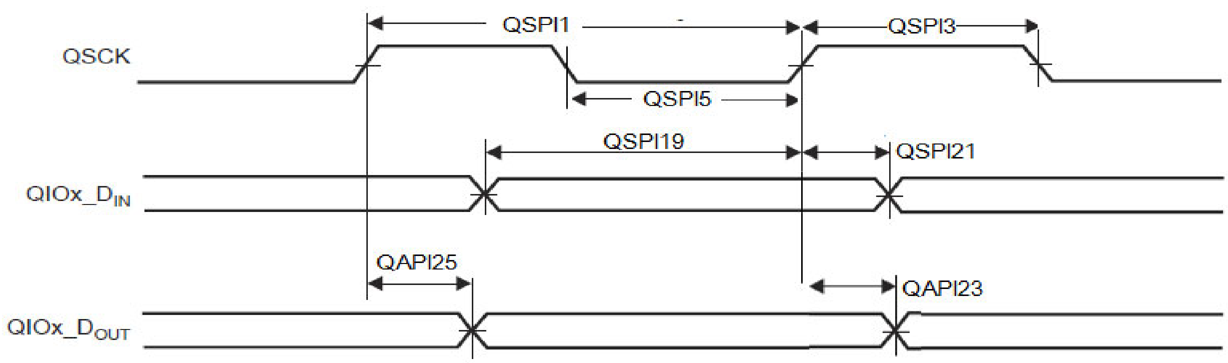


Figure 52-26. QSPI DDR Mode 0 Write Timing Diagram

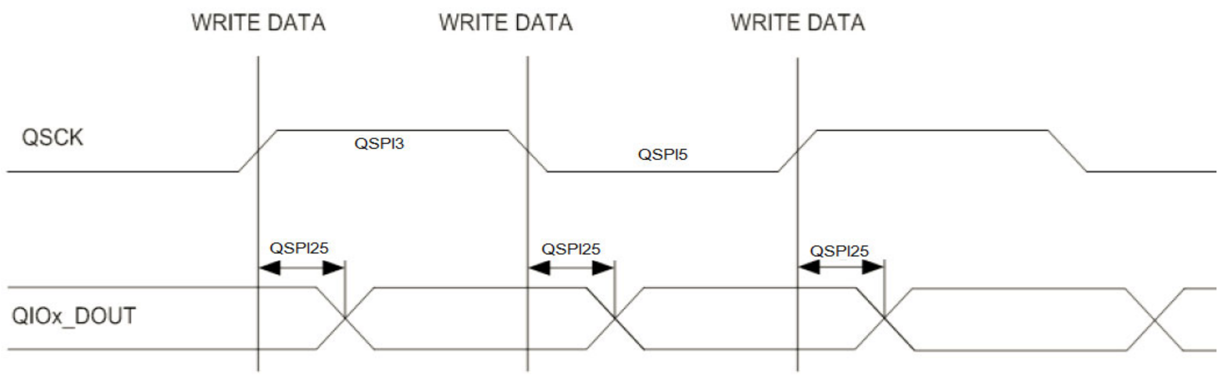


Figure 52-27. QSPI_DDR Mode 0 Read Timing Diagram

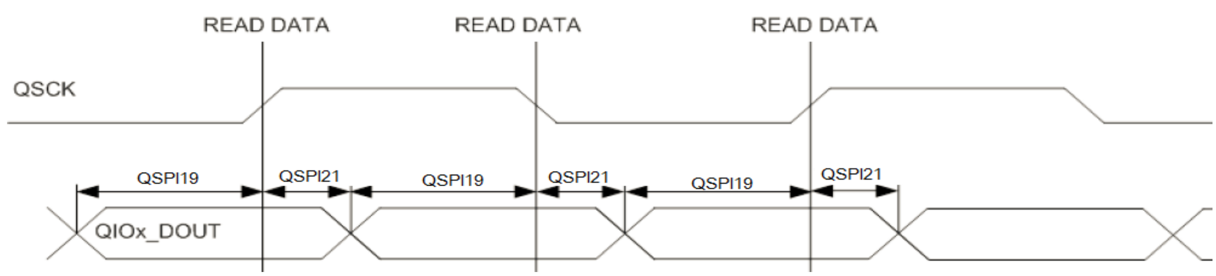


Table 52-38. QSPI Module Electrical Specifications (1)

AC CHARACTERISTICS				Standard Operating Conditions: VDDREG=VDDIO=AVDD 1.71V to 3.63V (unless otherwise stated) **Operating temperature:** **-40°C ≤ TA ≤ +85°C for Industrial**				
Param. No.	**Symbol**	**Characteristics**		**Min.**	**Typ.**	**Max.**	**Units**	**Conditions**
QSPI_1	FCLK	SQI Serial Clock Frequency		—	—	83	MHz	SDR Read VDDIOx = 3.3V, CLOAD = 30 pF(MAX)
				—	—	69	MHz	SDR Read VDDIOx = 1.8V, CLOAD = 30 pF(MAX)
				—	—	118	MHz	SDR Write VDDIOx = 3.3V, CLOAD = 30 pF(MAX)
				—	—	100	MHz	SDR Write VDDIOx = 1.8V, CLOAD = 30 pF(MAX)
				—	—	66	MHz	DDR Read VDDIOx = 3.3V, CLOAD = 30 pF(MAX)
				—	—	56	MHz	DDR Read VDDIOx = 1.8V, CLOAD = 30 pF(MAX)
QSPI_3	TSCKH	Serial Clock High Time		1/(2*FCLK)	—	—	µs	—
QSPI_5	TSCKL	Serial Clock Low Time		1/(2*FCLK)	—	—	µs	—
QSPI_7	TSCKR	Serial Clock Rise Time		—	—	DI_29	ns	See parameter in I/O Specifications
QSPI_9	TSCKF	Serial Clock Fall Time		—	—	DI_31	ns	See parameter in I/O Specifications
QSPI_11	TCSS	CS Active Setup Time	SDR	14	—	—	ns	—
			DDR	16	—	—	ns	—
QSPI_13	TCSH	CS Active Hold Time		0	—	—	ns	—
QSPI_15	TCHS	CS Not Active Setup Time	SDR	7	—	—	ns	—
			DDR	9	—	—	ns	—
QSPI_17	TCHH	CS Not Active Hold Time		0	—	—	ns	—
QSPI_19	TDIS	Data In Setup Time	SDR	1	—	—	ns	—
			DDR	2	—	—	ns	—
QSPI_21	TDIH	Data In Hold Time	SDR	1	—	—	ns	—
			DDR	1	—	—	ns	—
QSPI_23	TDOH	Data Out Hold		0	—	—	ns	—
QSPI_25	TDOV	Data Out Valid	SDR	—	—	2	ns	—
			DDR	—	—	5	ns	—
QSPI_27	QSPI_GCLK	GCLK_QSPI		—	—	FCLK_73	MHz	—

Note:
1. Assumes VDDIOx(min) and 30 pF external load on all SQI pins unless otherwise noted.

52.27. Controller Area Network (CAN) Electrical Specifications

Figure 52-28. CANx Module AC Timing Diagram

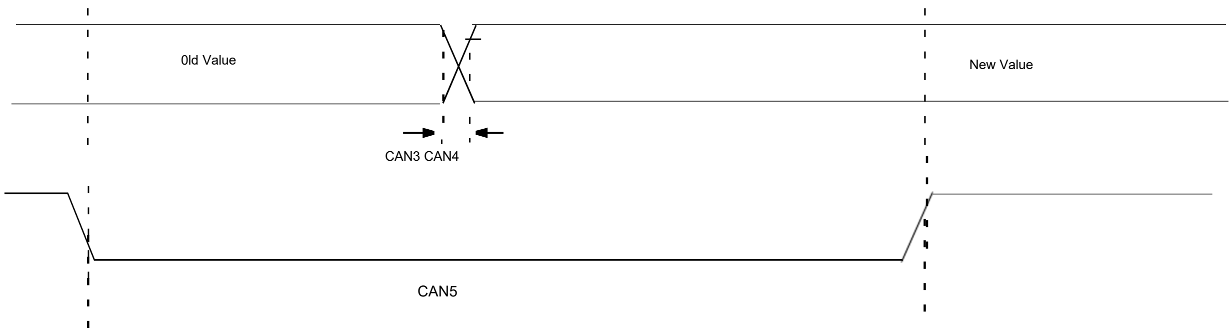


Table 52-39. CANx Module AC Electrical Specifications

AC CHARACTERISTICS			Standard Operating Conditions: VDDREG=VDDIO=AVDD 1.71V to 3.63V (unless otherwise stated) **Operating temperature:** **-40°C ≤ TA ≤ +85°C for Industrial**				
Param. No.	Symbol	Characteristics	Min.	Typ.	Max.	Units	Conditions
CAN_1	CANRATE	CAN data rate	—	—	8	Mbps	VDDIOx = 3.3V, CLOAD = 30 pF(MAX)
			—	—	8		VDDIOx = 1.8V, CLOAD = 30 pF(MAX)
CAN_3	CANFALL	Port Output Fall Time	—	—	DI_27 or DI_31	ns	See parameter in I/O Specifications
CAN_4	CANRISE	Port Output Rise Time	—	—	DI_25 or DI_29	ns	See parameter in I/O Specifications
CAN_7	fCAN_GCLK	CANx Input clk freq, GCLK_CAN	—	—	FCLK_25	MHz	VDDIOx = 1.8V

Note:
1. VDDIOx(min) and 30pF external load on all CAN pins unless otherwise noted.

52.28. Timer Counter for Control Applications (TCC) Electrical Specifications

Figure 52-29. TCCx Timer Capture Input AC Timing

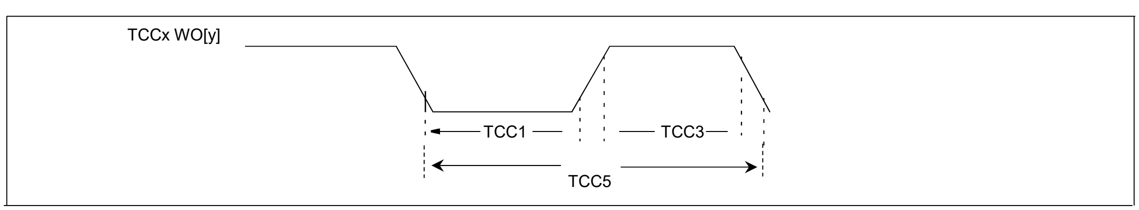


Figure 52-30. TCCx Timer Compare Output Module AC Timing Diagrams

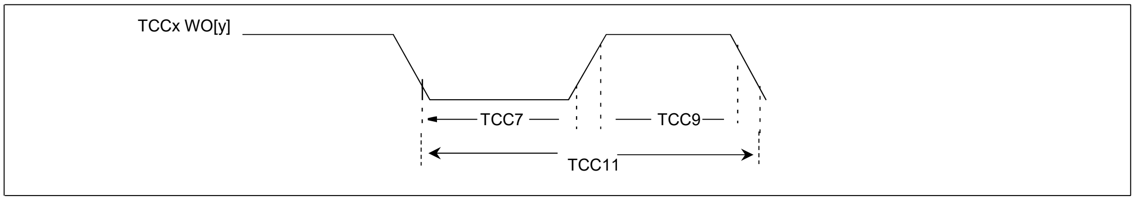


Figure 52-31. TCCx Timer Compare Fault Output Module AC Timing Diagrams

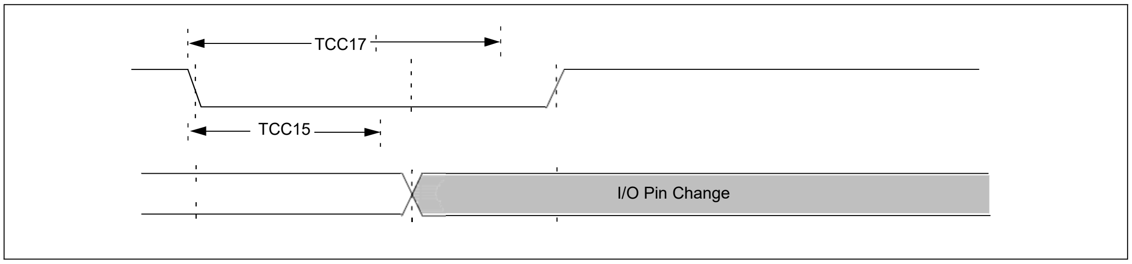


Table 52-40. TCCx Timer Capture Module AC Electrical Specifications

AC CHARACTERISTICS			Standard Operating Conditions: VDDREG=VDDIO=AVDD 1.71V to 3.63V (unless otherwise stated) **Operating temperature:** **-40°C ≤ TA ≤ +85°C for Industrial**				
Param. No.	Symbol	Characteristics	Min.	Typ.	Max.	Units	Conditions
TCC_1	TCCINLOW	Capture TCCx Input Low Time	2/fGLK_TCCx	—	—	µs	VDDIOx(min) and meet TCC_5 spec
TCC_3	TCCINHIGH	Capture TCCx Input High Time	2/fGLK_TCCx	—	—	µs	VDDIOx(min) and meet TCC_5 spec
TCC_5	TCCINPERIOD	Capture Input Period	4/fGLK_TCCx	—	—	µs	VDDIOx(min)
TCC_7	TCCOUTLOW	Compare TCCx Output Low Time	3*DI_27 or 3*DI_31	—	—	ns	VDDIOx(min) and meet TCC_11 spec
TCC_9	TCCOUTHIGH	Compare TCCx Output High Time	3*DI_25 or 3*DI_29	—	—	ns	VDDIOx(min) and meet TCC_11 spec
TCC_11	TCCOUTPERIOD	Compare Output Period	TCC_7+TCC_9	—	—	ns	VDDIOx(min)
TCC_13	fGCLK_TCCx	TCC peripheral module clock frequency	—	—	FCLK_35	MHz	
TCC_15	TCCFD	Fault Input to I/O Pin Change (1)	—	—	41	ns	
TCC_17	TCCFLT	Fault Input Pulse Width	10	—	—	ns	

Note:

1. Assumes VDDIOx(min) and 20 pF external load on TCC Fault Input I/O pin.

52.29. Universal Serial Bus (USB) Electrical Specifications

Table 52-41. USB Full-Speed or Low-Speed AC Electrical Specifications

AC CHARACTERISTICS			Standard Operating Conditions: VDDIO 3.0V to 3.6V (unless otherwise stated) **Operating temperature:** **-40°C ≤ TA ≤ +85°C for Industrial**				
Param. No.	Symbol	Characteristics [1]	Min.	Typ.	Max.	Units	Conditions
USB_1	VDDUSB[4]	USB Transceiver Voltage	3	—	3.6	V	Voltage on VDDIOx must be in this range for proper USB operation
Clocks							
USB_23	USBFSCLKS[2]	USBFS Clock Source	—	48	—	MHz	fGCLK_USBFS = 48 MHz
USB_24	USBHSCLKS[3]	USBHS Clock Source	—	12 or 24	—	MHz	—
USB_25	USBFSAHB	Minimum AHB Clock for USBFS Operations	12	—	—	MHz	—
USB_26	USBHSAHB	Min. AHB Clock for USBHS Operations	60	—	—	MHz	—

Notes:
1. USBFS: Supports full-speed and low-speed communication. Compliant with USB 2.0 specification. USBHS: Supports high-speed, full-speed, and low-speed communication. Compliant with USB 2.0 specification.
2. USBFS: Required clock accuracy: full-speed = ±0.25%, low-speed = ±1.5%.
3. USBHS: XOSC clock source. Required clock accuracy: ±0.05% for all speeds.
4. USBFS voltage range: when TA = 125 °C, the voltage range will be limited to 3.3V ±5%.

Table 52-42. USB Full-Speed or Low-Speed Clock Configuration

USBFS Clock configuration			Standard Operating Conditions: VDDIO 3.0V to 3.6V (unless otherwise stated) **Operating temperature:** **-40°C ≤ TA ≤ +85°C for Extended Temp**	
Param. No.	Symbol	Characteristics	Device mode operation.	Host mode operation.
USB_30	USBFS_GCLK_SRC	DFLL48M Open loop	No	No
		DFLL48M Close loop, Ref. internal OSC source	No	No
		DFLL48M close loop, Ref. external XOSC/XOSC32K source	Yes	No
		DFLL48M close loop, Ref. SOF (USB recovery mode)	Yes	N/A
		PLL internal OSC (DFLL48M)	No	No
		PLL external XOSC	Yes	Yes
		XOSC Crystal	Yes	Yes

52.30. Parallel Capture Controller (PCC) Electrical Specifications

Table 52-43. Parallel Capture Controller (PCC) AC Electrical Specifications

AC CHARACTERISTICS				Standard Operating Conditions: VDDREG=VDDIO=AVDD 1.71V to 3.63V (unless otherwise stated) **Operating temperature:** **-40°C ≤ TA ≤ +85°C for Industrial**				
Param. No.	Symbol	Characteristics		Min.	Typ.	Max.	Units	Conditions
PCC_1	PCC_CLK [1]	External PCC Data Input Clock Frequency	PCC DSIZE = 2	—	—	20	MHz	VDDIOx(min)
PCC_1	PCC_CLK [1]	External PCC Data Input Clock Frequency	PCC DSIZE = 1	—	—	10	MHz	VDDIOx(min)
PCC_1	PCC_CLK [1]	External PCC Data Input Clock Frequency	PCC DSIZE = 0	—	—	5	MHz	VDDIOx(min)
PCC_3	PCCHIGH	PCC Data Clock input High time		500/PCC_CLK	—	—	ns	VDDIOx(min)
PCC_5	PCCLOW	PCC Data Clock input Low time		500/PCC_CLK	—	—	ns	VDDIOx(min)
PCC_7	PCCDSETUP	Clock to Data Setup Time		5	—	—	ns	VDDIOx(min)
PCC_9	PCCDHOLD	Clock to Data Hold Time		0	—	—	ns	VDDIOx(min)
PCC_11	PCCDENBSETUP [2]	Data Enable [1,2] Setup Time		3	—	—	ns	VDDIOx(min)
PCC_13	PCCDENBHOLD [3]	Data Enable [1,2] Hold Time		—	—	0	ns	VDDIOx(min)
PCC_15	CLK_APB_PCC [4]	Internal MCLK PCC Module Clock		—	—	fAPBn	MHz	VDDIOx(min)

Notes:

1. PCC ISIZE = 0 (8-bits). DMA descriptors used for continuous data acquisition, with highest priority for the DMA on the AHB bus (DMA.CTRLB.QOSx=2). fAPBn running at Max frequency (PCC_15).
2. PCC_DENBSETUP ≥ PCC_7.
3. PCC_DENBHOLD ≥ PCC_9.
4. CLK_APB_PCC_ ≥ (2 * PCC__CLK).

52.31. Non-Volatile Memory Controller (NVM) Electrical Specifications

Table 52-44. Flash NVM AC Electrical Specifications

AC CHARACTERISTICS				Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated) **Operating temperature:** **-40°C ≤ TA ≤ +85°C for Industrial**				
Param. No.	Symbol	Characteristics		Min.	Typ.	Max.	Units	Conditions
NVM_1	FRETEN	Flash Data Retention		20	—	—	Yrs	Under all conditions less than Absolute Maximum Ratings specifications
NVM_3	EP [3]	Cell Endurance (Flash Erase and Write Operation)		10k	—	—	Cycles	Under all conditions less than Absolute Maximum Ratings specifications
NVM_5	FREAD [1]	Flash Read	0 Wait States	—	—	20	MHz	FCR.CTRLA.AUTOWS = 0, FCR.CTRLA.ADRWS = 0 [2]
NVM_5	FREAD [1]	Flash Read	1 Wait States	—	—	40	MHz	FCR.CTRLA.AUTOWS = 0, FCR.CTRLA.ADRWS = 0 [2]
NVM_5	FREAD [1]	Flash Read	2 Wait States	—	—	60	MHz	FCR.CTRLA.AUTOWS = 0, FCR.CTRLA.ADRWS = 0 [2]
NVM_5	FREAD [1]	Flash Read	3 Wait States	—	—	80	MHz	FCR.CTRLA.AUTOWS = 0, FCR.CTRLA.ADRWS = 1 [2]
NVM_5	FREAD [1]	Flash Read	4 Wait States	—	—	100	MHz	FCR.CTRLA.AUTOWS = 0, FCR.CTRLA.ADRWS = 1 [2]
NVM_5	FREAD [1]	Flash Read	5 Wait States	—	—	120 [4]	MHz	FCR.CTRLA.AUTOWS = 0, FCR.CTRLA.ADRWS = 1 [2]

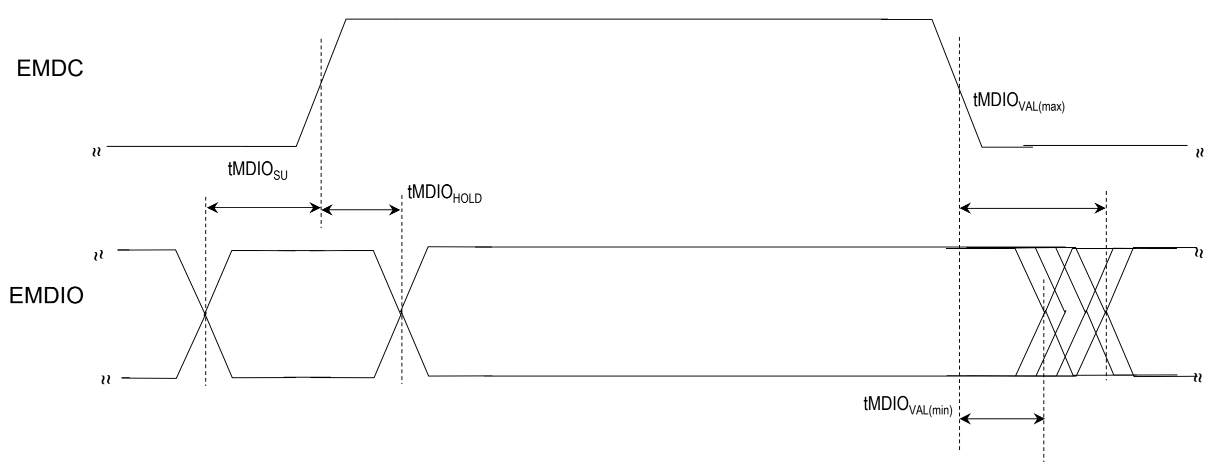
Table 52-44. Flash NVM AC Electrical Specifications (continued)

AC CHARACTERISTICS				Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated) **Operating temperature:** **-40°C ≤ TA ≤ +85°C for Industrial**				
Param. No.	Symbol	Characteristics		Min.	Typ.	Max.	Units	Conditions
NVM_7	TFW	Program Cycle Time	Write Double Word	—	—	20	μs	Under all conditions less than Absolute Maximum Ratings specifications
NVM_8	TFPP		Pre-Program Double Word	—	—	3.5	μs	
NVM_9	TCE		Erase Chip	—	—	20	ms	
NVM_11	TFEP		Erase Page	—	—	20	ms	
NVM_13	IDDPROG	Supply Current during Programming		—	—	PAI_401	mA	VDDIOx = 3.3V

Notes:
1. 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. Theses tables provide the device maximum operating frequency defined by the FWS field of the FCW CTRLA register when automatic wait states (AUTOWS) is disabled. This field defines the number of Wait states required to access the Embedded Flash Memory.
2. When frequency is lower than FCLK_1/2, the ADRWS bit of the NVMCTRL CTRLA register must be set to 0. When frequency is higher or equal to FCLK_1/2, the ADRWS bit of the NVMCTRL CTRLA register must be set to 1.
3. Cell Endurance is reached with Flash pre-programming option disabled. Enabling such option will improve the cell endurance but slow-down the programming time.
4. This frequency cannot be achieved in the temperature range +85°C < TA < +125°C.

52.32. Ethernet MAC (ETH) Electrical Specifications

Figure 52-32. MII Ethernet Module Output AC Timing Diagrams (Media-Independent Interface)



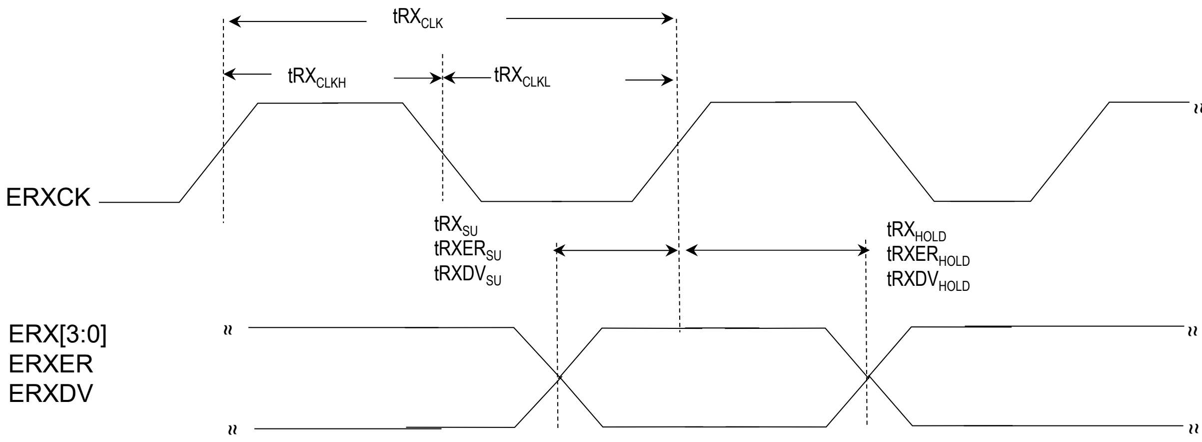


Figure 52-33. GMII Receive Ethernet Module AC Timing Diagrams (Gigabit Media-Independent Interface)

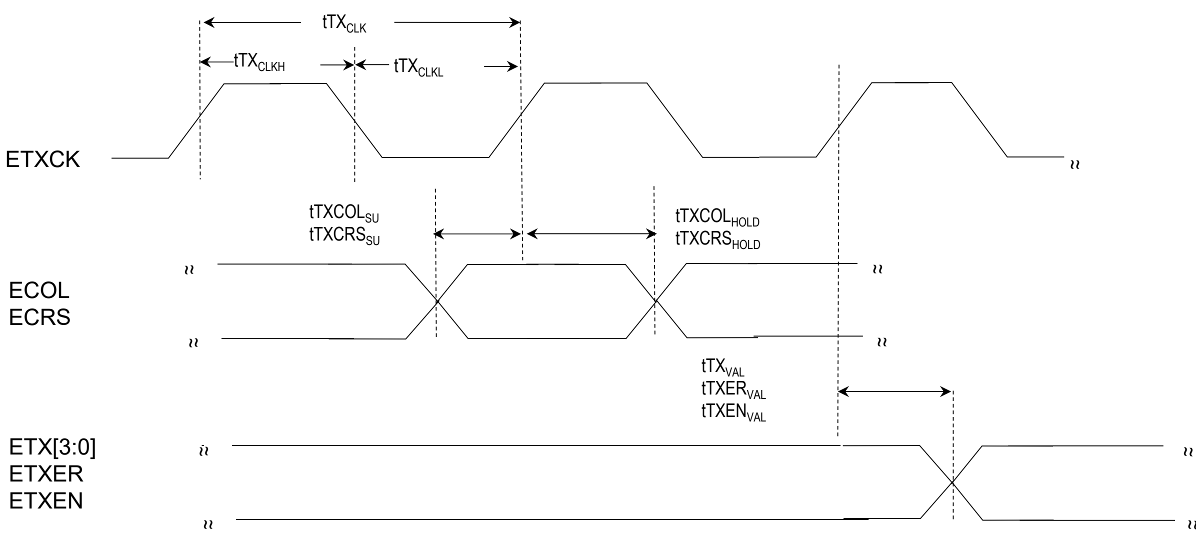


Figure 52-34. GMII Transmit Ethernet Module AC Timing Diagrams (Gigabit Media-Independent Interface)

Table 52-45. MII RX/TX Ethernet Module AC Electrical Specifications (Media-Independent Interface)

AC CHARACTERISTICS			Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated) **Operating temperature:** **-40°C ≤ TA ≤ +85°C for Industrial**				
Param. No.	Symbol	Characteristics	Min.	Typ.	Max.	Units	Conditions
MII Output AC TIMING Requirements							
ET_1	tMDIOSU	MDIO Set-up Time	-32	—	—	ns	VDDIO = 3.3V with CLOAD = 20 pF
ET_3	tMDIOHOLD	MDIO Hold Time	0	—	—	ns	
ET_5	tMDIOVAL	MDIO Output Valid Time	36	—	40	ns	
MII RX AC TIMING Requirements							

Table 52-45. MII RX/TX Ethernet Module AC Electrical Specifications (Media-Independent Interface) (continued)

AC CHARACTERISTICS			Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated) **Operating temperature:** **-40°C ≤ TA ≤ +85°C for Industrial**				
Param. No.	Symbol	Characteristics	Min.	Typ.	Max.	Units	Conditions
ET_7	tRXCLK	RXCLK Period	Note 1	—	—	ns	VDDIO = 3.3V with CLOAD = 20 pF
ET_9	tRXCLKH	RXCLK High Time	tRXCLK / 2	—	—	ns	
ET_11	tRXCLKL	RXCLK Low Time	tRXCLK / 2	—	—	ns	
ET_13	tRXSU tRXERSU tRXDVSU	ERX[3:0], ERXER & ERXDV Set-up Time	4	—	—	ns	
ET_15	tRXHOLD tRXERHOLD tRXDVHOLD	ERX[3:0], ERXER & ERXDV Hold Time	0	—	—	ns	
MII TX AC TIMING Requirements							
ET_17	tTXCLK	TXCLK Period	Note 1	—	—	ns	VDDIO = 3.3V with CLOAD = 20 pF
ET_19	tTXCLKH	TXCLK High Time	tTXCLK / 2	—	—	ns	
ET_21	tTXCLKL	TXCLK Low Time	tTXCLK / 2	—	—	ns	
ET_23	tTXCOLSU tTXCRSSU	TXCOL & TXCRS Set-up Time	29	—	—	ns	
ET_25	tTXCOLHOLD tTXCRSHOLD	TXCOL & TXCRS Hold Time	9	—	—	ns	
ET_27	tTX[3:0]VAL tTXERVAL tTXENVAL	TX[3:0], TXER & TXEN valid times	9	—	11	ns	—

Note:

1. 40 ns, (i.e., 25 MHz), for 100BASE-TX operation, 400 ns, (i.e., 2.5 MHz), for 10BASE-T operation. Clock source ≤ 50 ppm. Clock accuracy and stability requirements will depend on performance targets and certification requirements.

Figure 52-35. RMII Ethernet Module AC Timing Diagrams (Reduced Media-Independent Interface)

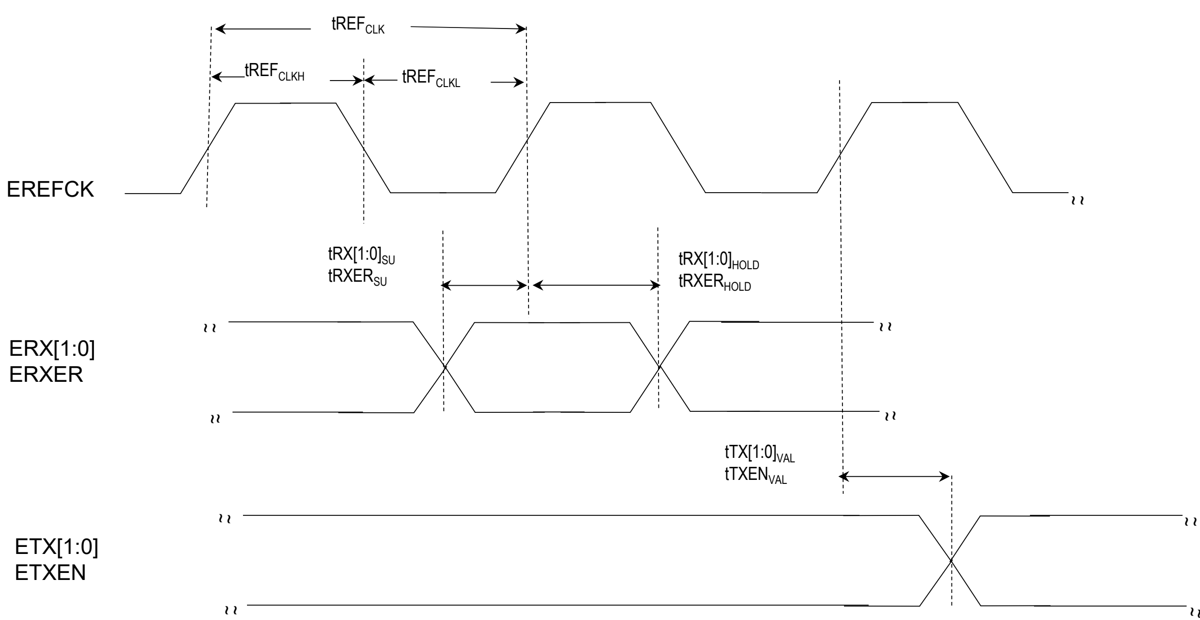


Table 52-46. RMII Ethernet Module AC Electrical Specifications (Reduced Media-Independent Interface)

AC CHARACTERISTICS			Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated) **Operating temperature:** **-40°C ≤ TA ≤ +85°C for Industrial**				
Param. No.	Symbol	Characteristics	Min.	Typ.	Max.	Units	Conditions
RMII AC TIMING Requirements							
ET_29	tREFCLK	Reference Clock Period	—	Note 1	—	ns	VDDIO = 3.3V with CLOAD = 20 pF
ET_31	tREFCLKIH	Reference Clock High Time	—	tREFCLK / 2	—	ns	
ET_33	tREFCLKIL	Reference Clock Low Time	—	tREFCLK / 2	—	ns	
ET_35	REFCLKDC	Reference Clock Duty Cycle	—	50	—	%	
ET_41	tRX[1:0]SU tRXERSU	RXD[1:0], RXER Set-up time	4	—	—	ns	
ET_43	tRX[1:0]HOLD tRXERHOLD	RXD[1:0], RXER hold time	0	—	—	ns	
ET_45	tTX[1:0]VAL tTXENVAL	TX[1:0], TXEN valid time	9	—	10	ns	

Note:
1. 40 ns, (i.e., 25 MHz), for 100BASE-TX operation, 400 ns, (i.e. 2.5 MHz), for 10BASE-T operation. Clock source ≤ 50 ppm. Clock accuracy and stability requirements will depend on performance targets and certification requirements.

52.33. Frequency Meter (FREQM) Electrical Specifications

Table 52-47. Frequency Meter AC Electrical Specifications

AC CHARACTERISTICS			Standard Operating Conditions: VDDREG=VDDIO=AVDD 1.71V to 3.63V (unless otherwise stated) **Operating temperature:** **-40°C ≤ TA ≤ +85°C for Industrial**				
Param. No.	Symbol	Characteristics	Min.	Typ.	Max.	Units	Conditions
FM_1	FMLOW	GCLK_IOx Input Low Time	FMPERIOD/2	—	—	ns	VDDIOx(min) and meet FM5 spec
FM_3	FMHIGH	GCLK_IOx Input High Time	FMPERIOD/2	—	—	ns	
FM_5	FMPERIOD	GCLK_IOx Input Period	1000/FCLK_45	—	—	ns	VDDIOx(min)
FM_7	fGCLK_FREQM_REF	FREQM Reference	—	—	FCLK_17	MHz	
FM_9	fGCLK_FREQM_MSR	FREQM Measure	—	—	FCLK_15	MHz	

52.34. True Random Number Generator (TRNG) Electrical Specifications

Table 52-48. True Random Number Generator Module Electrical Specifications

AC CHARACTERISTICS			Standard Operating Conditions: VDDREG=VDDIO=AVDD 1.71V to 3.63V (unless otherwise stated) **Operating temperature:** **-40°C ≤ TA ≤ +85°C for Industrial**				
Param. No.	Symbol	Characteristics	Min.	Typ.	Max.	Units	Conditions
TRNG_1	TRNGWKUP	Delay between TRNG Enable (CTRLA.ENABLE=1) and first random number read	100	—	—	ms	—

52.35. Position Decoder (PDEC) Electrical Specifications

Figure 52-36. PDEC Timing Diagrams Counter Mode AC Timing Diagrams

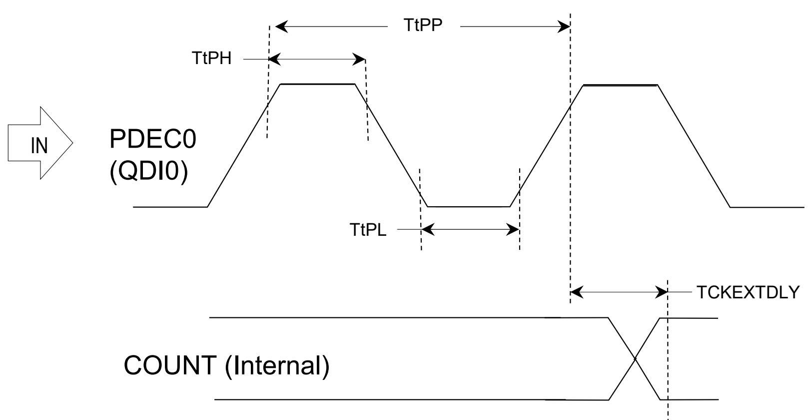


Figure 52-37. PDEC Input AC Timing Diagrams

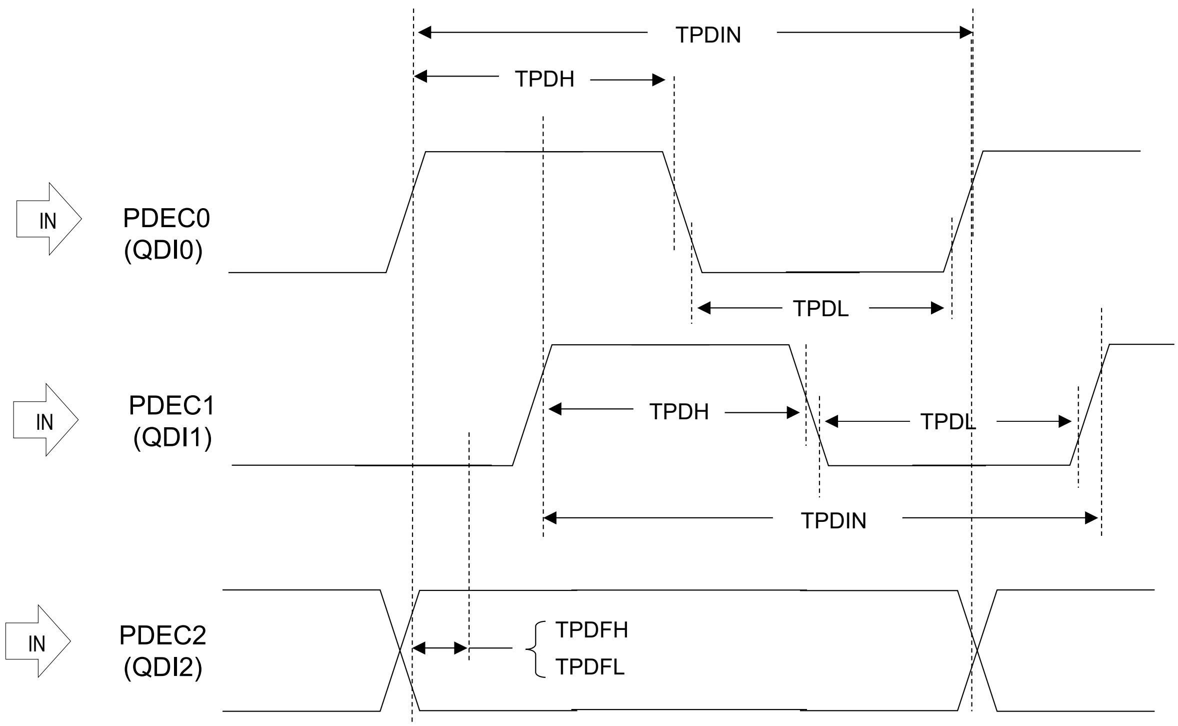


Table 52-49. Position Decoder Interface AC Electrical Specifications

AC CHARACTERISTICS			Standard Operating Conditions: VDDREG=VDDIO=AVDD 1.71V 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
PDEC_1	TtPH	TPCK high time	2/fGCLK_PDEC	—	—	μs	VDDIOx(min) -to-VDDIOx(max)
PDEC_3	TtPL	TPCK low time	2/fGCLK_PDEC	—	—	μs	
PDEC_5	TtPP	TPCK input period	4/fGCLK_PDEC	—	—	μs	
PDEC_7	TCKEXTDLY	Delay from External TxCK Clock Edge to counter Increment	—	—	4/fGCLK_PDEC	μs	
PDEC_11	TPDH	Position Decoder Input High Time	4/fGCLK_PDEC	—	—	μs	
PDEC_13	TPDL	Position Decoder Input Low Time	4/fGCLK_PDEC	—	—	μs	
PDEC_15	TPDIN	Position Decoder Input Period	8/fGCLK_PDEC	—	—	μs	
PDEC_21	TPDFH	Filter Time to Recognize High, with Digital Filter	4/fGCLK_PDEC	—	—	μs	
PDEC_23	TPDFL	Filter Time to Recognize Low, with Digital Filter	4/fGCLK_PDEC	—	—	μs	
PDEC_24	fGCLK_PDEC	GCLK_PDEC	—	—	FCLK_35A	MHz	

52.36. SD/MMC Host Controller (SDHC) Electrical Specifications

Figure 52-38. SD/SDIO/MMC SDHC Module AC Timing Diagram

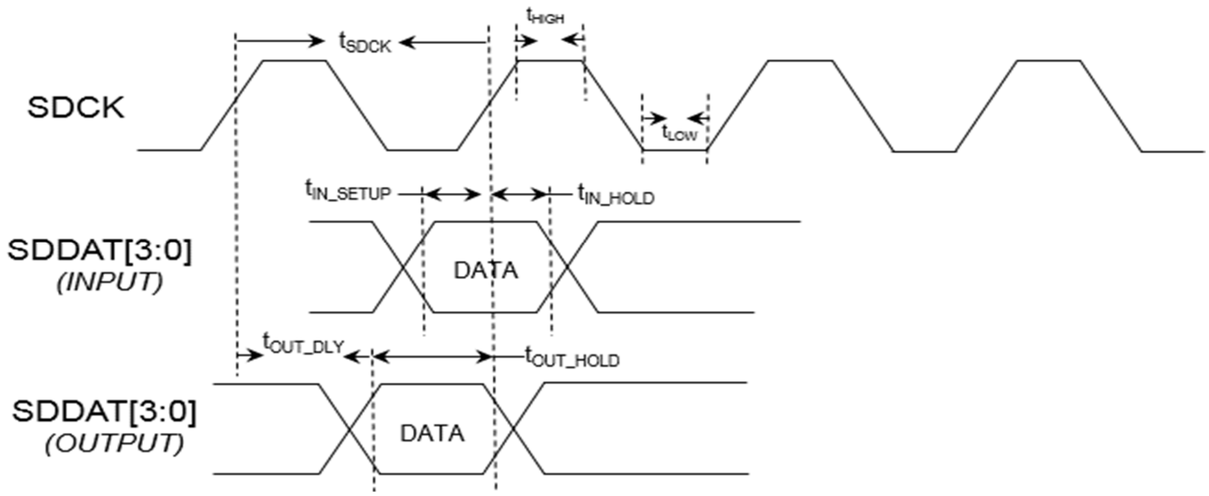


Table 52-50. SD Host Controller AC Timing Specifications [1]

AC CHARACTERISTICS			Standard Operating Conditions: VDDREG=VDDIO=AVDD 2.7V to 3.63V (unless otherwise stated) **Operating temperature:** **-40°C ≤ TA ≤ +85°C for Industrial**				
Param. No.	Symbol	Characteristics	Min.	Typ.	Max.	Units	Conditions
SD_1	fGCLK_SDHCx_SLOW	Common SDHC slow input clock frequency	—	—	FCLK_31	MHz	—
SD_3	fGCLK_SDHCx_CORE	SDHCx input clock frequency	—	—	FCLK_33	MHz	—
SD/SDIO Default Speed Mode							
SD_5	tSDCK	Clock Frequency	—	—	25	MHz	—
SD_7	tDUTY	Duty Cycle	—	50	—	%	—
SD_9	tHIGH	Clock High Time	500/tSDCK	—	—	ns	—
SD_11	tLOW	Clock Low Time	500/tSDCK	—	—	ns	—
SD_13	tRISE	Clock Rise Time	—	—	DI_29	ns	See parameter in I/O Specifications
SD_15	tFALL	Clock Fall Time	—	—	DI_31	ns	See parameter in I/O Specifications
SD_17	tIN_SETUP	Input Setup Time	3	—	—	ns	—
SD_19	tIN_HOLD	Input Hold Time	0	—	—	ns	—
SD_21	tOUT_DLY	Output Delay Time	—	—	7	ns	VDDIOx = 3.3V, CLOAD = 30 pF(MAX)
SD_23	tOUT_HOLD	Output HOLD Time	1	—	—	ns	
SD/SDIO High Speed Mode							
SD_25	tSDCK	Clock Frequency	—	—	52	MHz	—
SD_27	tDUTY	Duty Cycle	—	50	—	%	—
SD_29	tHIGH	Clock High Time	500/tSDCK	—	—	ns	—
SD_31	tLOW	Clock Low Time	500/tSDCK	—	—	ns	—
SD_29	tRISE	Clock Rise Time	—	—	DI_29	ns	See parameter in I/O Specifications

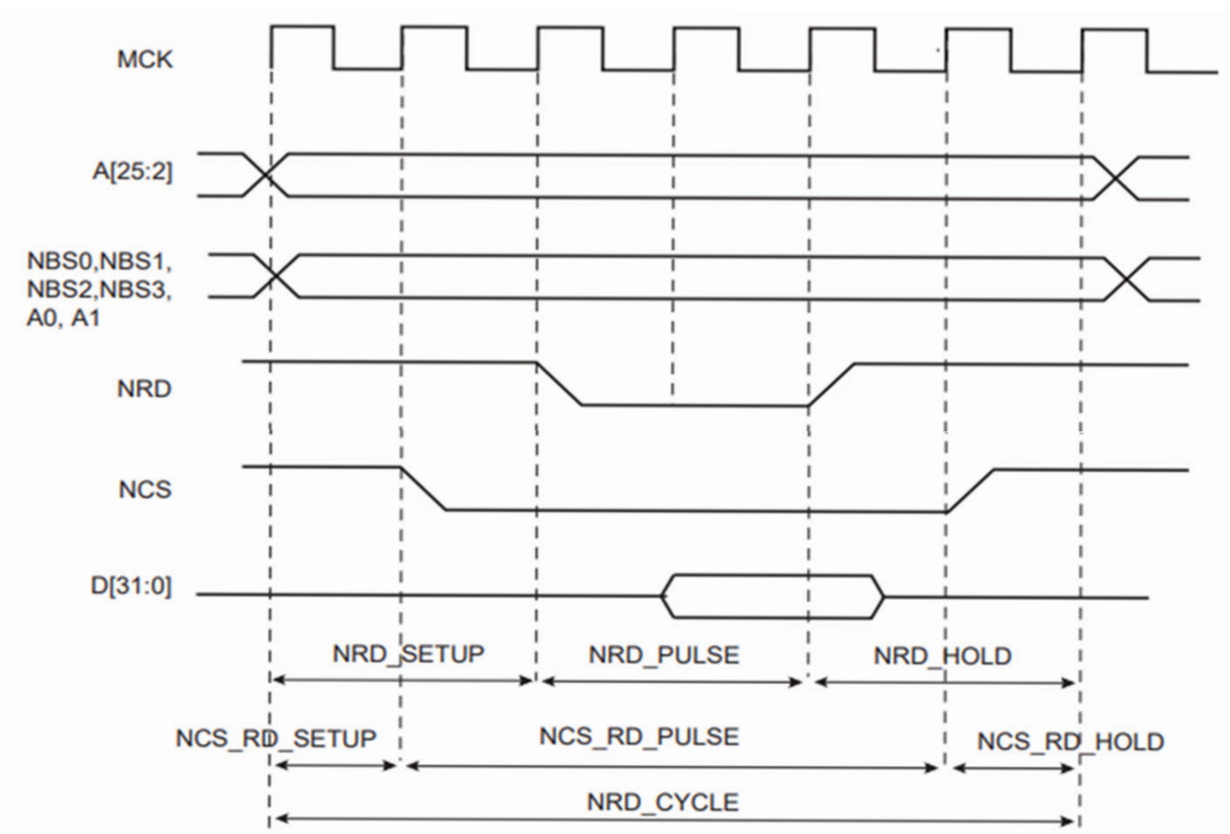
Table 52-50. SD Host Controller AC Timing Specifications [1] (continued)

AC CHARACTERISTICS			Standard Operating Conditions: VDDREG=VDDIO=AVDD 2.7V to 3.63V (unless otherwise stated) **Operating temperature:** **-40°C ≤ TA ≤ +85°C for Industrial**				
Param. No.	Symbol	Characteristics	Min.	Typ.	Max.	Units	Conditions
SD_31	tFALL	Clock Fall Time	—	—	DI_31	ns	See parameter in I/O Specifications
SD_33	tIN_SETUP	Input Setup Time	3	—	—	ns	—
SD_35	tIN_HOLD	Input Hold Time	0	—	—	ns	—
SD_37	tOUT_DLY	Output Delay Time	—	—	7	ns	VDDIOx = 3.3V, CLOAD = 30 pF(MAX)
SD_39	tOUT_HOLD	Output HOLD Time	0	—	—	ns	

Note:
1. All output pins with 30pF load.

52.37. External Bus Interface (EBI) Electrical Specifications

Figure 52-39. EBI Read Cycle AC Timing Diagram



READ_MODE = 1 (Write Operation Controlled by NRD):

- **NRD_SETUP:** NRD setup time is defined as the setup of address before the NRD falling edge
- **NRD_PULSE:** NRD pulse length is the time between NRD falling edge and NRD rising edge
- **NRD_HOLD:** NRD hold time is defined as the hold time of address after the NRD rising edge

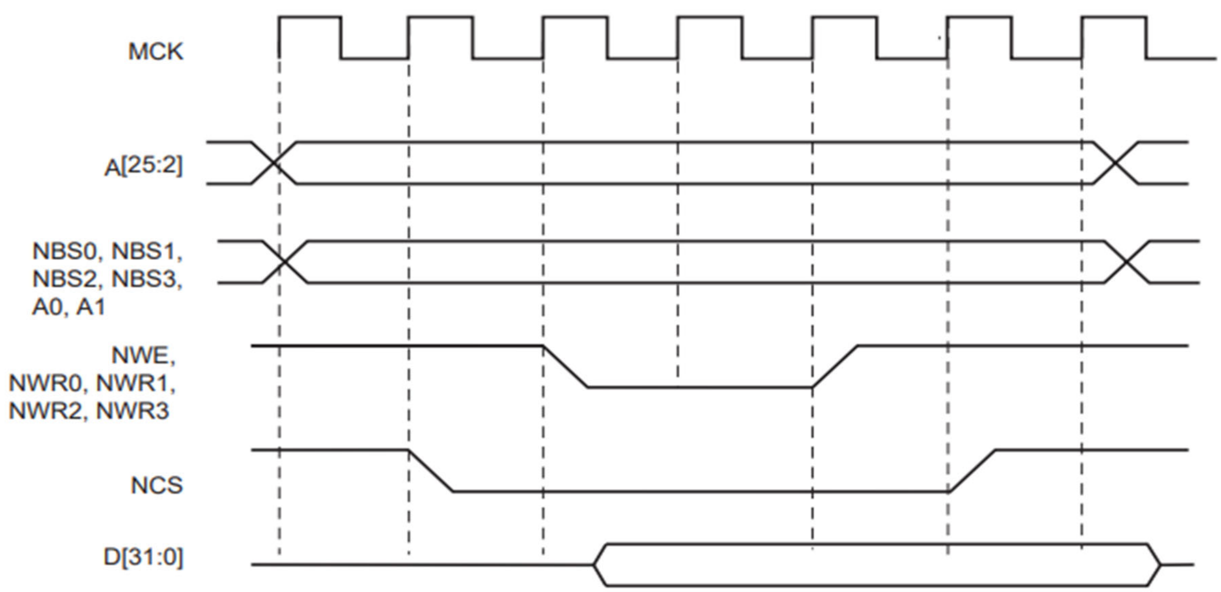
READ_MODE = 0 (Write Operation Controlled by NCS):

- **NCS_RD_SETUP:** NCS setup time is defined as the setup time of address before the NCS falling edge
- **NCS_RD_PULSE:** NCS pulse length is the time between NCS falling edge and NCS rising edge
- **NCS_RD_HOLD:** NCS hold time is defined as the hold time of address after the NCS rising edge

Read Cycle:

- NRD_CYCLE = NRD_SETUP + NRD_PULSE + NRD_HOLD
- NCS_CYCLE = NCS_RD_SETUP + NCS_RD_PULSE + NCS_RD_HOLD

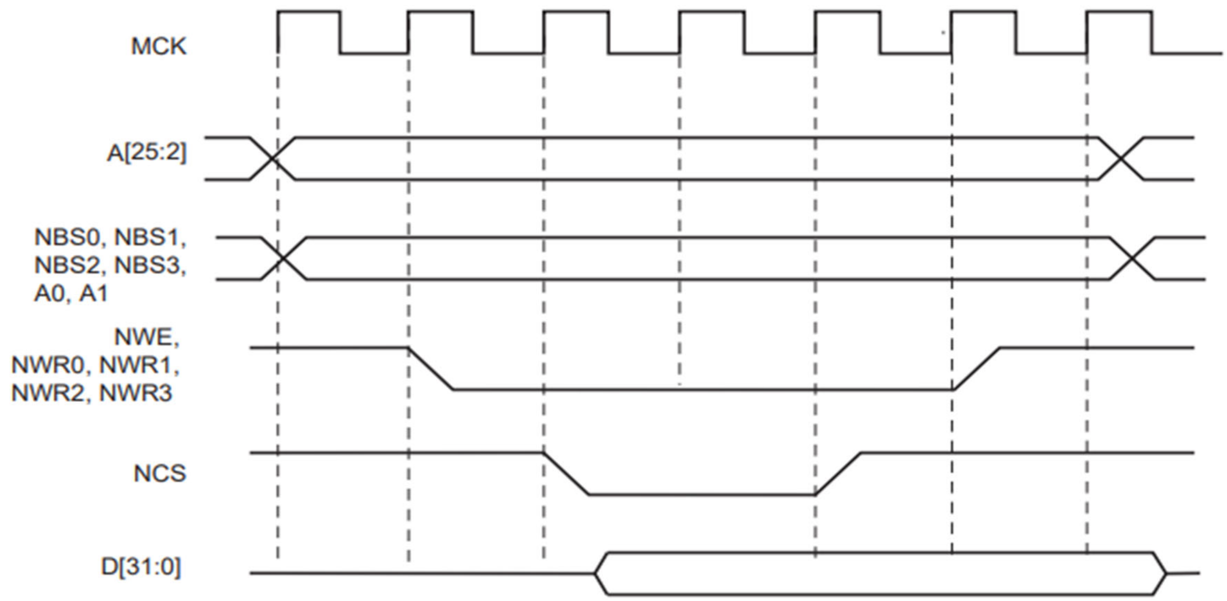
Figure 52-40. EBI Write Cycle AC Timing Diagram: WRITE_MODE = 1 (Write Operation Controlled by NWE)



Write Cycle:

- NWE_CYCLE = NWE_SETUP + NWE_PULSE + NWE_HOLD
- NWE_HOLD = NWE_CYCLE - NWE_SETUP - NWE_PULSE

Figure 52-41. EBI Write Cycle AC Timing Diagram: WRITE_MODE = 0 (Write Operation Controlled by NCS)



Write Cycle:

- NCS_CYCLE = NCS_WR_SETUP + NCS_WR_PULSE + NCS_WR_HOLD
- NCS_WR_HOLD = NWE_CYCLE - NCS_WR_SETUP - NCS_WR_PULSE

Table 52-51. EBI AC electrical Specifications

AC CHARACTERISTICS			Standard Operating Conditions: VDDIO=AVDD 2.4V to 3.63V, VDDREG=1.75V to 1.85V (unless otherwise stated) **Operating temperature:** **-40°C ≤ TA ≤ +85°C for Industrial**				
Param. No.	Symbol	Characteristics	Min.	Typ	Max.	Units	Conditions
EB3	NCS_CYCLE	EBI Read Cycle Time (TRC) NCS Controlled Read	33.519	—	—	ns	VDDIO = 3.3V, with 20 pF Load
EB5	NRD_CYCLE	EBI Read Cycle Time (TRC) NRD Controlled Read	26.848	—	—	ns	
EB7	NCS_CYCLE	EBI Write Cycle Time (TWC) NCS Controlled Write	20.04	—	—	ns	
EB9	NWE_CYCLE	EBI Write Cycle Time (TWC) NWE Controlled Write	26.707	—	—	ns	

52.38. JTAG Electrical Specifications

Figure 52-42. JTAG AC Timing Diagrams

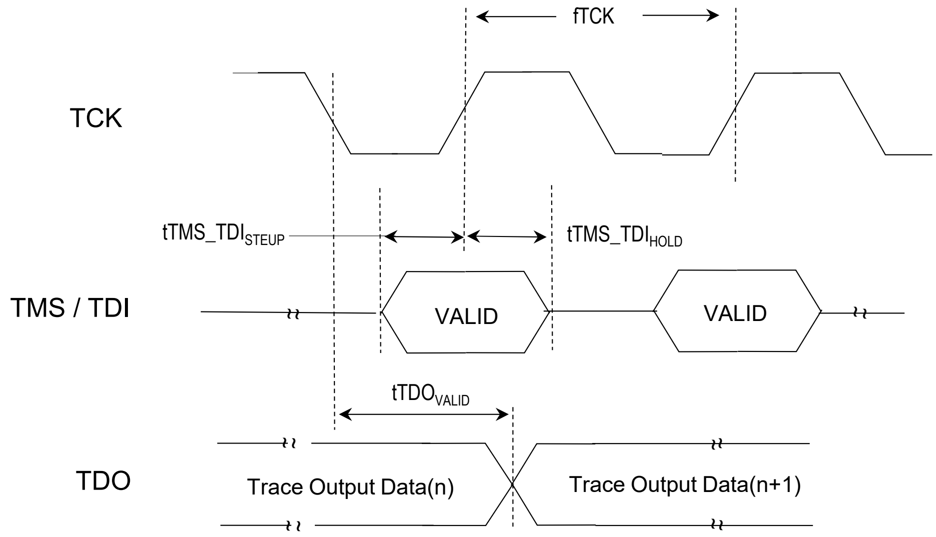


Table 52-52. JTAG AC Electrical Specifications

AC CHARACTERISTICS			Standard Operating Conditions: VDDREG=VDDIO=AVDD 1.71V 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
JTAG_1	fTCK	JTAG TCK Frequency	—	—	20	MHz	VDDIO(min) - VDDIO(max)
JTAG_3	tTMS_TDISTEUP	TMS/TDI Setup Time	5	—	—	ns	
JTAG_5	tTMS_TDIHOLD	TMS.TDI Hold Time	3	—	—	ns	
JTAG_7	tTDOVALID	TDO Output Valid	—	—	15	ns	

52.39. SWD 2-Wire Electrical Specifications

Figure 52-43. SWD 2-Wire Read/Write AC Timing Diagrams

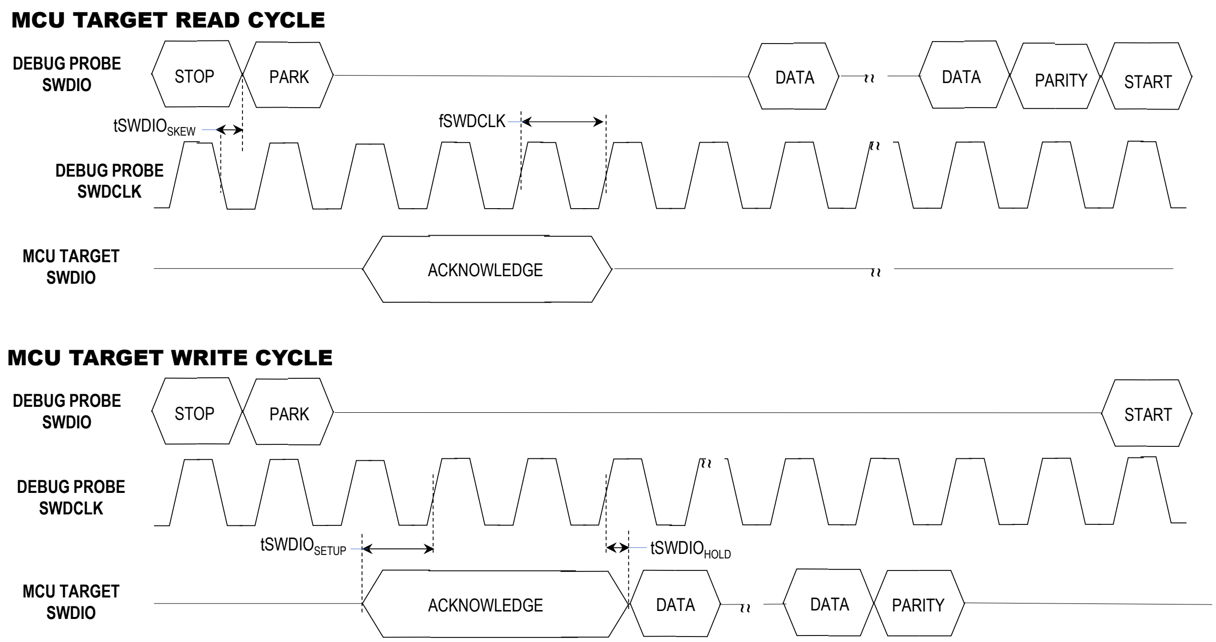


Table 52-53. SWD 2-Wire AC Electrical Specifications

AC CHARACTERISTICS			Standard Operating Conditions: VDDREG=VDDIO=AVDD 1.71V 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	—	—	30	MHz	VDDIO(min) - VDDIO(max)
SWD_3	tSWDCLKHIGH	SWDCLK Clock High Time	1 / (2 * fSWDCLK)	—	—	ns	
SWD_5	tSWDCLKLOW	SWDCLK Clock Low Time	1 / (2 * fSWDCLK)	—	—	ns	
SWD_7	tSWDIOSKEW	SWDIO Skew	-5	—	5	ns	
SWD_9	tSWDIOSETUP	SWDIO Setup Time	4	—	—	ns	
SWD_11	tSWDIOHOLD	SWDIO Hold Time	1	—	—	ns	

53. Extended Temperature Electrical Characteristics (125°C)

This section provides an overview of the PIC32CK SG/GC family electrical characteristics for devices running up to 125ºC. Additional information will be provided in future revisions of this document as it becomes available.

The specifications for Extended Temperature are identical to those shown in Electrical Characteristics, with the exception of the parameters listed in this chapter.

Absolute maximum ratings for the PIC32CK SG/GC family of devices 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.

53.1. 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

53.2. CPU Electrical Characteristics

Table 53-1. Operating Frequency Versus Voltage

Param. No.	VDDIOx, AVDD, Range	VDDREG Range	Temp. Range (in °C)	Max CPU Frequency	Comments
DC_5	1.7V to 3.63V	1.7-1.8V	-40°C to +125°C	100 MHz	Industrial

Table 53-2. CPU 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 —	—	125 150	°C
Power Dissipation: **Internal Chip Power Dissipation:** PINT = (VDDIOx x (IDD –∑ IOH_{VDDIOx})) + (VUSB3V3 x IDDUSB) + (AVDD x (IDDANA - ∑ IOH_{AVDD})) + (VDDREG x IDDREG) **I/O Pin Power Dissipation:** PI/O = ∑ ((VDDIOx – VOH_{VDDIOx}) x IOH_{VDDIOx}) + ∑ (VOL x IOL_{VDDIOx}) + ∑ ((AVDD – VOH_{AVDD}) x IOH_{AVDD}) + ∑ (VOL x IOL_{AVDD})	PD	$P_{INT} + P_{I/O}$			W
Maximum Allowed Power Dissipation	PDMAX	$(T_J - T_A)/\theta_{JA}$			W

Table 53-3. Thermal Packaging Characteristics (1)

Characteristics	Symbol	Typ.	Max.	Unit
Thermal Resistance, 144-pin TQFP (16x16x1 mm) Package	θ_{JA}	17.1	—	°C/W
Thermal Resistance, 100-pin eTQFP (14x14x1 mm) Package	θ_{JA}	18.5	—	°C/W
Thermal Resistance, 64-pin QFN (9x9x0.9 mm) Package	θ_{JA}	18.6	—	°C/W
Thermal Resistance, 64-pin TQFP (10x10x1 mm) Package	θ_{JA}	21.3	—	°C/W

Note:
1. Junction to ambient thermal resistance, Theta-JA (θ_{JA}) numbers are achieved by package simulations.

53.3. I/O Pin AC/DC Electrical Specifications

Table 53-4. I/O Pin AC/DC Electrical Specifications

AC - DC CHARACTERISTICS			Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.6V (unless otherwise stated) **Operating temperature:** **-40°C ≤ TA ≤ +125°C for Extended Temp**				
Param. No.	**Symbol**	**Characteristics**	**Min.**	**Typ**	**Max.**	**Units**	**Conditions**
DI_25	TRISE	I/O pin Rise Time (DRVSTR = 4x)	—	—	11.1	ns	VDDIOx(min), CLOAD = 30 pF(max)
DI_27	TFALL	I/O pin Fall Time (DRVSTR = 4x)	—	—	10.6	ns	
DI_29	TRISE	I/O pin Rise Time (DRVSTR = 8x, SLEW = FAST)	—	—	6.3	ns	
		I/O pin Rise Time (DRVSTR = 8x, SLEW = SLOW4)	—	—	21.1	ns	
		I/O pin Rise Time (DRVSTR = 8x, SLEW = SLOW8)	—	—	38.7	ns	
		I/O pin Rise Time (DRVSTR = 8x, SLEW = SLOW12)	—	—	54.6	ns	
DI_31	TFALL	I/O pin Fall Time (DRVSTR = 8x, SLEW = FAST)	—	—	6.1	ns	
		I/O pin Fall Time (DRVSTR = 8x, SLEW = SLOW4)	—	—	24	ns	
		I/O pin Fall Time (DRVSTR = 8x, SLEW = SLOW8)	—	—	41.7	ns	
		I/O pin Fall Time (DRVSTR = 8x, SLEW = SLOW12)	—	—	61.5	ns	

Notes:
1. VIL source < (GND - 0.3). Characterized but not tested.
2. VIH source > (VDDIO + 0.3) for non-5V tolerant pins only.
3. Digital 5V tolerant pins do not have an internal high-side diode to VDDIO, therefore cannot tolerate any positive input injection current.
4. 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 > (VDDIO + 0.3) or VIL source < (GND - 0.3)).
5. Any number and the combination of I/O pins not excluded under IICL or IICH conditions are permitted provided the absolute instantaneous sum of the input injection currents from all pins do not exceed the specified ∑IICT limit. To limit the injection current, the user must insert a resistor in series RSERIES (i.e., RS) between the input source voltage and device pin. The resistor value is calculated according to the following:
   – For negative Input voltages less than (GND -0.3): RS ≥ absolute value of | ((VIL source - (GND -0.3)) / IICL) |
   – For positive input voltages greater than (VDDIO +0.3): RS ≥ ((VIH source - (VDDIO +0.3))/IICH)
   – For Vpin voltages > VDDIO +0.3 and < GND -0.3, then RS = The larger of the values calculated above

53.4. Maximum Clock Frequencies Electrical Specifications

Table 53-5. Maximum Clock Frequencies AC Electrical Specifications

AC CHARACTERISTICS		Standard Operating Conditions: VDDREG=VDDIO=AVDD 1.71V to 3.6V (unless otherwise stated) **Operating temperature:** **-40°C ≤ TA ≤ +125°C for Extended Temp**		
Param. No.	**Symbol**	**Characteristics**	**Max**	**Units**
FCLK_1	FCY	MCU clock freq	100	MHz
FCLK_3	fAHB	AHB clock freq	100	MHz
FCLK_5	fAPBn	APBA, APBB, APBC clock freq	100	MHz
FCLK_6	fGCLKGEN[0]	GCLK clock freq output	100	MHz
	fGCLKGEN[1:7]		160	MHz
	fGCLKGEN[8:11]		80	MHz
FCLK_7	fGCLK_PLL	PLL reference clock freq	48	MHz
FCLK_11	fGCLK_DFLL48M_REF	DFLL 48M reference clock freq	1	MHz
FCLK_13	fGCLK_EIC	EIC input clock freq	80	MHz
FCLK_15	fGCLK_FREQM_MSR	FREQM Measure Clock Frequency	160	MHz
FCLK_17	fGCLK_FREQM_REF	FREQM Reference Clock Frequency	80	MHz
FCLK_19	fGCLK_EVSYS_CHANNELx	EVSYS channel x input clock freq	80	MHz
FCLK_21	fGCLK_SERCOMx_SLOW	Common SERCOM slow input clock freq	32	kHz
FCLK_23	fGCLK_SERCOMx_CORE, x = 1 to 7	SERCOMx input clock freq	80	MHz
	fGCLK_SERCOM0_CORE	SERCOM0 input clock freq	160	MHz
FCLK_25	fGCLK_CANx	CAN input clock freq	80	MHz
FCLK_27	fGCLK_USBFS	USB FS input clock freq	48	MHz
FCLK_29	fGCLK_IxS	IxS input clock freq	80	MHz
FCLK_31	fGCLK_SDHCx_SLOW	Common SDHC slow input clock freq	12	MHz
FCLK_33	fGCLK_SDHCx_CORE	SDHCx input clock freq	106	MHz
FCLK_35	fGCLK_TCCx	TCCx input clock freq	160	MHz
FCLK_36	fGCLK_PDEC	PDEC input clock freq	160	MHz
FCLK_43	fGCLK_CCL	CCL input clock freq	80	MHz
FCLK_45	fGCLK_GCLKINx	External GCLKx input clock freq	50	MHz
FCLK_47	fGCLK_CM33_TRACE	CM33 Trace input clock freq	50	MHz
FCLK_49	fGCLK_AC	Analog comparator peripheral module clock freq	80	MHz
FCLK_51	fGCLK_ADCx	ADCx input clock freq	80	MHz
FCLK_55	fGCLK_PTC	PTC input clock freq	80	MHz
FCLK_61	fGCLK_ETH	Ethernet input clock freq	(*)	MHz
FCLK_62	fGCLK_ETH_TSU	Ethernet TSU input clock freq	(*)	MHz
FCLK_73	fGCLK_QSPI	QSPI internal GCLK freq	120	MHz

53.5. ADC Electrical Specifications

Table 53-6. ADC Conversion and Sample AC Electrical Specifications

AC CHARACTERISTICS			Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.6V (unless otherwise stated) **Operating Temperature:** **-40°C ≤ TA ≤ +125°C for Extended Temp**				
Param. No.	Symbol	Characteristics	Min.	Typ.	Max.	Units	Conditions
ADC Throughput Rates							
ADC_57	FTPR [1]	Sample-Rate for ADC with SAMC = 4 (min)	—	—	2.5	Msps	12-bit resolution, Rsource Impedance ≤ 200 Ω
			—	—	2.8571		10-bit resolution, Rsource Impedance ≤ 250 Ω
			—	—	3.333333		8-bit resolution, Rsource Impedance ≤ 300 Ω
			—	—	4		6-bit resolution, Rsource Impedance ≤ 400 Ω

Notes:
1. ADC Throughput Rate FTP = ((1 / ((TSAMP + TCNV) * TAD)) / (Number of user active analog inputs in use on the specific target ADC module)).
2. Specification values assume only one AINx channel in use.

53.6. Analog Comparator (AC) Electrical Specifications

Table 53-7. Comparator AC Electrical Specifications

AC CHARACTERISTICS			Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.6V (unless otherwise stated) **Operating Temperature:** **-40°C ≤ TA ≤ +125°C for Automotive**				
Param. No.	Symbol	Characteristics	Min.	Typ.[1]	Max.	Units	Conditions
CMP_1	VIOFF	Input Offset Voltage	-10	—	10	mV	COMPCTRLn.HYST = 0x3. Comparator reference voltage = AVDD/2
CMP_5	VHYST_00	Input Hysteresis Voltage	2	—	18	mV	COMPCTRLn.HYST = 0x0. Comparator reference voltage = AVDD/2
	VHYST_01	Input Hysteresis Voltage	5	—	28	mV	COMPCTRLn.HYST = 0x1. Comparator reference voltage = AVDD/2
	VHYST_02	Input Hysteresis Voltage	18	—	61	mV	COMPCTRLn.HYST = 0x2. Comparator reference voltage = AVDD/2
	VHYST_03	Input Hysteresis Voltage	30	—	93	mV	COMPCTRLn.HYST = 0x3. Comparator reference voltage = AVDD/2

Table 53-7. Comparator AC Electrical Specifications (continued)

AC CHARACTERISTICS			Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.6V (unless otherwise stated) **Operating Temperature:** **-40°C ≤ TA ≤ +125°C for Automotive**				
Param. No.	Symbol	Characteristics	Min.	Typ.[1]	Max.	Units	Conditions
CMP_15	TRESPSS ain<0:1>	Small Signal Response Time - High speed	—	(*)	64	ns	COMPCTRLn.SPEED = 0x0 COMPCTRLn.MUXPOS = 0x0 COMPCTRLn.MUXNEG = 0x1 COMPCTRLn.HYST = 0x3 Comparator reference voltage = AVDD/2. Input overdrive = +/- 180 mV
		Small Signal Response Time - Low Speed	—	(*)	260	ns	COMPCTRLn.SPEED = 0x1 COMPCTRLn.MUXPOS = 0x0 COMPCTRLn.MUXNEG = 0x1 COMPCTRLn.HYST = 0x3 Comparator reference voltage = AVDD/2. Input overdrive = +/- 180 mV
	TRESPSS ain<2:3>	Small Signal Response Time - High speed	—	(*)	102	ns	COMPCTRLn.SPEED = 0x0 COMPCTRLn.MUXPOS = 0x2 COMPCTRLn.MUXNEG = 0x3 COMPCTRLn.HYST = 0x3 Comparator reference voltage = AVDD/2. Input overdrive = +/- 180 mV
		Small Signal Response Time - Low Speed	—	(*)	260	ns	COMPCTRLn.SPEED = 0x1 COMPCTRLn.MUXPOS = 0x2 COMPCTRLn.MUXNEG = 0x3 COMPCTRLn.HYST = 0x3 Comparator reference voltage = AVDD/2. Input overdrive = +/- 180 mV

Note:

1. Typical values at 25°C 2: Comparator Reference voltage cannot exceed: VIN(max) - VIOFF(max) - CMP_5(max) - 100 mV) ≥CMP_23 ≥ (VIN(min) + abs(VIOFF(min)) + CMP_5(max) + 100 mV).

Table 53-8. DAC Module AC Electrical Specifications

AC CHARACTERISTICS			Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.6V (unless otherwise stated) **Operating Temperature :** **-40°C ≤ TA ≤ +125°C for Extended Temp**				
Param. No.	Symbol	Characteristics	Min.	Typ.[1]	Max.	Units	Conditions
DAC_1	DRES	DAC Resolution	—	—	7	Bits	—
DAC_7	VOUT	Output Voltage Range	AVSS	—	AVDD - 1LSB	V	—
DAC_17	Tset_FS	DAC Full Scale Settling Time	—	—	1.8	μs	+/-4 LSB of final value for step size from 10% to 90%

Table 53-8. DAC Module AC Electrical Specifications (continued)

AC CHARACTERISTICS			Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.6V (unless otherwise stated) **Operating Temperature :** **-40°C ≤ TA ≤ +125°C for Extended Temp**				
Param. No.	Symbol	Characteristics	Min.	Typ.(1)	Max.	Units	Conditions
DAC_18a	Thold	DAC Holding Time	0.25	—	—	ms	This parameter applies only for Low- Power mode: DACCTRLn.SHENm = 0x1 Time for 1 LSB voltage loss of the DAC output
DAC_18b			0.31	—	—	ms	This parameter applies only for Low-Power mode: DACCTRLn.SHENm = 0x1 Time for 10 LSB voltage loss of the DAC output
SDAC_19	INL	Integral Non Linearity	-0.5	—	0.5	LSB	—
SDAC_21	DNL	Differential Non Linearity	0.05	—	0.4	LSB	—
SDAC_23	GERR	Gain Error	-2	—	-0.2	LSB	—
SDAC_25	EOFF	Offset Error	-1.4	—	1.4	LSB	Normal mode: DACCTRLn.SHENm = 0x0
			-6.4	—	6.4	LSB	Low-Power mode: DACCTRLn.SHENm = 0x1

53.7. NVM Block (Flash, Data Flash, NVM Configuration Rows) Electrical Specifications

Table 53-9. Flash NVM AC Electrical Specifications

AC CHARACTERISTICS				Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.6V (unless otherwise stated) **Operating Temperature:** **-40°C ≤ TA ≤ +125°C for Extended Temp**				
Param. No.	Symbol	Characteristics		Min.	Typ.	Max.	Units	Conditions
NVM_5	FREAD(1)	Flash Read(1)	0 Wait States	—	—	(*)	MHz	VDDIOx = 3.3V, FCR.CTRLA.AUTOWS = 0
			1 Wait States	—	—	(*)		
			2 Wait States	—	—	(*)		
			3 Wait States	—	—	(*)		
			4 Wait States	—	—	(*)		
			5 Wait States	—	—	(*)		
			0 Wait States	—	—	(*)	MHz	VDDIOx = 1.8V or VDDIO(min) whichever is greater, FCR.CTRLA.AUTOWS = 0
			1 Wait States	—	—	(*)		
			2 Wait States	—	—	(*)		
			3 Wait States	—	—	(*)		
			4 Wait States	—	—	(*)		
			5 Wait States	—	—	(*)		

Table 53-9. Flash NVM AC Electrical Specifications (continued)

AC CHARACTERISTICS			Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.6V (unless otherwise stated) **Operating Temperature:** **-40°C ≤ TA ≤ +125°C for Extended Temp**				
Param. No.	Symbol	Characteristics	Min.	Typ.	Max.	Units	Conditions

Notes:

1. 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. Theses tables provide the device maximum operating frequency defined by the FWS field of the FCW CTRLA register when automatic wait states (AUTOWS) is disabled. This field defines the number of Wait states required to access the Embedded Flash Memory.
2. Cell Endurance is reached with Flash Pre-Programming option disabled. Enabling such option will improve the cell endurance but slow-down the programming time.

54. Packaging Information

54.1. Package Marking Information

All devices are marked with the Microchip logo, a shortened ordering code, the ARM logo, and additional information that will be marked as follows:

YYWWNNN

Where:

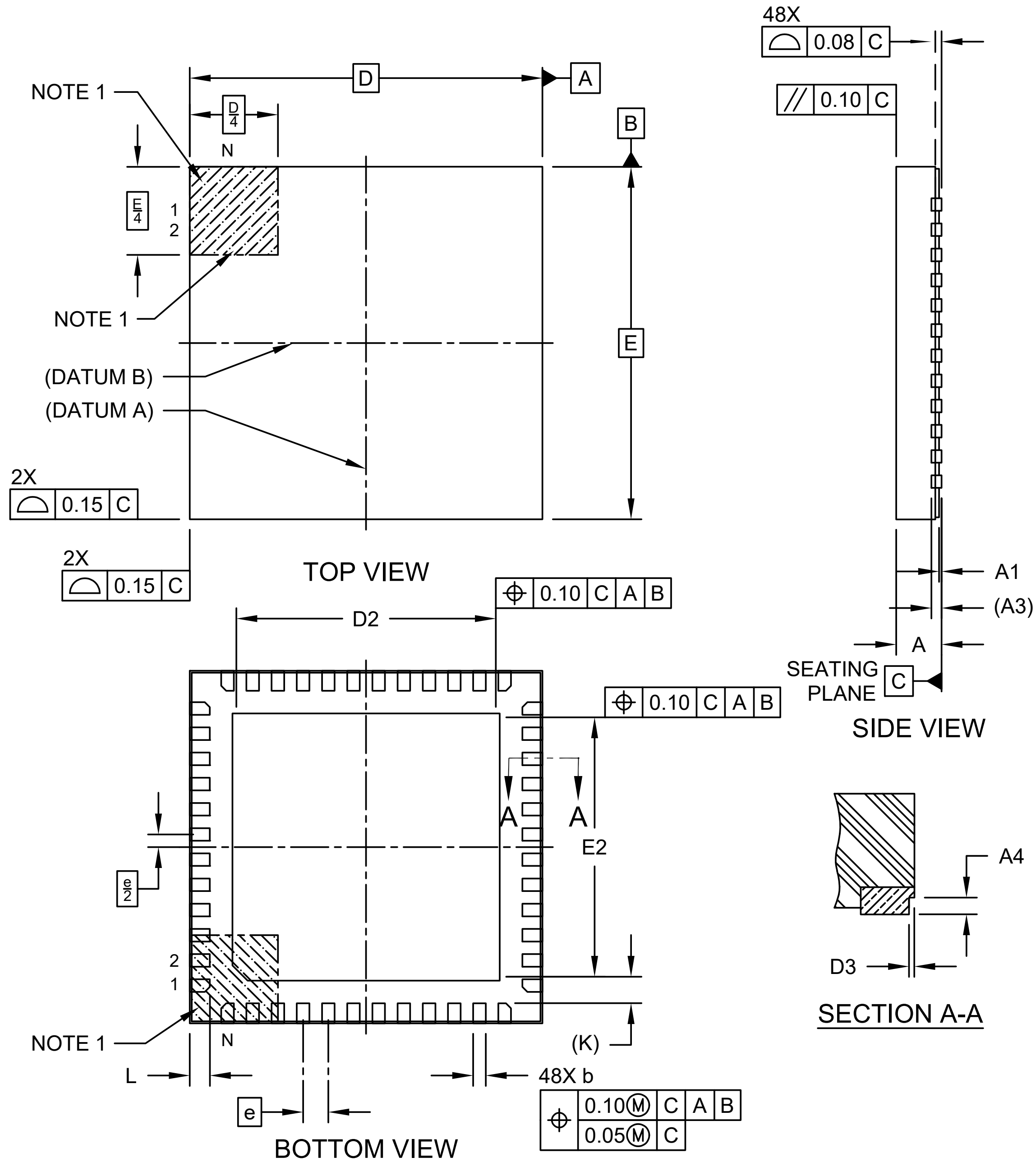
- "YY": Manufacturing Year
- "WW": Manufacturing Week
- "NNN": Internal Code

54.2. Package Drawings

54.2.1. 48-Lead VQFN Package

48-Lead Very Thin Plastic Quad Flat, No Lead Package (4YW) - 7x7x0.9mm Body [VQFN] With 5.3 mm Exposed Pad and Stepped Wettable Flanks

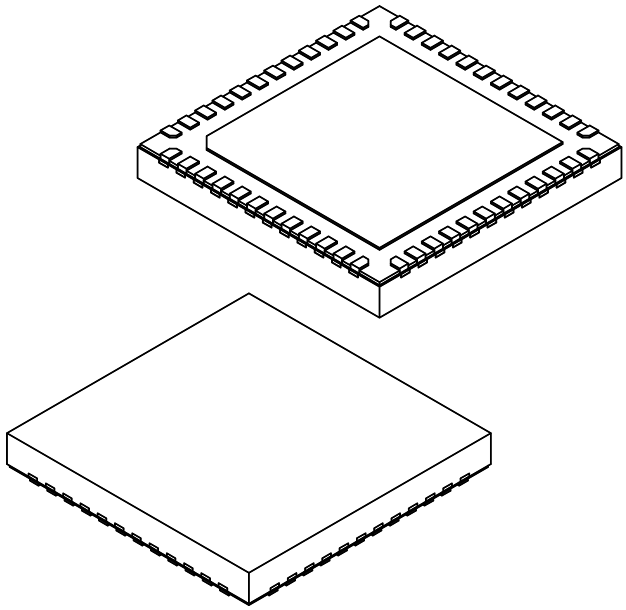
Note:	For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



Microchip Technology Drawing C04-601 Rev A Sheet 1 of 2

48-Lead Very Thin Plastic Quad Flat, No Lead Package (4YW) - 7x7x0.9mm Body [VQFN] With 5.3 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		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	5.20	5.30	5.40
Overall Width	E		7.00 BSC	
Exposed Pad Width	E2	5.20	5.30	5.40
Terminal Width	b	0.18	0.25	0.30
Terminal Length	L	0.30	0.40	0.50
Terminal-to-Exposed-Pad	K		0.45 REF	
Wettable Flank Step Length	D3	0.01	-	0.06
Wettable Flank Step Height	A4	0.10	-	–

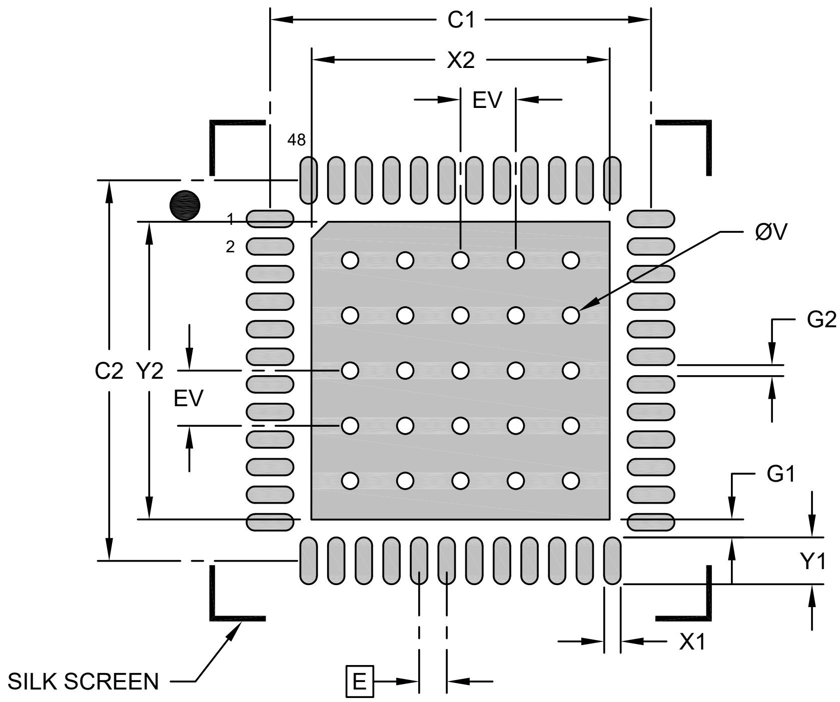
Notes:

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 C04-601 Rev A Sheet 2 of 2

48-Lead Very Thin Plastic Quad Flat, No Lead Package (4YW) - 7x7x0.9mm Body [VQFN] With 5.3 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			5.30
Optional Center Pad Length	Y2			5.30
Contact Pad Spacing	C1		7.00	
Contact Pad Spacing	C2		7.00	
Contact Pad Width (X48)	X1			0.30
Contact Pad Length (X48)	Y1			0.80
Contact Pad to Center Pad (X48)	G1	0.30		
Contact Pad to Contact Pad (X44)	G2	0.20		
Thermal Via Diameter	V		0.30	
Thermal Via Pitch	EV		1.00	

Notes:

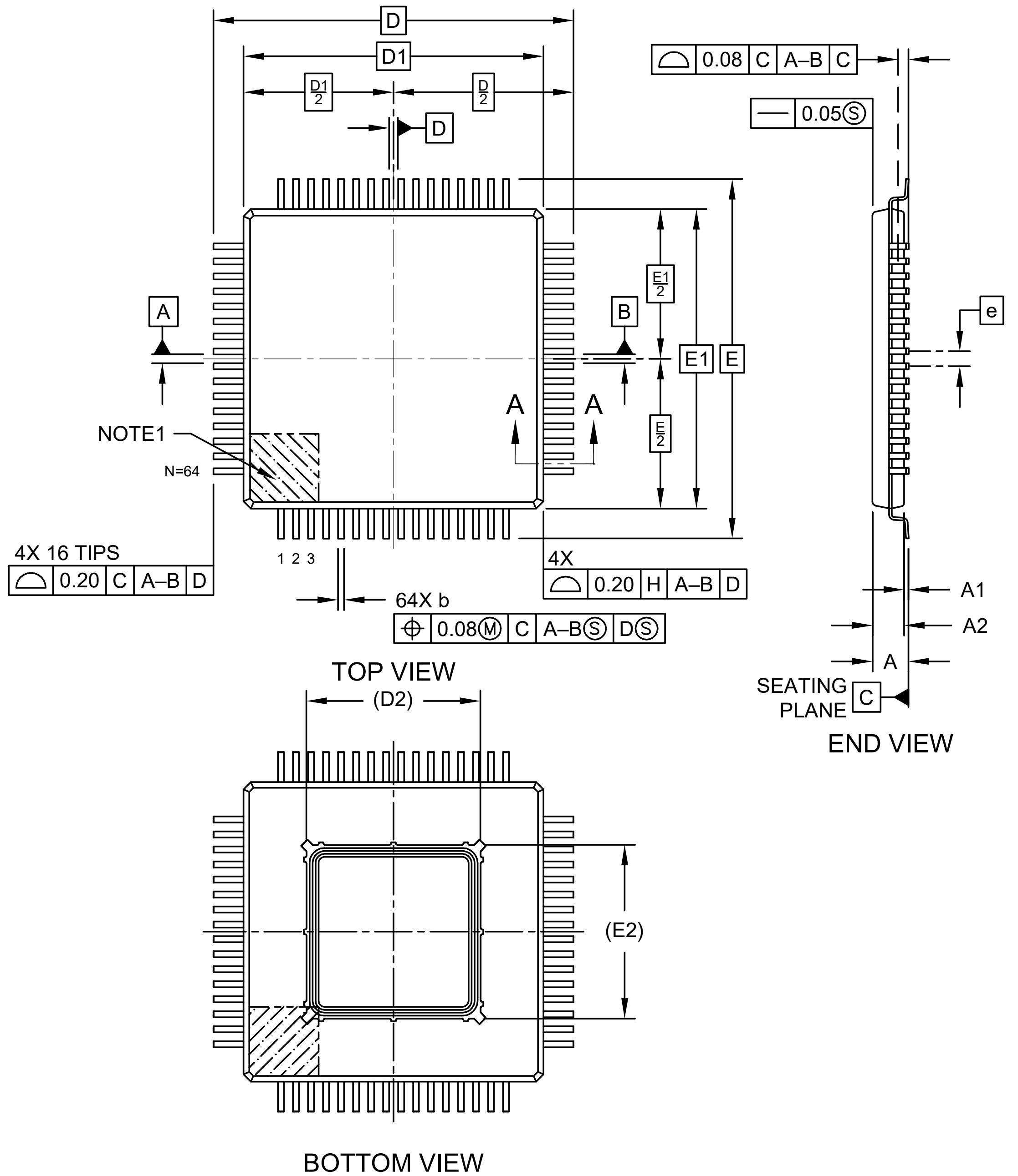
1. Dimensioning and tolerancing per ASME Y14.5M
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.
2. For best soldering results, thermal vias, if used, should be filled or tented to avoid solder loss during reflow process

Microchip Technology Drawing C04-2601 Rev A

54.2.2. 64-Lead TQFP Package

64-Lead Plastic Thin Quad Flat Pack (3ZB) - 10x10x1.0 mm Body [TQFP]
With 5.8 mm Grooved Exposed Pad

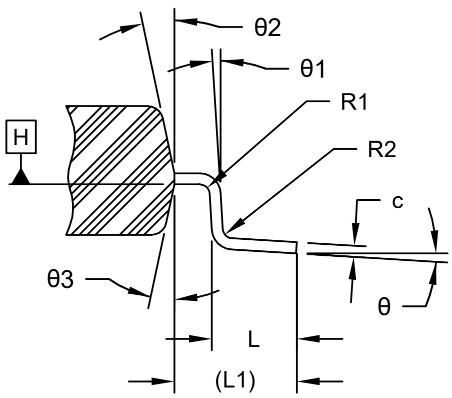
Note:	For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



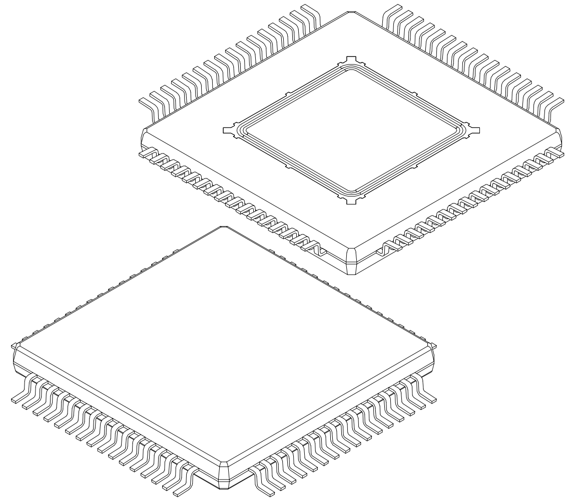
Microchip Technology Drawing C04-21544 Rev A Sheet 1 of 2

64-Lead Plastic Thin Quad Flat Pack (L8C) - 10x10x1.0 mm Body [TQFP] With 5.8 mm Grooved Exposed Pad

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		64	
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		12.00 BSC	
Molded Package Length	D1		10.00 BSC	
Exposed Pad Length	D2		5.80 REF	
Overall Width	E		12.00 BSC	
Molded Package Width	E1		10.00 BSC	
Exposed Pad Width	E2		5.80 REF	
Terminal Width	b	0.17	0.20	0.27
Terminal Thickness	c	0.09	0.127	0.20
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°
Mold Draft Angle	θ3	11°	12°	13°

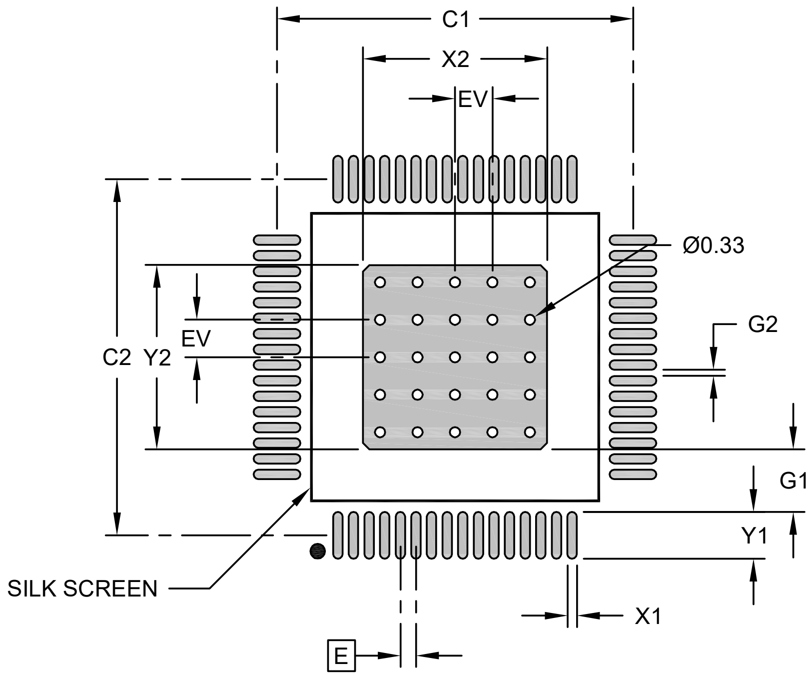
Notes:

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 C04-21544 Rev A Sheet 2 of 2

64-Lead Plastic Thin Quad Flat Pack (3ZB) - 10x10x1.0 mm Body [TQFP] With 5.8 mm Grooved Exposed Pad

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	
Center Pad Width	X2			5.90
Center Pad Length	Y2			5.90
Contact Pad Spacing	C1		11.40	
Contact Pad Spacing	C2		11.40	
Contact Pad Width (X64)	X1			0.30
Contact Pad Length (X64)	Y1			1.50
Contact Pad to Center Pad (X64)	G1	2.00		
Contact Pad to Contact Pad (X60)	G2	0.20		
Thermal Via Diameter	V		0.33	
Thermal Via Pitch	EV		1.20	

Notes:

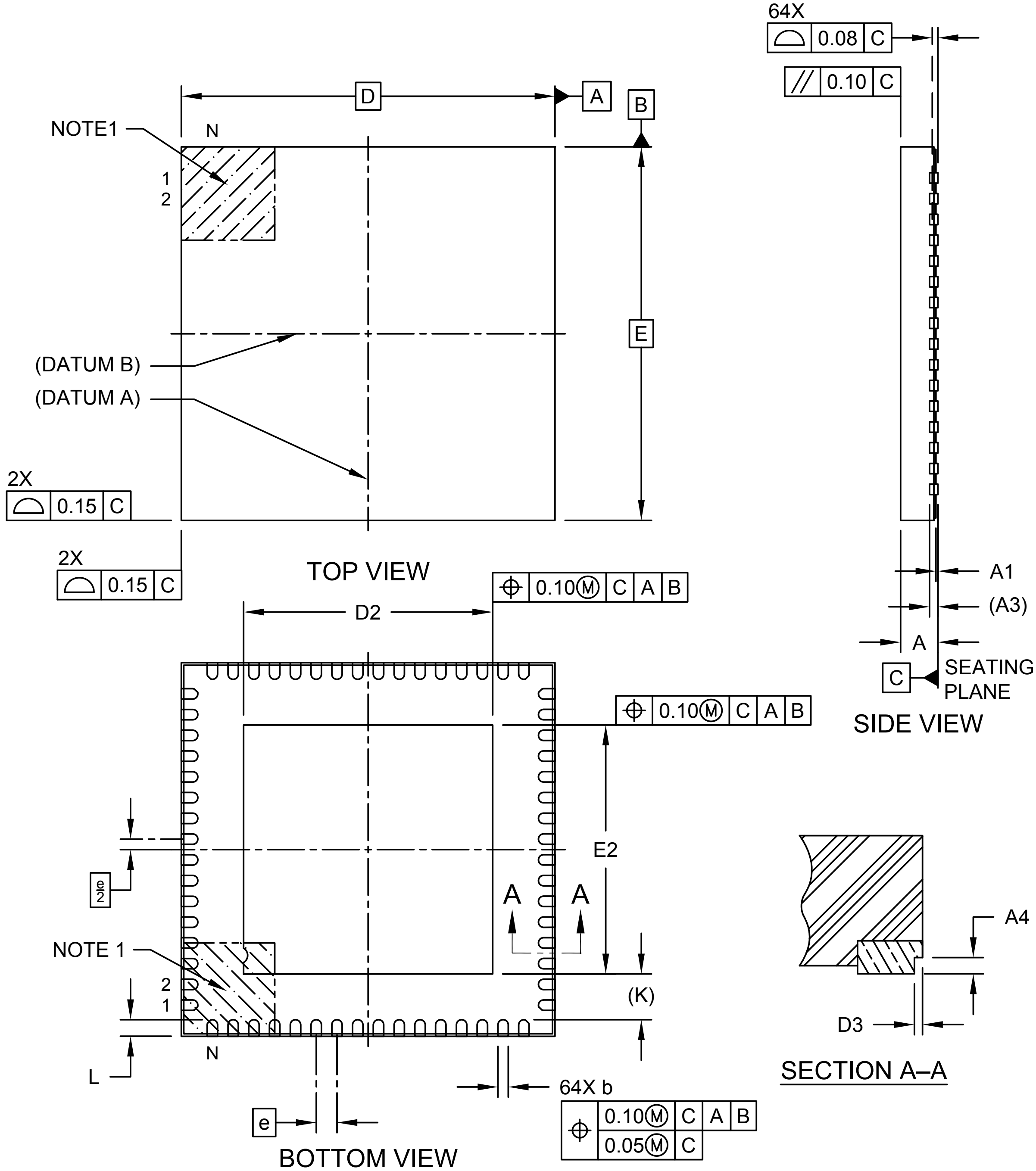
1. Dimensioning and tolerancing per ASME Y14.5M
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.
2. For best soldering results, thermal vias, if used, should be filled or tented to avoid solder loss during reflow process

Microchip Technology Drawing C04-23544 Rev A

54.2.3. 64-Lead VQFN

**64-Lead Very Thin Quad Flat, No Lead Package (LZX) - 9x9x0.9 mm Body [VQFN]
With 6.00 mm Exposed Pad and Stepped Wettable Flanks**

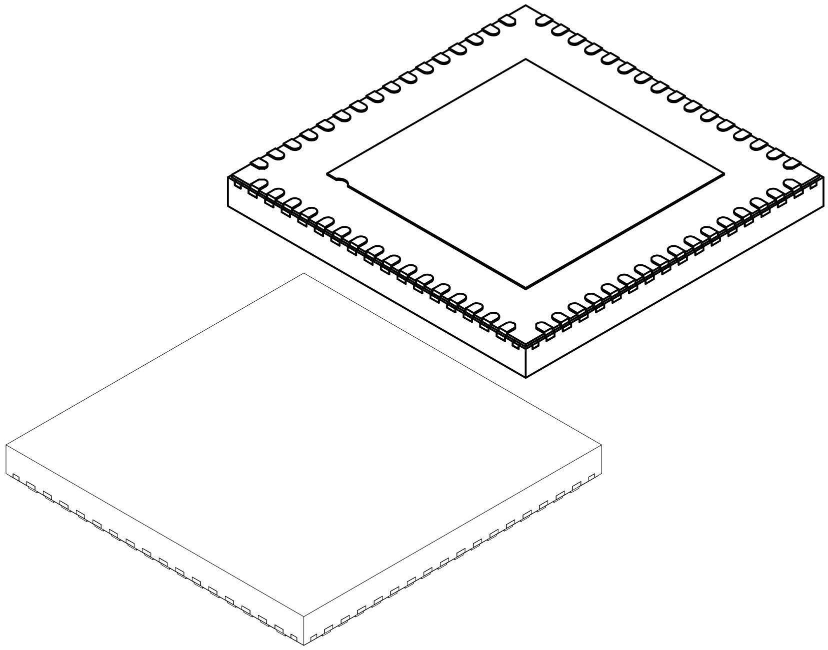
Note:	For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



Microchip Technology Drawing C04-514 Rev A Sheet 1 of 2

64-Lead Very Thin Quad Flat, No Lead Package (LZX) - 9x9x0.9 mm Body [VQFN] With 6.00 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	–	–	0.90
Standoff	A1	0.00	0.02	0.05
Terminal Thickness	A3		0.203 REF	
Overall Length	D		9.00 BSC	
Exposed Pad Length	D2	5.90	6.00	6.10
Overall Width	E		9.00 BSC	
Exposed Pad Width	E2	5.90	6.00	6.10
Terminal Width	b	0.18	0.25	0.30
Terminal Length	L	0.30	0.40	0.50
Terminal-to-Exposed-Pad	K		1.10 REF	
Wettable Flank Step Length	D3	-	-	0.085
Wettable Flank Step Height	A4	0.10	-	0.19

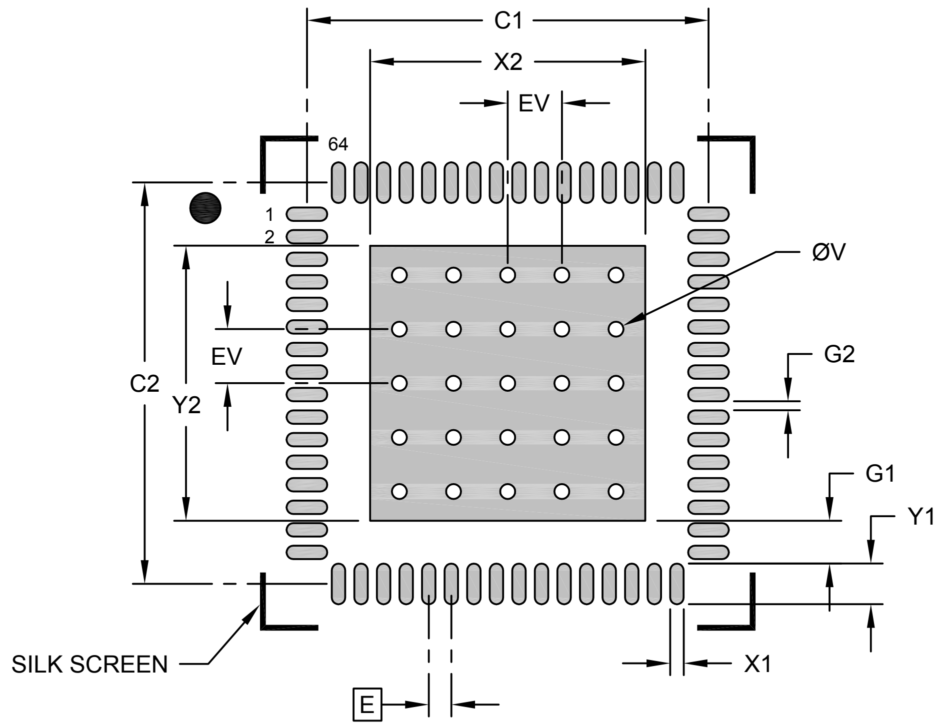
Notes:

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 C04-514 Rev A Sheet 2 of 2

64-Lead Very Thin Quad Flat, No Lead Package (LZX) - 9x9x0.9 mm Body [VQFN]
With 6.00 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			6.10
Optional Center Pad Length	Y2			6.10
Contact Pad Spacing	C1		8.90	
Contact Pad Spacing	C2		8.90	
Contact Pad Width (Xnn)	X1			0.30
Contact Pad Length (Xnn)	Y1			0.90
Contact Pad to Center Pad (Xnn)	G1	0.95		
Contact Pad to Contact Pad (Xnn)	G2	0.20		
Thermal Via Diameter	V		0.33	
Thermal Via Pitch	EV		1.20	

Notes:

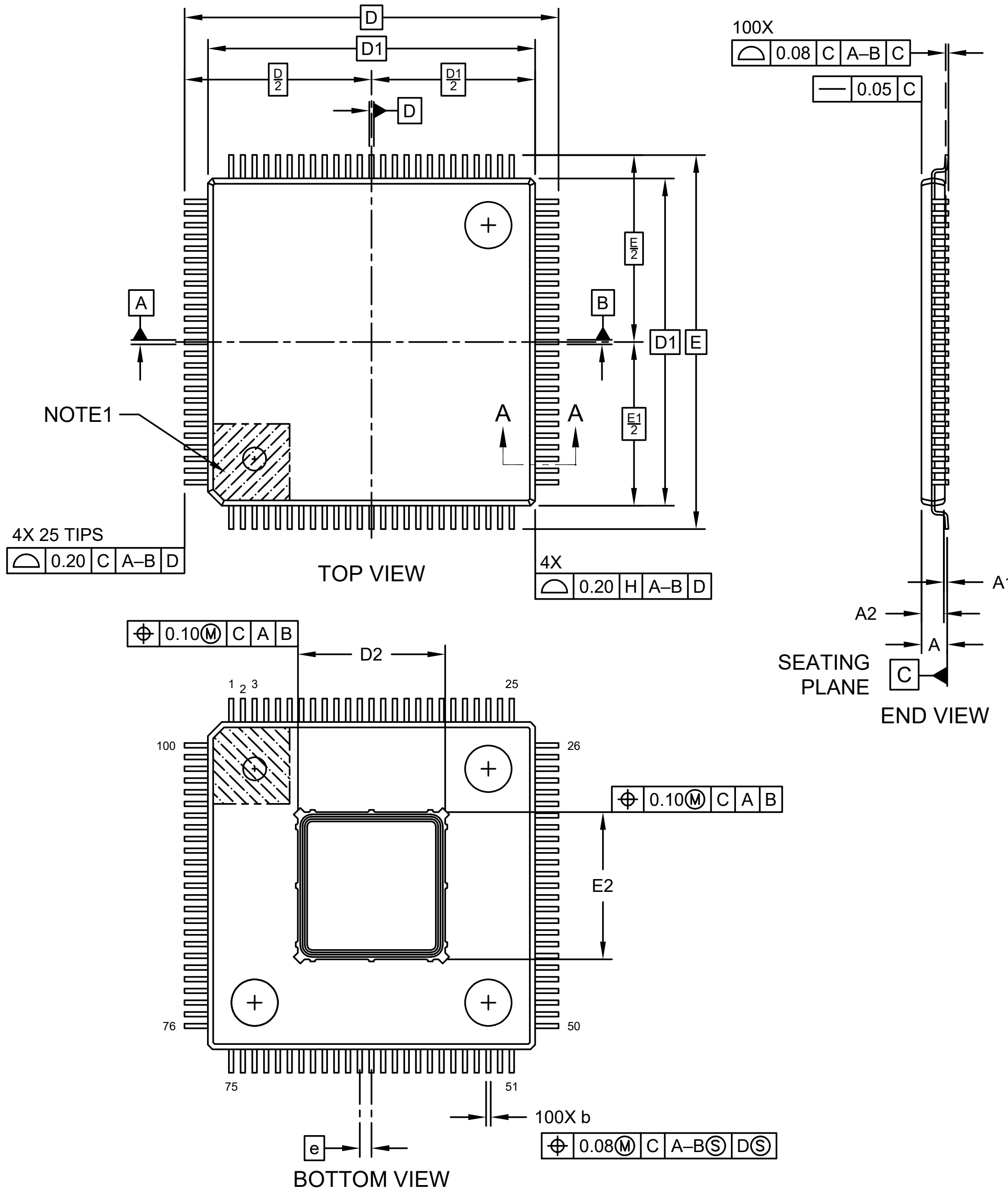
1. Dimensioning and tolerancing per ASME Y14.5M
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.
2. For best soldering results, thermal vias, if used, should be filled or tented to avoid solder loss during reflow process

Microchip Technology Drawing C04-2514 Rev A

54.2.4. 100-Lead TQFP Package

100-Lead Plastic Thin Quad Flat Pack (4SB) - 14x14x1.0 mm Body [TQFP]
With 6.3 mm Grooved Exposed Pad

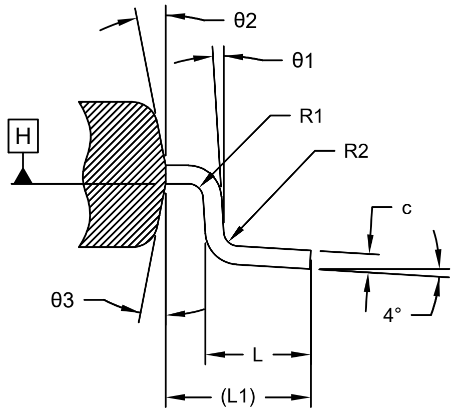
Note:	For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



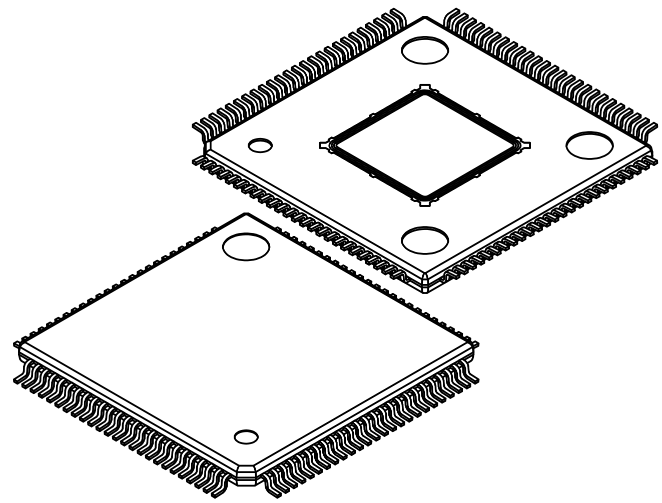
Microchip Technology Drawing C04-21553 Rev A Sheet 1 of 2

100-Lead Plastic Thin Quad Flat Pack (4SB) - 14x14x1.0 mm Body [TQFP] With 6.3 mm Grooved Exposed Pad

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		100	
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		16.00 BSC	
Overall Length	D1		14.00 BSC	
Exposed Pad Length	D2	6.20	6.30	6.40
Overall Width	E		16.00 BSC	
Overall Width	E1		14.00 BSC	
Exposed Pad Width	E2	6.20	6.30	6.40
Terminal Width	b	0.17	0.20	0.27
Terminal Thickness	c	0.09	0.13	0.20
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°
Mold Draft Angle	θ3	11°	12°	13°

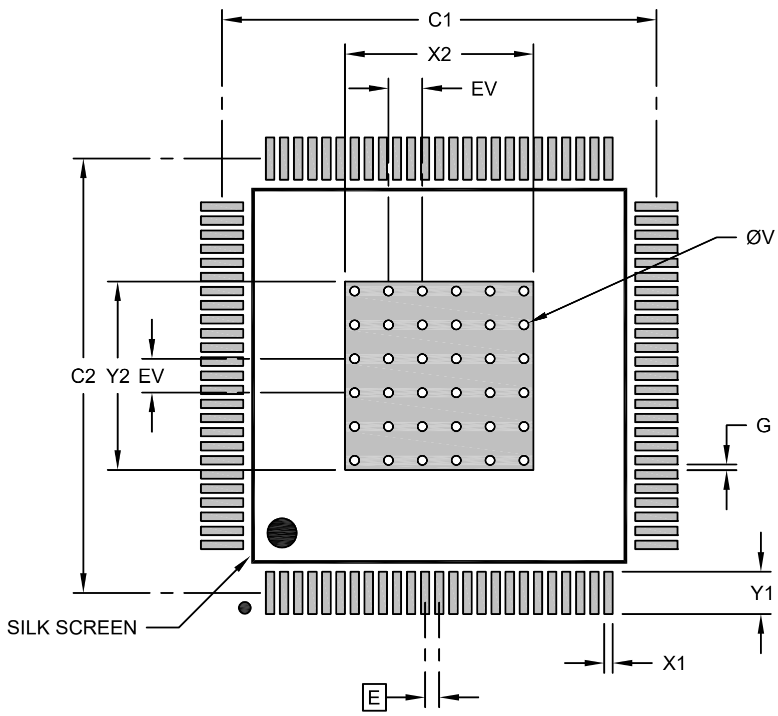
Notes:

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 C04-21553 Rev A Sheet 2 of 2

100-Lead Plastic Thin Quad Flat Pack (4SB) - 14x14x1.0 mm Body [TQFP]
With 6.3 mm Grooved Exposed Pad

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	
Center Pad Width	X2			6.68
Center Pad Length	Y2			6.68
Contact Pad Spacing	C1		15.40	
Contact Pad Spacing	C2		15.40	
Contact Pad Width (Xnn)	X1			0.30
Contact Pad Length (Xnn)	Y1			1.50
Contact Pad to Contact Pad (Xnn)	G	0.20		
Thermal Via Diameter	V		0.33	
Thermal Via Pitch	EV		1.20	

Notes:

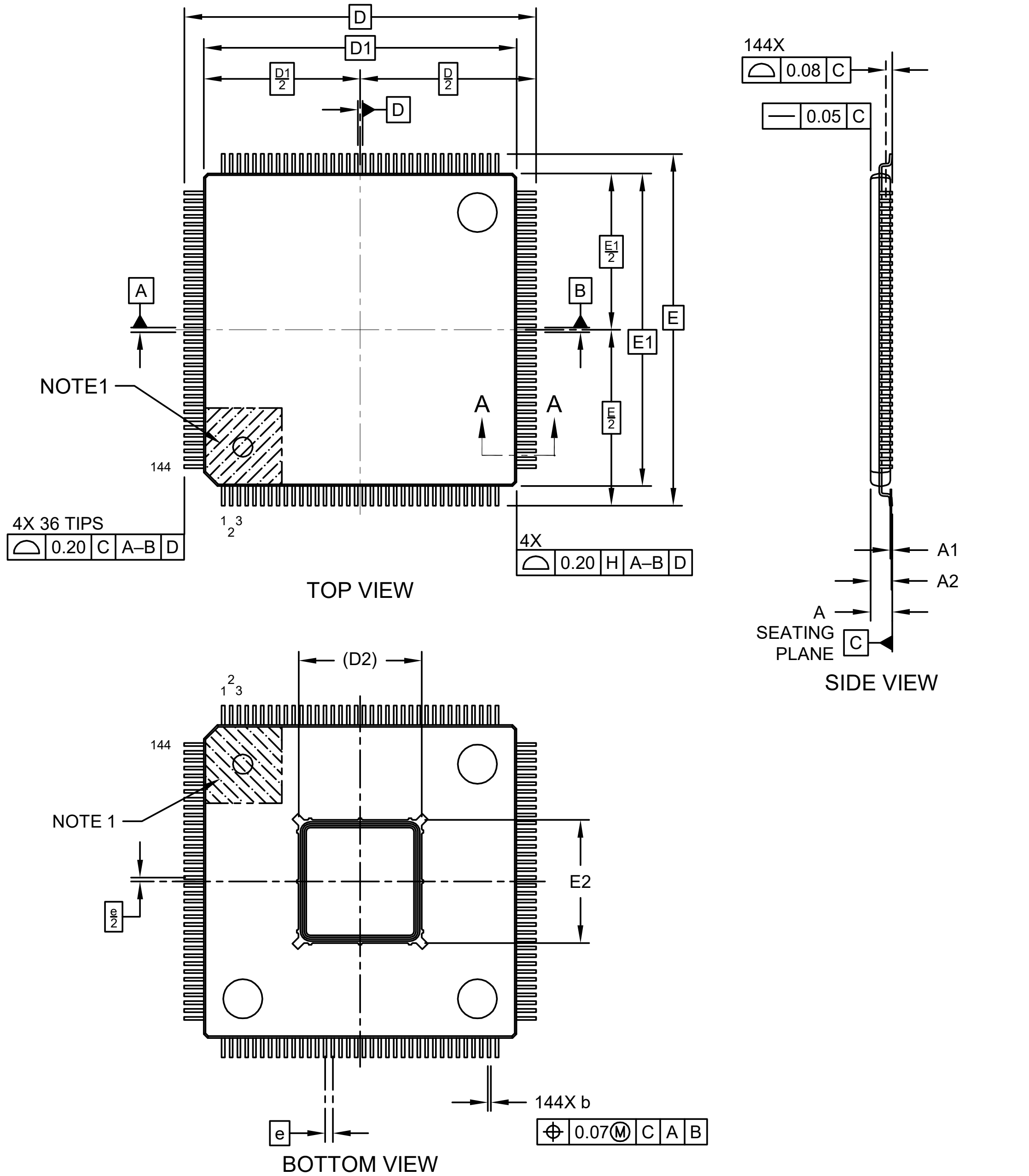
1. Dimensioning and tolerancing per ASME Y14.5M
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.
2. For best soldering results, thermal vias, if used, should be filled or tented to avoid solder loss during reflow process

Microchip Technology Drawing C04-23553 Rev A

54.2.5. 144-Lead TQFP-EP Package

**144-Lead Plastic Thin Quad Flatpack (4KB) - 16x16 mm Body [TQFP]
With 6.3 mm Grooved Exposed Pad**

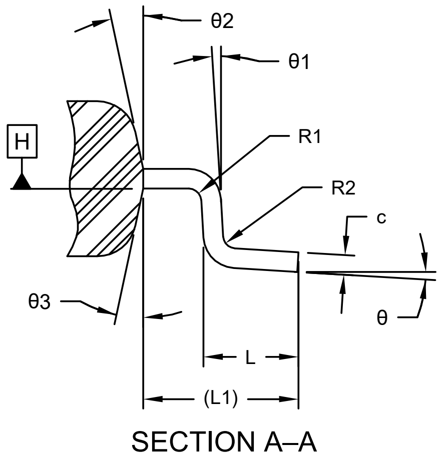
Note:	For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging

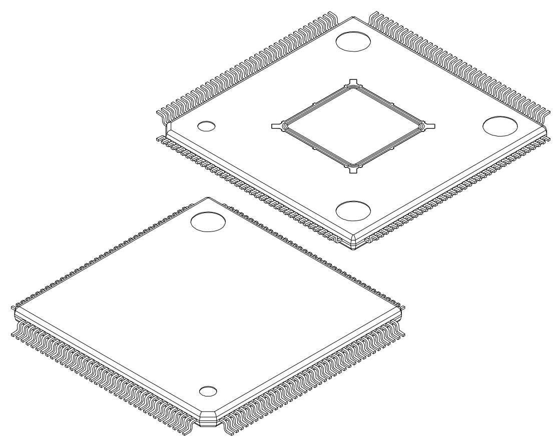


Microchip Technology Drawing C04-21546 Rev A Sheet 1 of 2

144-Lead Plastic Thin Quad Flatpack (4KB) - 16x16 mm Body [TQFP]
With 6.3 mm Grooved Exposed Pad

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		144	
Pitch	e		0.40 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		18.00 BSC	
Molded Package Length	D1		16.00 BSC	
Exposed Pad Length	D2		6.30 REF	
Overall Width	E		18.00 BSC	
Molded Package Width	E1		16.00 BSC	
Exposed Pad Width	E2		6.30 REF	
Terminal Width	b	0.13	0.16	0.23
Terminal Thickness	c	0.09	.127	0.20
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°
Mold Draft Angle	θ3	11°	12°	13°

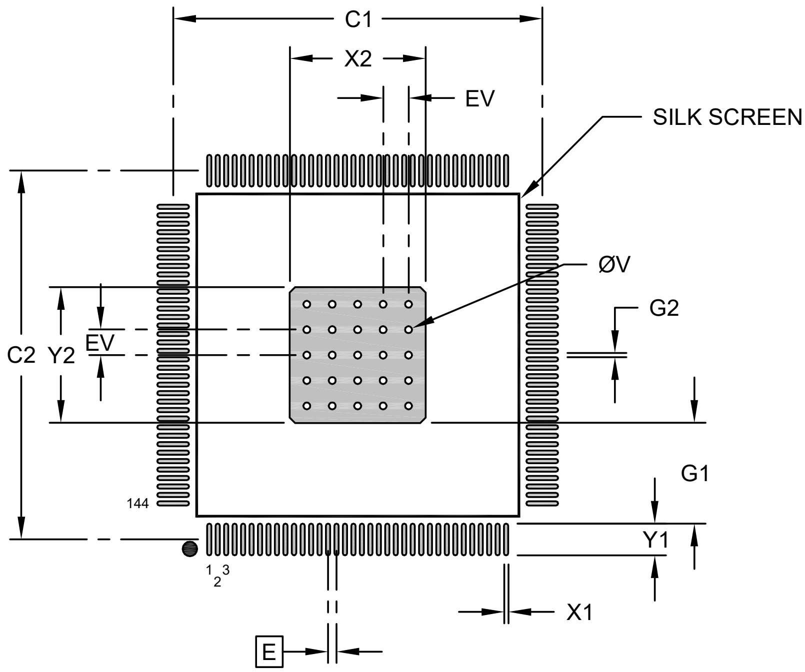
Notes:

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 C04-21546 Rev A Sheet 2 of 2

144-Lead Plastic Thin Quad Flatpack (4KB) - 16x16 mm Body [TQFP]
With 6.3 mm Grooved Exposed Pad

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



RECOMMENDED LAND PATTERN

		Units	MILLIMETERS	
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E		0.40 BSC	
Optional Center Pad Width	X2			6.40
Optional Center Pad Length	Y2			6.40
Contact Pad Spacing	C1		17.40	
Contact Pad Spacing	C2		17.40	
Contact Pad Width (Xnn)	X1			0.20
Contact Pad Length (Xnn)	Y1			1.50
Contact Pad to Center Pad (Xnn)	G1	4.75		
Contact Pad to Contact Pad (Xnn)	G2	0.20		
Thermal Via Diameter	V		0.33	
Thermal Via Pitch	EV		1.20	

Notes:

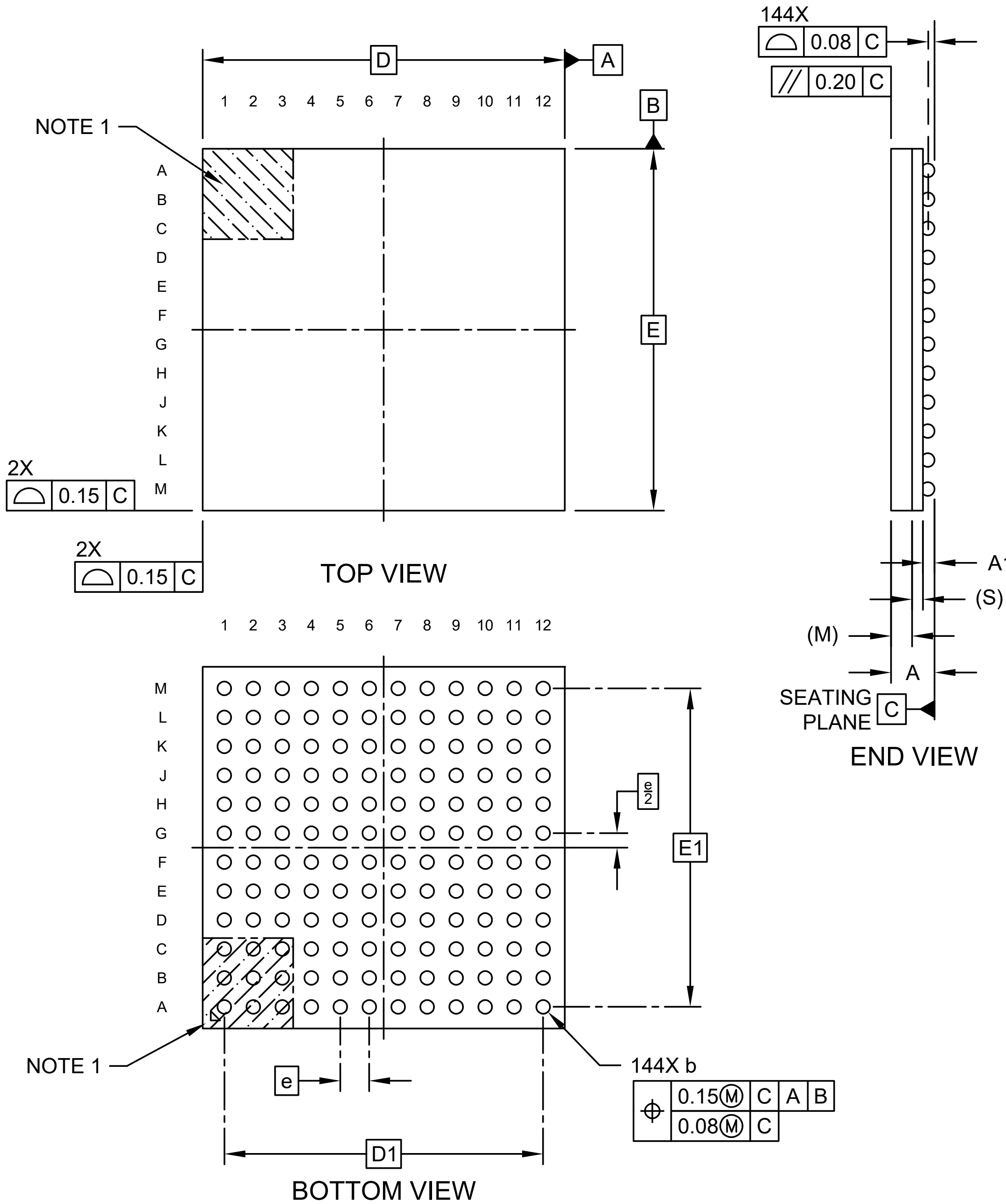
1. Dimensioning and tolerancing per ASME Y14.5M
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.
2. For best soldering results, thermal vias, if used, should be filled or tented to avoid solder loss during reflow process

Microchip Technology Drawing C04-23546 Rev A

54.2.6. 144-Pin TFBGA Package

144-Ball Thin Fine-Pitch Ball Grid Array Package (HQB) - 10x10x1.2 mm Body [TFBGA]

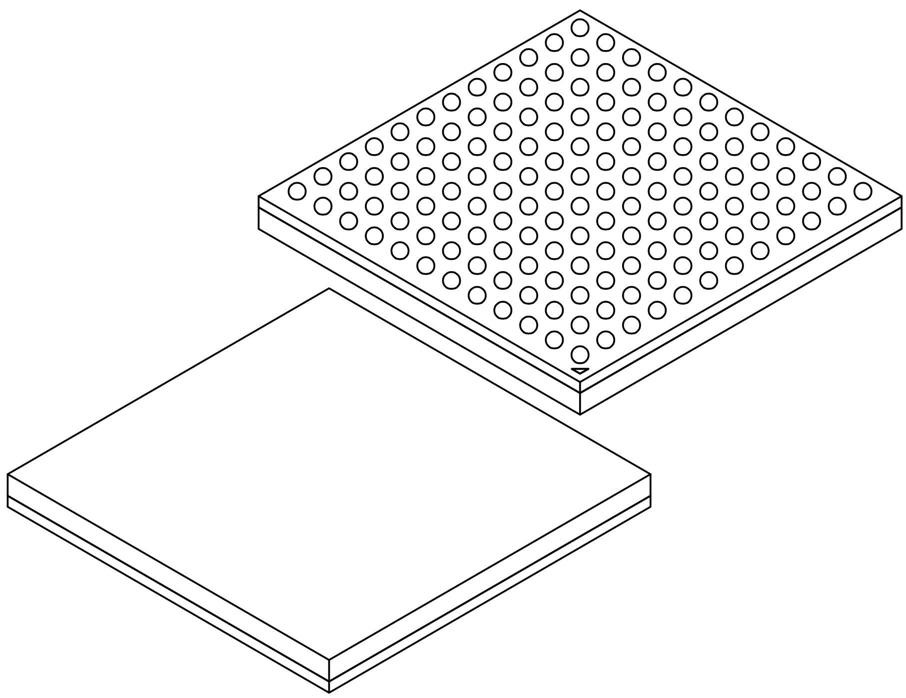
Note:	For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



Microchip Technology Drawing C04-21565 Rev A Sheet 1 of 2

144-Ball Thin Fine-Pitch Ball Grid Array Package (HQB) - 10x10x1.2 mm Body [TFBGA]

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	144		
Pitch	e	0.80 BSC		
Overall Height	A	–	–	1.20
Ball Height	A1	0.22	–	0.32
Mold Thickness	M	0.53 REF		
Substrate Thickness	S	0.246 REF		
Overall Length	D	10.00 BSC		
Ball Array Length	D2	8.80 BSC		
Overall Width	E	10.00 BSC		
Ball Array Width	E2	8.80 BSC		
Ball Diameter	b	0.32	–	0.42

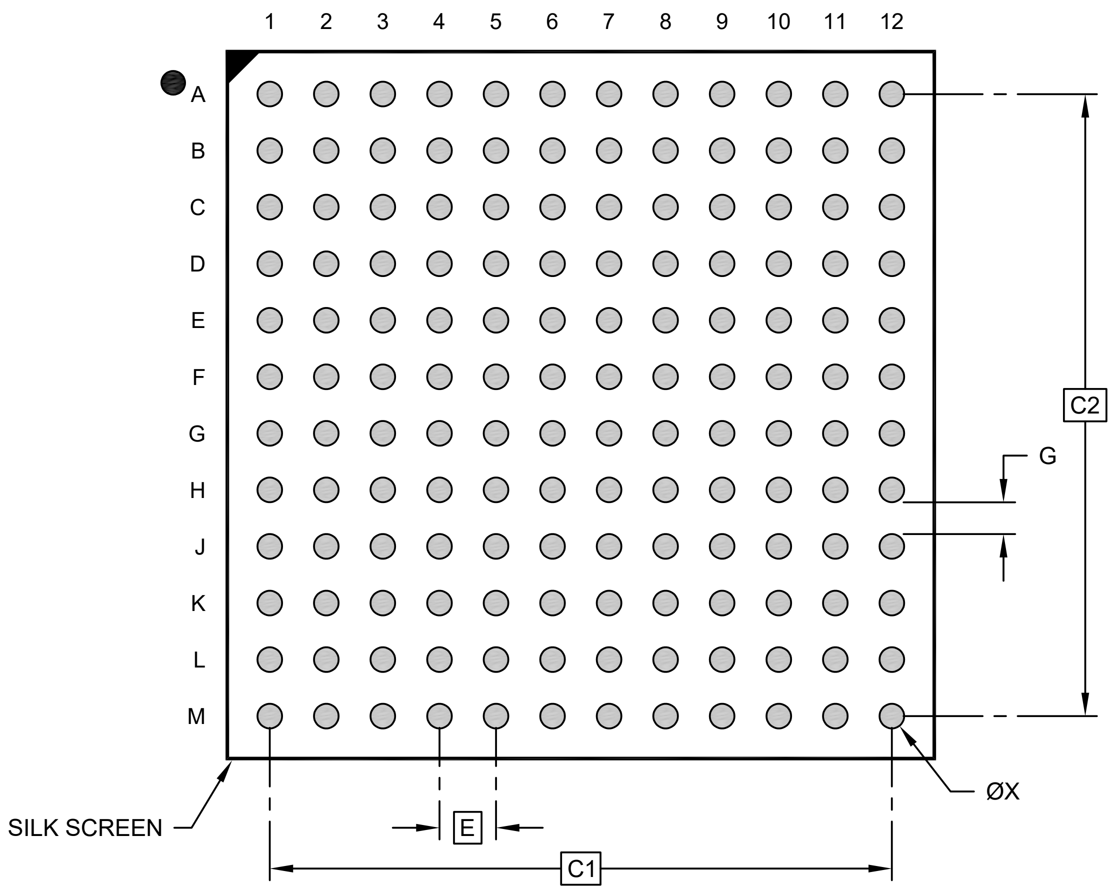
Notes:

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 C04-21565 Rev A Sheet 2 of 2

144-Ball Thin Fine-Pitch Ball Grid Array Package (HQB) - 10x10x1.2 mm Body [TFBGA]

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		0.80 BSC	
Contact Pad Spacing	C1		8.80 BSC	
Contact Pad Spacing	C2		8.80 BSC	
Contact Pad Diameter (X144)	X			0.35
Contact Pad to Contact Pad	G	0.45		

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 C04-23565 Rev A

54.3. Soldering Profile

The following table provides the recommended soldering profile from J-STD-20.

Table 54-1. Recommended Soldering Profile

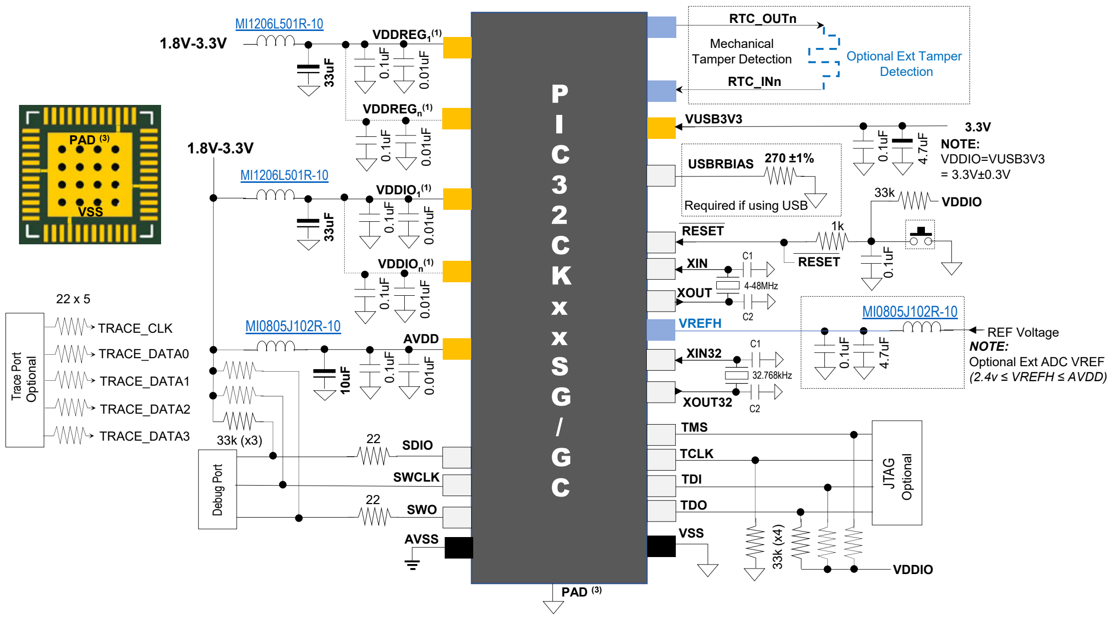
Profile Feature	Green Package
Average Ramp-up Rate (217°C to peak)	3°C/s max.
Preheat Temperature 175°C ±25°C	150-200°C
Time Maintained Above 217°C	60-150s
Time within 5°C of Actual Peak Temperature	30s
Peak Temperature Range	260°C
Ramp-down Rate	6°C/s max.
Time 25°C to Peak Temperature	8 minutes max.

A maximum of three reflow passes is allowed per component.

55. Schematic Checklist

55.1. Introduction

This chapter describes a common checklist which should be used when starting and reviewing the schematics for a PIC32CK SG/GC design. This chapter illustrates the recommended power supply connections, how to connect the external analog references, programmer, and debugger.



56. Common Conventions

56.1. Numerical Notation

Table 56-1. 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

56.2. Memory Size and Type

Table 56-2. 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

56.3. Frequency and Time

Table 56-3. Frequency and Time

Symbol	Description
kHz	1 kHz = 10^3 Hz = 1,000 Hz
KHz	1 KHz = 1,024 Hz, 32 KHz = 32,768 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
µs	microsecond
ns	nanosecond

56.4. Registers and Bits

Table 56-4. 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.

Table 56-4. Register and Bit Mnemonics (continued)

Symbol	Description
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.

57. Acronyms and Abbreviations

The below table contains acronyms and abbreviations used in this document.

Table 57-1. 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
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
DMAC	DMA (Direct Memory Access) Controller
DSU	Device Service Unit
EEPROM	Electrically Erasable Programmable Read-Only Memory
EIC	External Interrupt Controller
EVSYS	Event System
FREQM	Frequency Meter
GCLK	Generic Clock Controller
GND	Ground
GPIO	General Purpose Input/Output
I^2C	Inter-Integrated Circuit
IF	Interrupt Flag
INT	Interrupt
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
OSC	Oscillator
PAC	Peripheral Access Controller
PC	Program Counter
PER	Period

Table 57-1. Acronyms and Abbreviations (continued)

Abbreviation	Description
PLL	Phase Locked Loop
PM	Power Manager
POR	Power-on Reset
PORT	I/O Pin Controller
PWM	Pulse-Width Modulation
RAM	Random-Access Memory
REF	Reference
RTC	Real-Time Counter
RX	Receiver/Receive
SEEP	SmartEEPROM Page
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
V_{DD}	Common voltage to be applied to VDDIO and AVDD
V_{DDIN}	Digital Supply Voltage
V_{DDIO}	Digital Supply Voltage
AV_{DD}	Analog Supply Voltage
VREF	Voltage Reference
WDT	Watchdog Timer
XOSC	Crystal Oscillator

58. Revision History

Revision G - June 2025

Section	Description
Up to 2 MB Live-Update Flash and 512 KB SRAM with Hardware Security Module (HSM) for Secure Connectivity Applications	• Added in new temperature range for Operating Conditions
Package and Pinout	• Updates to Signal Names for the EIC, PTC, and CAN were implemented for the following tables: – 48-Pin VQFN Packages – 64-Pin TQFP/VQFN Packages – 100-Pin TQFP Package – 144-Pin TQFP Package – 144-Pin TFBGA Package
Signal Description	• Replaced the Serial Communication Interface (SERCOM) Signals table with a new table
Multi-Channel RAM Controller (MCRAMC)	• All new updated chapter
Clock Distribution System	• Added a new note to Overview for asynchronicity
PORT	• Added the following Registers: – INTENCLR – INTENSET – INTFLAG – NONSEC – NSCHK
WDT	• Updated the EWOFFSET bitfield of the EWCTRL Register with a new signal name, replacing GCLK_WDT with CLK_WDT_OSC
DMAC	• Updated the Register output for all Channel k Registers to properly display in the Register Summary
ETH	• Updated the table in Signal Interface, replacing ETH_TXCK1 with ETH_TX_CLK
SERCOM USART	• Updated the bit value from 8 to 16 in the Clock Generation Diagram in Clock Generation and Selection • Updated *Parity* to include 0x5 for error checking • Updated the PMODE bit in the CTRLB Register
SERCOM I^{2}C	• Added a Table to the Block Diagram for I^{2}C Pad Descriptions.
TCC	• Updated the Diagram for the Circular Buffer

Revision History (continued)

Section	Description
Electrical Characteristics 85°C	• Added note 4 to the following sections: – Non-Volatile Memory Controller (NVM) Electrical Specifications – Universal Serial Bus (USB) Electrical Specifications • Added in new diagrams to the following sections: – MCU Active Power – MCU Idle Power – MCU Standby Power
Electrical Characteristics 125°C	• All new Electrical Specifications chapter

Revision F - December 2024

Section	Description
Up to 2 MB Live-Update Flash and 512 KB SRAM with Hardware Security Module (HSM) for Secure Connectivity Applications	• Removed erroneous 512K references
Pinout	• Updated the table titles in 64-Pin TQFP Packages to correct the family name • Updated the table titles in 100-Pin TQFP Packages to correct the family name and updated erroneous VSS references to new Pad names
Product Mapping	• Removed erroneous 512K content from the following sections: – Code Address Space – SRAM Address Space – Peripheral Address Space – External RAM Address Space
Memories	• Updated the Physical Memory Map with a new table

Revision E - October 2024

Section	Description
Configuration Summary	• Added new tables to reflect new package offerings
Package and Pinout	• Added a new pinout for the 144-pin TFBGA and 48-pin VQFN
Signal Descriptions	• Updated all tables to display new pin counts for the 48-pin TQFP and 144-pin TFBGA
Power Supplies and Startup Considerations	• Updated Table 7-1 with new signal names • Updated the Power Domain Block Diagram with new signal names • Updated Power Domain Constraints with new signal names • Updated Signal names for figure 7-2 in Typical Powering Schematic • Updated signal names in Supply Order
Memories	• Added a note to FUCFG3

Revision History (continued)

Section	Description
SUPC	• Added a note to Overview • Updated the *Block Diagram* with new signal names
RSTC	• Updated the BORVDDA bitfield of the RCAUSE Register with a new note
PORT	• Added a new table to the PMUXm Register
TCC	• Added timers descriptions to the Overview • Added a new column for Counter Size to the Feature Summary • Updated the number of comparators in the Block Diagram • Added notes to the following Registers: – PER – CCy – PERBUF – CCBUFy
CCL	• Added a note to the INSELy bitfield of the LUTCTRLn Register
ADC	• Updated the input Signal names from AIN to VINP throughout the chapter
Electrical Characteristics (85°C)	• Added in the 144-pin TFBGA package for the Thermal Packaging Characteristics table • Updated signal names in Analog-to-Digital Converter (ADC) Electrical Specifications
Packaging	• Added a new 48-pin VQFN • Added a new 144-pin TFBGA Package

Revision D - July 2024

Section	Description
Package and Pinout	• Updated the following Pinouts updating VCCAPP3 to VUSB3V3: – 100-pin TQFP – 144-pin TQFP
Signal Description	• Updated Table *6-13 PORTA through PORTD Signals* with a new pin number for PB00
Clock Distribution System	• Updated the Clock Distribution Block Diagram converting MCLK to GCLK
TrustRAM (TRAM)	• Updated the Overview with a new value for the security RAM • Updated the Address Count in the diagram for Data Remanence Prevention • Updated the RAM register with a new offset equation
Electrical Characteristics (85°C)	• Added eight new graphs for IOH, VOL, VOH, and IOL to the I/O Pin Electrical Specifications

Revision C - April 2024

Section	Description
General	Minor changes of format. • *VREFP0* changed by *VREFH* • *SPIxC- TRL* changed by *SPIxCTRL* • *VDDF* changed by *VDDFLASH* • *ADC0* changed by *ADC* • *ADCn* changed by *ADC* • *INSELx* changed by *INSELy* • Remove VBAT references
Features	• Updated info of USB2.0 and Advanced Analog Features and Touch. Removed NIST compliant information.
Configuration Summary	• Updated *Brown-out Detection* info in: Table 1-1 and Table 1-2
Guidelines for Getting Started	• Replaced *Pie-Filters* by *Pi-Filters* in the *EMI/EMC/EFT (IEC61000-4-4 and IEC 61000-4-2) Suppression Considerations* section
Package and Pinout	• Updated MISC column in: – Table 5-6 – Table 5-8 – Table 5-9
Signal Description	• Added info of 64-Pin and 100-Pin in all the tables
Power Supplies and Startup Considerations	• Removed VBAT in Figure 7-1
Product Mapping	• Removed *Base Address* in: – Table 8-5 – Table 8-6 – Table 8-7 – Table 8-8
Peripherals	• Updated Figure 9-1
Processor and Architecture	• Modified format of Table 10-2 and added note
Memories	• 0x0054 is Reserved in User Configuration Map • Updated FUCFG5 - User Configuration Register 5 – Reserved
Cortex-M Cache Controller (CMCC)	• Updated table in *Peripheral Dependencies* section • Updated reference section in Tightly Coupled Memory section • Minor changes in Cache Invalidate by Line Operation section • Minor changes in Cache Performance Monitoring section
Peripheral Access Controller (PAC)	• Updated table in Peripheral Dependencies section • Removed bullet in the *Register Access Protection* section

Revision History (continued)

Section	Description
Device Service Unit (DSU)	• Updated Features section • Added notes in Signal Description section • Updated table in Peripheral Dependencies section • Updated first paragraph in Cold Plugging • Removed reference in Boot Communication Channels section
Clock Distribution System	• Updated Figure 18-1 • Minor changes in Overview section
Oscillator Controller (OSCCTRL)	• Updated table in Peripheral Dependencies section • Replaced *XOSCCTRLB[3]* by *XOSCCTRLB.GBW[1:0]* and *XOSCCTRLB[1:0]* by *XOSCCTRLB.GMAN[1:0]* in External Multipurpose Crystal Oscillator (XOSC) Operation section • Replaced *DFLL48M Loose Lock* by *DFLL48M Lose Lock After Wake* and added note in Additional DFLL48M Features section • Added note in XOSCFAIL bitfield in the STATUS Register • Added note in BWSEL bitfield in the PLL0CTRL Register • Added note in POSTDIVn bitfield in the PLL0POSTDIVA Register
Generic Clock Controller (GCLK)	• Updated Generic Clock Generators to 12 and updated Peripheral Channels to 48 • Updated the *Peripheral Clock* • Added Enabling a Peripheral Clock • Added note 5 in the Selecting the Clock Source for a Peripheral section • Updated the GENCTRLn Register • Updated PCHCTRLm Register
Main Clock (MCLK)	• Updated Figure 21-2
32 KHz Oscillators Controller (OSC32KCTRL)	• Updated bits of XOSC32K.STARTUP in 32 kHz External Crystal Oscillator (XOSC32K) Operation section
Frequency Meter (FREQM)	• Minor changes of format in Overview and Measurement sections • Added notes in SWRST bitfield in the CTRLA Register • Added note in REFNUM bitfield in the CFGA Register
Real-Time Counter (RTC)	• Updated Figure 25-3 • Minor changes in Active Layer Protection section • Added Important note in CTRLA Register
Direct Memory Access Controller (DMAC)	• Updated channels of DMA Controller modules 0 and 1 • Updated DMA Event/Trigger Mapping
Supply Controller (SUPC)	• Updated Features section • Minor changes in Block Diagram section and removed VBAT in Figure 27-1 • Updated table in Peripheral Dependencies section • Removed *Battery Backup Power Switch* section • Added note in LVHIB bitfield in the VREGCTRL Register • Added note in LPHIB and LPSTDBY bitfields in the VREFCTRL Register

Revision History (continued)

Section	Description
Power Manager (PM)	• Updated VREGSW in Table 28-3 • Updated the *Standby Mode* section • Updated the *Hibernate and Backup Modes* section • Removed a sentence in I/O Line State Retention in Hibernate and Backup Modes section
Reset Controller (RSTC)	• Removed VBAT reference in Features and the *Reset Causes and Effects* sections
External Interrupt Controller (EIC)	• Replaced *EXTINTx* by *EIC_EXTINTx* in Figure 30-1 • Minor changes in Initialization section • Added note in NMIFLAG Register
Event System (EVSYS)	• Minor changes in Edge Detection section • Added Important note in PATH bitfield in CHANNELn Register
Serial Communication Interface (SERCOM)	• Minor changes of format in Clock Generation – Baud-Rate Generator section • Updated FIFO Operation section
Serial Quad Interface (SQI)	• Updated Chip Selects to 4 • Updated Figure 36-1 • Updated CSEN bitfield in CFG Register
Universal Serial Bus Hi-Speed (USBHS)	• Minor changes of format in Overview section • Replaced *UPLLFSEL* by *CTRLA.REFCLKSEL* in Block Diagram section • Removed *OTG* in Signal Description section
Controller Area Network (CAN)	• Updated number of CAN in Features section • Updated note in Block Diagram section • Updated Table 39-1 • Minor changes in the *Clocks* section
True Random Number Generator (TRNG)	• Updated second paragraph in Overview section • Removed one bullet in Features section
Configurable Custom Logic (CCL)	• Removed *Timer/Counter Inputs (TC)* section • Removed note in Timer/Counter for Control Application Inputs (TCC) section • Updated INSELy bitfield in LUTCTRLn Register

Revision History (continued)

Section	Description
Analog-to-Digital Converter (ADC)	• Updated Overview section • Updated Features section • Removed note in Block Diagram section • Removed note 3 in Singnal Description section • Added Important note in Conversion Timing and Sampling Rate section • Updated Initialization section • Updated Interrupts section • Updated Events section • Updated SWCNVEN, SAMP and RQCNVRT bitfields in CTRLB Register • Removed VBAT reference in ADCHSEL bitfield in CTRLB Register • Bits 14 and 30 as reserved in CONFIG[0].CHNCFG1 Register • Updated LVLk and CHNCMPEk bitfields in CONFIG[0].CHNCFG1 Register • Updated FRACk and CSSk bitfields in CONFIG[0].CHNCFG2 Register • Updated note 5 of TRGSRCk bitfield in CONFIG[0].CHNCFG4 and CONFIG[0].CHNCFG5 Registers
Analog Comparators (AC)	• Updated MUXPOS bitfield in COMPCTRL0 Register
Position Decoder (PDEC)	• Minor changes in Secure Decoder Detection section • Updated Re-trigger Software Command or Event Action section • Updated EVACT bitfield in EVCTRL Register
Timer/Counter for Control Applications (TCC)	• Updated number of TCC in Overview section • Updated number of Compare/Capture Channels in Features • Updated values of table in the *Feature Summary* section • Updated table in Signal Description section • Updated value of TCC bus clocks in the *Clocks* section • Updated Principle of Operation section • Minor changes in Initialization, Enabling, Disabling, and Resetting, Prescaler Selection, Counter Operation and Circular Buffer sections • Removed last paragraph of Stop Command in Counter Operation section • Added note in Alternate RAMP2 (RAMP2A) Operation section • Updated Table 48-4 • Added Important note in DMA Operation section • Updated CPTEN bitfield in CTRLA Register • Added note in DIR bitfield in CTRLBSET • Updated CC bitfield in SYNCBUSY Register • Updated CCBUFV and CMP bitfields in STATUS Register • Updated CCy Register • Updated CCBUFy Register
TrustRAM (TRAM)	• Added note 4 in CTRLA Register • Added note 2 in CTRLA Register
Peripheral Touch Controller (PTC)	• Minor changes in Clocks section
Electrical Characteristics (85°C)	• Updated Note 4

Revision B - May 2023

Section	Description
General	• Updated DPLL to read PLL throughout the document
Features	• Updated Advanced Analog Features and Touch instances with new data for channel numbers • Reformatted the Communication Interfaces/Digital Peripherals section • Updated the DMA instances
Configuration Summary	• Replaced the existing tables with all new content
Signal Description	• Replaced the existing tables with all new content
Block Diagram	• Updated the Diagram with a new image
Pinout	• Updated the following Pinout tables with all new content: – 144-pin TQFP – 100-pin TQFP – 64-pin QFN and TQFP
Power Supplies and Startup Considerations	• Updated the table in Power Supplies and Startup Considerations with new valued for the VDDREG • Updated the bulleted items in Power Domain Constraints removing references to VDDCORE_USB1 and VDDUSB1 • Updated the bulleted items in Power Supplies removing references to VDDUSB3V1 • Renamed Boot ROM to Boot Flash (BFM) in Initial Instructions Fetching • Updated VDDREG verbiage in Supply Order • Updated I/O Pins with all new verbiage
Processor and Architecture	• Removed the erroneous Overview section • Updated NVIC Interrupt Line Mapping to remove ULDOOVHEAT and ULDORDY
Memories	• Updated the Physical Memory Map • Added s new section, PFM Error Correction • Removed erroneous references to DAL and DAL-CFG in Configuration Flash Memory • Removed an erroneous section pertaining to DAL Configuration • Removed an erroneous section pertaining to CAL-OTP Configuration
CMCC	• Updated the WAYNUM bitfield in the TYPE Register with a new table
IDAU	• This chapter was completely rewritten in this revision
DSU	• Corrected erroneous bitfield output for the CPUx bitfield in the DAL Register • Removed an obsolete DATA Register Summary and associated Register
Clock Distribution System	• Updated the Block Diagram with an all new image • Updated the Diagram Example of a SERCOM_0 Clock to remove erroneous DPLL references

Revision History (continued)

Section	Description
OSCCTRL	• Removed erroneous clock references from the Block Diagram • Updated Phase Locked Loop (PLL) Operation to remove erroneous Digital references • Removed the following erroneous registers: – DPLL1CTRL – DPLL1FBDIV – DPLL1REFDIV – DPLL1POSTDIVA – FRACDIV1 • Added a new register, PLL0POSTDIVB
GCLK	• Updated Figure 19-3 in Generic Clock Generator, changed DPLL to PLL and removed erroneous clock signals • Corrected erroneous bitfield output in the following registers, and/or updated bitfield descriptions: – SYNCBUSY – GENCTRLn – PCHCTRLm
MCLK	• Added a new register, CLKDIV
OSC32KCTRL	• Updated the CGM and STARTUP bitfields to properly display content in the XOSC32K Register
WDT	• Updated the following registers to properly display bitfield information: – EWCTRL
RTC	• Updated the TAMPCTRL Register to properly display the INnACT bitfield output
SUPC	• Removed erroneous LDO information from the Block Diagram • Removed erroneous bitfields for ULDOOVHEAT, and ULDORDY from the following registers: – INTENCLR – INTENSET – INTFLAG – STATUS • Removed erroneous bitfields for BKUP_VLD, SRAM_VLD, ULDOLEVEL, ULDSTDBY, ULDOEN, and OFFSTDBY from the VREGCTRL Register • Added new information for the LVSTDBY and VREGOUT bitfields in the VREGCTRL Register
RSTC	• Updated Reset Causes and Effects to remove an erroneous reference to BOD33
NVMCTRL	• Replaced the image for the Block Diagram • Updated Configuration Flash Memory to remove erroneous DALCFG, CAL, and SDAL references

Revision History (continued)

Section	Description
ADC	• Updated throughput rates for the ADC in Features • Removed erroneous cores from the ADC block diagram, removed references to other modules from the ADC Trigger Block Diagram • Updated the verbiage in the *Signal Description* to remove erroneous mentions to modules • Updated the following registers with new naming conventions and/or verbiage updates to bitfields, including new notes and tables: – CONFIG[0].CORCTRL – CONFIG[0].CHNCFG1 – CONFIG[0].CHNCFG2 – CONFIG[0].CHNCFG3 – CONFIG[0].CHNCFG5 – CONFIG[0].CALCTRL – EVCTRL0 – FLTCTRL0 – CORCHDATAID – PFFDATA – SYNCBUSY – INTENCLR – INTFLAG • Added a new note to the CTRLC Register for the CNT bitfield
Electrical Specifications 85°C	• Updated the typ specifications for Wake-Up Timing • Updated FCLK_31 in Maximum Clock Frequencies • Updated Xosc_3 and XOSC_5, and added XOSC_39 to External Oscillator (XOSC) Electrical Specifications • Updated DFLL/PLL Electrical Specifications with new min, typ, and max specifications • Updated the tables in Analog-to-Digital Converter (ADC) Electrical Specifications with new min, typ, and max specifications • Replaced the tables in Comparator Electrical Specifications • Updated Peripheral Touch Controller (PTC) Electrical Specifications with new max specifications • Updated I^{2}S Host Mode AC Electrical Specifications with a new max specification for I2S_33 • Updated Timer Counter for Control Application (TCC) Electrical Specifications with a new max specification for TCC_15 • Updated Non-Volatile Memory Controller (NVM) Electrical Specifications with a new min specification for NVM_3 • Updated max specifications for SD_5 and SD_25 in SD/MMC Host Controller (SDHC) Electrical Specifications • Updated min and max specifications in JTAG Electrical Specifications • Updated SWD_1 max specifications in SWD 2-Wire Electrical Specifications
Extended Temperature Electrical Characterstics (125°C)	• All new section
Schematic Checklist	• Updated the voltage specifications in the Diagram in Introduction

Revision History (continued)

Section	Description
Packaging	• Updated the following packages with new images and/or new information: – 64-pin TQFP – 64 Lead VQFN – 100 Lead TQFP – 144 Lead TQFP

Revision A - November 2022

Terminology used in this document may not match with the contents of other Microchip documentation and collateral. For any questions or concerns regarding terminology, contact a Microchip support or sales representative.

This is the initial released version of this document.

59. Product Identification System

To order or obtain information, e.g., on pricing or delivery, refer to the factory or the listed sales office.

PART NO.	[X][(1)]	–X	/XX
Device	Tape and Reel	Temperature Range	Package

Device:	Device A, Feature A, (Package A) Device B, Feature B, (Package B)	
Tape & Reel Option:	Blank	= Tube
	T	= Tape & Reel
Temperature Range:	I	= -40°C to +85°C (Industrial)
	E	= -40°C to +125°C (Extended)
Package:	AA	= Package AA
	BB	= Package BB

Examples:

- MCPXXXXXAT-E/AA: Tape and Reel, Extended temperature, XAA package
- MCPXXXXXBT-E/BB: Tape and Reel Extended temperature, XBB package

Notes:

1. Tape and Reel identifier only appears in the catalog part number description. This identifier is used for ordering purposes and is not printed on the device package. Check with your Microchip Sales Office for package availability with the Tape and Reel option.
2. Small form-factor packaging options may be available. Please check www.microchip.com/packaging for small-form factor package availability, or contact your local Sales Office.

Microchip Information

Trademarks

ISBN: 979-8-3371-1450-7

Legal Notice

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 products are 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.

Product Page Links

PIC32CK1025GC01064, PIC32CK1025GC01100, PIC32CK1025GC01144, PIC32CK1025SG01064, PIC32CK1025SG01100, PIC32CK1025SG01144, PIC32CK2051GC01064, PIC32CK2051GC01100, PIC32CK2051GC01144, PIC32CK2051SG01064, PIC32CK2051SG01100, PIC32CK2051SG01144