**Welcome to E-XFL.COM**

## What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

## Applications of "Embedded - Microcontrollers"

| Details | |
|---|---|
| Product Status | Active |
| Core Processor | ARM® Cortex®-M0+ |
| Core Size | 32-Bit Single-Core |
| Speed | 48MHz |
| Connectivity | I²C, LINbus, SPI, UART/USART, USB |
| Peripherals | Brown-out Detect/Reset, DMA, I²S, POR, PWM, WDT |
| Number of I/O | 38 |
| Program Memory Size | 128KB (128K x 8) |
| Program Memory Type | FLASH |
| EEPROM Size | - |
| RAM Size | 16K x 8 |
| Voltage - Supply (Vcc/Vdd) | 1.62V ~ 3.6V |
| Data Converters | A/D 14x12b; D/A 1x10b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 125°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 48-TQFP |
| Supplier Device Package | 48-TQFP (7x7) |
| Purchase URL | https://www.e-xfl.com/product-detail/microchip-technology/atsamd21g17a-af |

### 13.5.3 Clocks

The DSU bus clocks (CLK_DSU_APB and CLK_DSU_AHB) can be enabled and disabled by the Power Manager. Refer to *PM – Power Manager*

**Related Links**

PM – Power Manager

### 13.5.4 DMA

Not applicable.

### 13.5.5 Interrupts

Not applicable.

### 13.5.6 Events

Not applicable.

### 13.5.7 Register Access Protection

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

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

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

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.

**Related Links**

PAC - Peripheral Access Controller

### 13.5.8 Analog Connections

Not applicable.

## 13.6 Debug Operation

### 13.6.1 Principle of Operation

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

- CPU reset extension
- Debugger probe detection

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

### 13.6.2 CPU Reset Extension

“CPU reset extension” refers to the extension of the reset phase of the CPU core after the external reset is released. This ensures that the CPU is not executing code at startup while a debugger connects to the system. It is detected on a $\overline{\text{RESET}}$ release event when SWCLK is low. At startup, SWCLK is internally pulled up to avoid false detection of a debugger if SWCLK is left unconnected. When the CPU is held in

| Offset | Name | Bit Pos. | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x100C ... 0x1FCB | Reserved | | | | | | | | | |
| 0x1FCC | MEMTYPE | 7:0 | | | | | | | | SMEMP |
| 0x1FCD | | 15:8 | | | | | | | | |
| 0x1FCE | | 23:16 | | | | | | | | |
| 0x1FCF | | 31:24 | | | | | | | | |
| 0x1FD0 | PID4 | 7:0 | FKBC[3:0] | | | | JEPCC[3:0] | | | |
| 0x1FD1 | | 15:8 | | | | | | | | |
| 0x1FD2 | | 23:16 | | | | | | | | |
| 0x1FD3 | | 31:24 | | | | | | | | |
| 0x1FD4 ... 0x1FDF | Reserved | | | | | | | | | |
| 0x1FE0 | PID0 | 7:0 | PARTNBL[7:0] | | | | | | | |
| 0x1FE1 | | 15:8 | | | | | | | | |
| 0x1FE2 | | 23:16 | | | | | | | | |
| 0x1FE3 | | 31:24 | | | | | | | | |
| 0x1FE4 | PID1 | 7:0 | JEPIDCL[3:0] | | | | PARTNBH[3:0] | | | |
| 0x1FE5 | | 15:8 | | | | | | | | |
| 0x1FE6 | | 23:16 | | | | | | | | |
| 0x1FE7 | | 31:24 | | | | | | | | |
| 0x1FE8 | PID2 | 7:0 | REVISION[3:0] | | | | JEPU | JEPIDCH[2:0] | | |
| 0x1FE9 | | 15:8 | | | | | | | | |
| 0x1FEA | | 23:16 | | | | | | | | |
| 0x1FEB | | 31:24 | | | | | | | | |
| 0x1FEC | PID3 | 7:0 | REVAND[3:0] | | | | CUSMOD[3:0] | | | |
| 0x1FED | | 15:8 | | | | | | | | |
| 0x1FEE | | 23:16 | | | | | | | | |
| 0x1FEF | | 31:24 | | | | | | | | |
| 0x1FF0 | CID0 | 7:0 | PREAMBLEB0[7:0] | | | | | | | |
| 0x1FF1 | | 15:8 | | | | | | | | |
| 0x1FF2 | | 23:16 | | | | | | | | |
| 0x1FF3 | | 31:24 | | | | | | | | |
| 0x1FF4 | CID1 | 7:0 | CCLASS[3:0] | | | | PREAMBLE[3:0] | | | |
| 0x1FF5 | | 15:8 | | | | | | | | |
| 0x1FF6 | | 23:16 | | | | | | | | |
| 0x1FF7 | | 31:24 | | | | | | | | |
| 0x1FF8 | CID2 | 7:0 | PREAMBLEB2[7:0] | | | | | | | |
| 0x1FF9 | | 15:8 | | | | | | | | |
| 0x1FFA | | 23:16 | | | | | | | | |
| 0x1FFB | | 31:24 | | | | | | | | |
| 0x1FFC | CID3 | 7:0 | PREAMBLEB3[7:0] | | | | | | | |
| 0x1FFD | | 15:8 | | | | | | | | |
| 0x1FFE | | 23:16 | | | | | | | | |
| 0x1FFF | | 31:24 | | | | | | | | |

# 16. PM – Power Manager

## 16.1 Overview

The Power Manager (PM) controls the reset, clock generation and sleep modes of the device.

Utilizing a main clock chosen from a large number of clock sources from the GCLK, the clock controller provides synchronous system clocks to the CPU and the modules connected to the AHB and the APBx bus. The synchronous system clocks are divided into a number of clock domains; one for the CPU and AHB and one for each APBx. Any synchronous system clock can be changed at run-time during normal operation. The clock domains can run at different speeds, enabling the user to save power by running peripherals at a relatively low clock frequency, while maintaining high CPU performance. In addition, the clock can be masked for individual modules, enabling the user to minimize power consumption.

Before entering the STANDBY sleep mode the user must make sure that a significant amount of clocks and peripherals are disabled, so that the voltage regulator is not overloaded. This is because during STANDBY sleep mode the internal voltage regulator will be in low power mode.

Various sleep modes are provided in order to fit power consumption requirements. This enables the PM to stop unused modules in order to save power. In active mode, the CPU is executing application code. When the device enters a sleep mode, program execution is stopped and some modules and clock domains are automatically switched off by the PM according to the sleep mode. The application code decides which sleep mode to enter and when. Interrupts from enabled peripherals and all enabled reset sources can restore the device from a sleep mode to active mode.

The PM also contains a reset controller to collect all possible reset sources. It issues a device reset and sets the device to its initial state, and allows the reset source to be identified by software.

## 16.2 Features

- Reset control
  - Reset the microcontroller and set it to an initial state according to the reset source
  - Multiple reset sources
    - Power reset sources: POR, BOD12, BOD33
    - User reset sources: External reset ($\overline{\text{RESET}}$), Watchdog Timer reset, software reset
  - Reset status register for reading the reset source from the application code
- Clock control
  - Controls CPU, AHB and APB system clocks
    - Multiple clock sources and division factor from GCLK
    - Clock prescaler with 1x to 128x division
  - Safe run-time clock switching from GCLK
  - Module-level clock gating through maskable peripheral clocks
- Power management control
  - Sleep modes: IDLE, STANDBY
  - SleepWalking support on GCLK clocks

**Bit 0 – PAC1: PAC1 APB Clock Enable**

| Value | Description |
|---|---|
| 0 | The APBB clock for the PAC1 is stopped. |
| 1 | The APBB clock for the PAC1 is enabled. |

### 16.8.10 APBC Mask

**Name:** APBCMASK
**Offset:** 0x20
**Reset:** 0x00010000
**Property:** Write-Protected

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 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 |
|---|---|---|---|---|---|---|---|---|
| | | | | I2S | PTC | DAC | AC | ADC |
| Access | R | R | | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | | 0 | 0 | 0 | 0 | 1 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | TC7 | TC6 | TC5 | TC4 | TC3 | TCC2 | TCC1 | TCC0 |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | SERCOM5 | SERCOM4 | SERCOM3 | SERCOM2 | SERCOM1 | SERCOM0 | EVSYS | PAC2 |
| Access | R/W | 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 – I2S: I2S APB Clock Enable**

| Value | Description |
|---|---|
| 0 | The APBC clock for the I2S is stopped. |
| 1 | The APBC clock for the I2S is enabled. |

**Bit 19 – PTC: PTC APB Clock Enable**

| Value | Description |
|---|---|
| 0 | The APBC clock for the PTC is stopped. |
| 1 | The APBC clock for the PTC is enabled. |

**Bit 18 – DAC: DAC APB Clock Enable**

| Value | Description |
|---|---|
| 0 | The APBC clock for the DAC is stopped. |
| 1 | The APBC clock for the DAC is enabled. |

  - 32KHz to 2MHz input reference clock frequency range
  - Three possible sources for the reference clock
  - Adjustable proportional integral controller
  - Fractional part used to achieve 1/16th of reference clock step
- 3.3V Brown-Out Detector (BOD33)
  - Programmable threshold
  - Threshold value loaded from Flash User Calibration at startup
  - Triggers resets or interrupts
  - Operating modes:
    - Continuous mode
    - Sampled mode for low power applications (programmable refresh frequency)
  - Hysteresis
- Internal Voltage Regulator system (VREG)
  - Operating modes:
    - Normal mode
    - Low-power mode
  - With an internal non-configurable Brown-out detector (BOD12)
- 1.2V Brown-Out Detector (BOD12)
  - Programmable threshold
  - Threshold value loaded from Flash User Calibration at start-up
  - Triggers resets or interrupts
  - Operating modes:
    - Continuous mode
    - Sampled mode for low power applications (programmable refresh frequency)
  - Hysteresis
- Voltage Reference System (VREF)
  - Bandgap voltage generator with programmable calibration value
  - Temperature sensor
  - Bandgap calibration value loaded from Flash Factory Calibration at start-up

The OSCULP32K oscillator is used to clock the start-up counter.

| STARTUP[3:0] | Number of OSCULP32K Clock Cycles | Number of XOSC Clock Cycles | Approximate Equivalent Time(1)(2)(3) |
|---|---|---|---|
| 0x0 | 1 | 3 | 31µs |
| 0x1 | 2 | 3 | 61µs |
| 0x2 | 4 | 3 | 122µs |
| 0x3 | 8 | 3 | 244µs |
| 0x4 | 16 | 3 | 488µs |
| 0x5 | 32 | 3 | 977µs |
| 0x6 | 64 | 3 | 1953µs |
| 0x7 | 128 | 3 | 3906µs |
| 0x8 | 256 | 3 | 7813µs |
| 0x9 | 512 | 3 | 15625µs |
| 0xA | 1024 | 3 | 31250µs |
| 0xB | 2048 | 3 | 62500µs |
| 0xC | 4096 | 3 | 125000µs |
| 0xD | 8192 | 3 | 250000µs |
| 0xE | 16384 | 3 | 500000µs |
| 0xF | 32768 | 3 | 1000000µs |

**Note:**

1. Number of cycles for the start-up counter
2. Number of cycles for the synchronization delay, before PCLKSR.XOSCRDY is set.
3. Actual start-up time is n OSCULP32K cycles + 3 XOSC cycles, but given the time neglects the 3 XOSC cycles.

**Bit 11 – AMPGC: Automatic Amplitude Gain Control**

This bit must be set only after the XOSC has settled, indicated by the XOSC Ready flag in the PCLKSR register (PCLKSR.XOSCRDY).

| Value | Description |
|---|---|
| 0 | The automatic amplitude gain control is disabled. |
| 1 | The automatic amplitude gain control is enabled. Amplitude gain will be automatically adjusted during Crystal Oscillator operation. |

**Bits 10:8 – GAIN[2:0]: Oscillator Gain**

These bits select the gain for the oscillator. The listed maximum frequencies are recommendations, and might vary based on capacitive load and crystal characteristics. Setting this bit group has no effect when the Automatic Amplitude Gain Control is active.

**Table 18-1. WDT Operating Modes**

| CTRL.ENABLE | CTRL.WEN | INTENSET.EW | Mode |
|---|---|---|---|
| 0 | x | x | Stopped |
| 1 | 0 | 0 | Normal |
| 1 | 0 | 1 | Normal with Early Warning interrupt |
| 1 | 1 | 0 | Window |
| 1 | 1 | 1 | Window with Early Warning interrupt |

### 18.6.2 Basic Operation

#### 18.6.2.1 Initialization

The following bits are enable-protected:

- Window Mode Enable in the Control register (CTRL.WEN)
- Always-On in the Control register (CTRL-ALWAYSON)

The following registers are enable-protected:

- Configuration register (CONFIG)
- Early Warning Interrupt Control register (EWCTRL)

Any writes to these bits or registers when the WDT is enabled or is being enabled (CTRL.ENABLE=1) will be discarded. Writes to these registers while the WDT is being disabled will be completed after the disabling is complete.

Enable-protection is denoted by the Enable-Protected property in the register description.

Initialization of the WDT can be done 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 required, the Window Enable bit in the Control register (CTRL.WEN) must be written to one and the Window Period bits in the Configuration register (CONFIG.WINDOW) must be defined.

**Normal Mode**

- Defining the required Time-Out Period bits in the Configuration register (CONFIG.PER).

**Normal Mode with Early Warning interrupt**

- Defining the required Time-Out Period bits in the Configuration register (CONFIG.PER).
- Defining Early Warning Interrupt Time Offset bits in the Early Warning Interrupt Control register (EWCTRL. EWOFFSET).
- Setting Early Warning Interrupt Enable bit in the Interrupt Enable Set register (INTENSET.EW).

**Window Mode**

- Defining Time-Out Period bits in the Configuration register (CONFIG.PER).
- Defining Window Mode Time-Out Period bits in the Configuration register (CONFIG.WINDOW).
- Setting Window Enable bit in the Control register (CTRL.WEN).

**Window Mode with Early Warning interrupt**

- Defining Time-Out Period bits in the Configuration register (CONFIG.PER).
- Defining Window Mode Time-Out Period bits in the Configuration register (CONFIG.WINDOW).

- Setting Window Enable bit in the Control register (CTRL.WEN).
- Defining Early Warning Interrupt Time Offset bits in the Early Warning Interrupt Control register (EWCTRL. EWOFFSET).
- Setting Early Warning Interrupt Enable bit in the Interrupt Enable Set register (INTENSET.EW).

#### 18.6.2.2 Configurable Reset Values

After a Power-on Reset, some registers will be loaded with initial values from the NVM User Row. Refer to *NVM User Row Mapping* for more details.

This encompasses the following bits and bit groups:

- Enable bit in the Control register, CTRL.ENABLE
- Always-On bit in the Control register, CTRL.ALWAYSON
- Watchdog Timer Windows Mode Enable bit in the Control register, CTRL.WEN
- Watchdog Timer Windows Mode Time-Out Period bits in the Configuration register, CONFIG.WINDOW
- Time-Out Period in the Configuration register, CONFIG.PER
- Early Warning Interrupt Time Offset bits in the Early Warning Interrupt Control register, EWCTRL.EWOFFSET

For more information about fuse locations, see *NVM User Row Mapping*.

**Related Links**

NVM User Row Mapping

#### 18.6.2.3 Enabling and Disabling

The WDT is enabled by writing a '1' to the Enable bit in the Control register (CTRL.ENABLE). The WDT is disabled by writing a '0' to CTRL.ENABLE.

The WDT can be disabled only if the Always-On bit in the Control register (CTRL.ALWAYSON) is '0'.

#### 18.6.2.4 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 register (CTRL.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.

### 18.8.2 Configuration

**Name:** CONFIG
**Offset:** 0x1
**Reset:** N/A - Loaded from NVM User Row at startup
**Property:** Write-Protected, Enable-Protected, Write-Synchronized

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | WINDOW[3:0] | | | | PER[3:0] | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | x | x | x | x | x | x | x | x |

**Bits 7:4 – WINDOW[3:0]: Window Mode Time-Out Period**
In window mode, these bits determine the watchdog closed window period as a number of oscillator cycles.

These bits are loaded from NVM User Row at start-up. Refer to *NVM User Row Mapping* for more details.

| Value | Description |
|---|---|
| 0x0 | 8 clock cycles |
| 0x1 | 16 clock cycles |
| 0x2 | 32 clock cycles |
| 0x3 | 64 clock cycles |
| 0x4 | 128 clock cycles |
| 0x5 | 256 clocks cycles |
| 0x6 | 512 clocks cycles |
| 0x7 | 1024 clock cycles |
| 0x8 | 2048 clock cycles |
| 0x9 | 4096 clock cycles |
| 0xA | 8192 clock cycles |
| 0xB | 16384 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 GCLK_WDT clock cycles. In window mode operation, these bits define the open window period.

These bits are loaded from NVM User Row at start-up. Refer to *NVM User Row Mapping* for more details.

| Value | Description |
|---|---|
| 0x0 | 8 clock cycles |
| 0x1 | 16 clock cycles |
| 0x2 | 32 clock cycles |
| 0x3 | 64 clock cycles |
| 0x4 | 128 clock cycles |
| 0x5 | 256 clocks cycles |
| 0x6 | 512 clocks cycles |
| 0x7 | 1024 clock cycles |
| 0x8 | 2048 clock cycles |
| 0x9 | 4096 clock cycles |

- 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.

**Related Links**

PAC - Peripheral Access Controller

### 25.5.9 Analog Connections

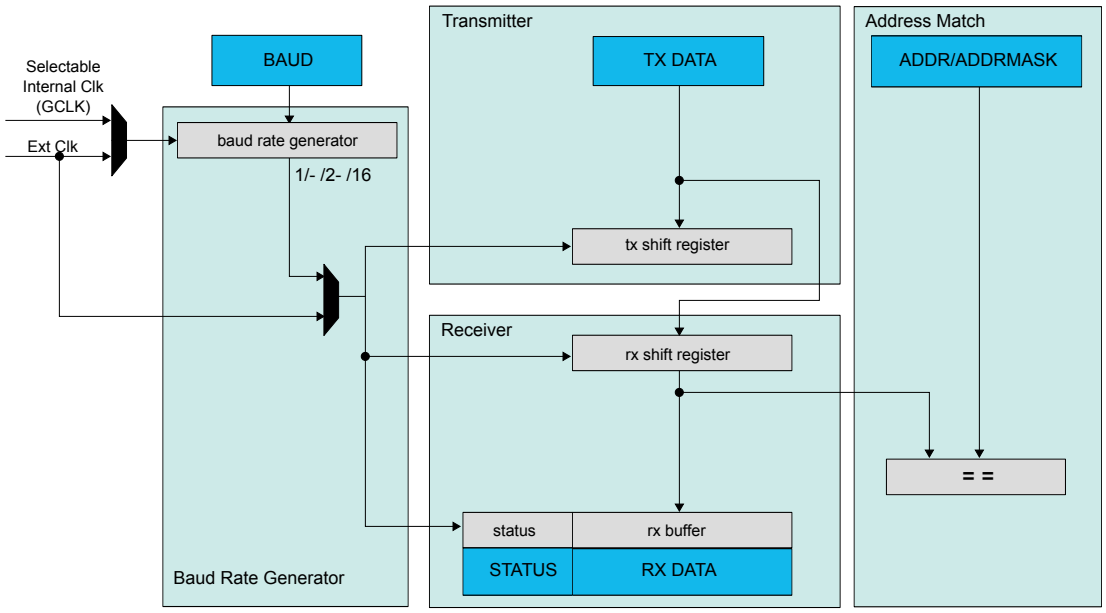
Not applicable.

## 25.6 Functional Description

### 25.6.1 Principle of Operation

The basic structure of the SERCOM serial engine is shown in Figure 25-2. Labels in capital letters are synchronous to the system clock and accessible by the CPU; labels in lowercase letters can be configured to run on the GCLK_SERCOMx_CORE clock or an external clock.

**Figure 25-2. SERCOM Serial Engine**




The transmitter consists of a single write buffer and a shift register.

The receiver consists of a two-level receive buffer and a shift register.

The baud-rate generator is capable of running on the GCLK_SERCOMx_CORE clock or an external clock.

Address matching logic is included for SPI and $I^2C$ operation.

### 25.6.2 Basic Operation

#### 25.6.2.1 Initialization

The SERCOM must be configured to the desired mode by writing the Operating Mode bits in the Control A register (CTRLA.MODE). Refer to table SERCOM Modes for details.

**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 |

### 26.8.3 Baud

**Name:** BAUD
**Offset:** 0x0C
**Reset:** 0x0000
**Property:** Enable-Protected, PAC Write-Protection

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | BAUD[15:8] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

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

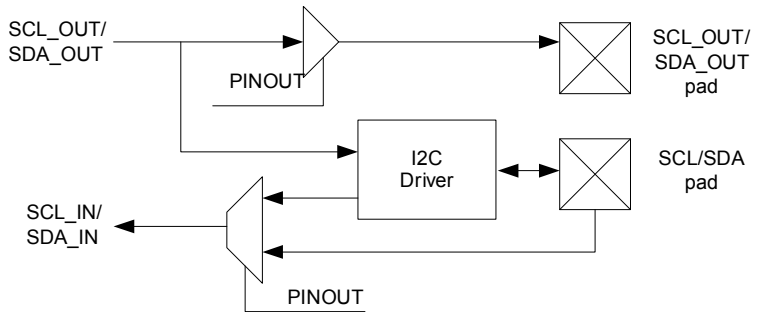
**Bits 15:0 – BAUD[15:0]: Baud Value**
Arithmetic Baud Rate Generation (`CTRLA.SAMPR[0]=0`):

These bits control the clock generation, as described in the SERCOM Baud Rate section.

If Fractional Baud Rate Generation (`CTRLA.SAMPR[0]=1`) bit positions 15 to 13 are replaced by FP[2:0] Fractional Part:

- **Bits 15:13 - FP[2:0]: Fractional Part**

**Figure 28-13. I²C Pad Interface**




#### 28.6.3.4 Quick Command

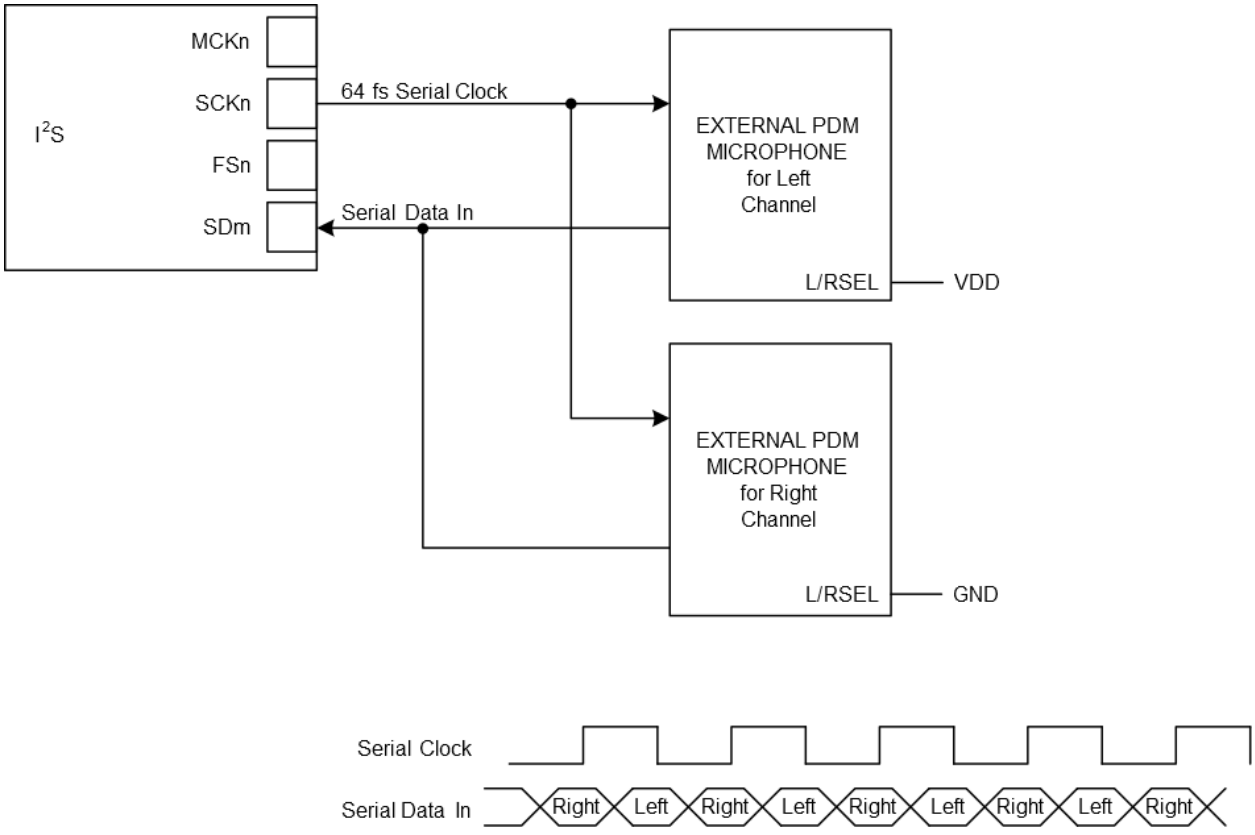
Setting the Quick Command Enable bit in the Control B register (CTRLB.QCEN) enables quick command. When quick command is enabled, the corresponding interrupt flag (INTFLAG.SB or INTFLAG.MB) is set immediately after the slave acknowledges the address. At this point, the software can either issue a stop command or a repeated start by writing CTRLB.CMD or ADDR.ADDR.

### 28.6.4 DMA, Interrupts and Events

**Table 28-1. Module Request for SERCOM I²C Slave**

| Condition | Request | | |
|---|---|---|---|
| | DMA | Interrupt | Event |
| Data needed for transmit (TX) (Slave transmit mode) | Yes (request cleared when data is written) | | NA |
| Data received (RX) (Slave receive mode) | Yes (request cleared when data is read) | | |
| Data Ready (DRDY) | | Yes | |
| Address Match (AMATCH) | | Yes | |
| Stop received (PREC) | | Yes | |
| Error (ERROR) | | Yes | |

**Figure 29-10. PDM Microphones Application Block Diagram**



## 29.8 Register Summary

| Offset | Name | Bit Pos. | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x00 | CTRLA | 7:0 | | | SEREN1 | SEREN0 | CKEN1 | CKEN0 | ENABLE | SWRST |
| 0x01<br>...<br>0x03 | Reserved | | | | | | | | | |
| 0x04 | CLKCTRLn0 | 7:0 | BITDELAY | FSWIDTH[1:0] | | NBSLOTS[2:0] | | | SLOTSIZE[1:0] | |
| 0x05 | | 15:8 | | | | SCKSEL | FSINV | | | FSSEL |
| 0x06 | | 23:16 | MCKDIV[4:0] | | | | | MCKEN | | MCKSEL |
| 0x07 | | 31:24 | MCKOUTINV | SCKOUTINV | FSOUTINV | MCKOUTDIV[4:0] | | | | |
| 0x08 | CLKCTRLn1 | 7:0 | BITDELAY | FSWIDTH[1:0] | | NBSLOTS[2:0] | | | SLOTSIZE[1:0] | |
| 0x09 | | 15:8 | | | | SCKSEL | FSINV | | | FSSEL |
| 0x0A | | 23:16 | MCKDIV[4:0] | | | | | MCKEN | | MCKSEL |
| 0x0B | | 31:24 | MCKOUTINV | SCKOUTINV | FSOUTINV | MCKOUTDIV[4:0] | | | | |
| 0x0C | INTENCLR | 7:0 | | | RXOR1 | RXOR0 | | | RXRDY1 | RXRDY0 |
| 0x0D | | 15:8 | | | TXUR1 | TXUR0 | | | TXRDY1 | TXRDY0 |
| 0x0E<br>...<br>0x0F | Reserved | | | | | | | | | |
| 0x10 | INTENSET | 7:0 | | | RXOR1 | RXOR0 | | | RXRDY1 | RXRDY0 |
| 0x11 | | 15:8 | | | TXUR1 | TXUR0 | | | TXRDY1 | TXRDY0 |
| 0x12<br>...<br>0x13 | Reserved | | | | | | | | | |
| 0x14 | INTFLAG | 7:0 | | | RXOR1 | RXOR0 | | | RXRDY1 | RXRDY0 |
| 0x15 | | 15:8 | | | TXUR1 | TXUR0 | | | TXRDY1 | TXRDY0 |
| 0x16<br>...<br>0x17 | Reserved | | | | | | | | | |
| 0x18 | SYNCBUSY | 7:0 | | | SEREN1 | SEREN0 | CKEN1 | CKEN0 | ENABLE | SWRST |
| 0x19 | | 15:8 | | | | | | | DATA1 | DATA0 |
| 0x1A<br>...<br>0x1F | Reserved | | | | | | | | | |
| 0x20 | SERCTRLn0 | 7:0 | SLOTADJ | | CLKSEL | TXSAME | TXDEFAULT[1:0] | | SERMODE[1:0] | |
| 0x21 | | 15:8 | BITREV | EXTEND[1:0] | | WORDADJ | | DATASIZE[2:0] | | |
| 0x22 | | 23:16 | SLOTDIS8 | SLOTDIS7 | SLOTDIS6 | SLOTDIS5 | SLOTDIS4 | SLOTDIS3 | SLOTDIS1 | SLOTDIS0 |
| 0x23 | | 31:24 | | | | | | RXLOOP | DMA | MONO |
| 0x24 | SERCTRLn1 | 7:0 | SLOTADJ | | CLKSEL | TXSAME | TXDEFAULT[1:0] | | SERMODE[1:0] | |
| 0x25 | | 15:8 | BITREV | EXTEND[1:0] | | WORDADJ | | DATASIZE[2:0] | | |
| 0x26 | | 23:16 | SLOTDIS8 | SLOTDIS7 | SLOTDIS6 | SLOTDIS5 | SLOTDIS4 | SLOTDIS3 | SLOTDIS1 | SLOTDIS0 |
| 0x27 | | 31:24 | | | | | | RXLOOP | DMA | MONO |
| 0x28<br>...<br>0x2F | Reserved | | | | | | | | | |

# 30. TC – Timer/Counter

## 30.1 Overview

The TC consists of a counter, a prescaler, compare/capture channels and control logic. The counter can be set to count events, or it can be configured to count clock pulses. The counter, together with the compare/capture channels, can be configured to timestamp input events, allowing capture of frequency and pulse width. It can also perform waveform generation, such as frequency generation and pulse-width modulation (PWM).

## 30.2 Features

- Selectable configuration
  - Up to five 16-bit Timer/Counters (TC) including one low-power TC, each configurable as:
    - 8-bit TC with two compare/capture channels
    - 16-bit TC with two compare/capture channels
    - 32-bit TC with two compare/capture channels, by using two TCs
- Waveform generation
  - Frequency generation
  - Single-slope pulse-width modulation
- Input capture
  - Event capture
  - Frequency capture
  - Pulse-width capture
- One input event
- Interrupts/output events on:
  - Counter overflow/underflow
  - Compare match or capture
- Internal prescaler
- Can be used with DMA and to trigger DMA transactions

**Bit 1 – ERR: Error Interrupt Enable**
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Error Interrupt Enable bit, which disables the Error interrupt.

| Value | Description |
|---|---|
| 0 | The Error interrupt is disabled. |
| 1 | The Error interrupt is enabled. |

**Bit 0 – OVF: Overflow Interrupt Enable**
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Overflow Interrupt Enable bit, which disables the Overflow interrupt request.

| Value | Description |
|---|---|
| 0 | The Overflow interrupt is disabled. |
| 1 | The Overflow interrupt is enabled. |

### 30.8.9 Interrupt Enable Set

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).

**Name:** INTENSET
**Offset:** 0x0D
**Reset:** 0x00
**Property:** PAC Write-Protection

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | MC1 | MC0 | SYNCRDY | | ERR | OVF |
| Access | | | R/W | R/W | R/W | | R/W | R/W |
| Reset | | | 0 | 0 | 0 | | 0 | 0 |

**Bits 5,4 – MCx: Match or Capture Channel x Interrupt Enable [x = 1..0]**
Writing a '0' to these bits has no effect.

Writing a '1' to MCx will set the corresponding Match or Capture Channel x Interrupt Enable bit, which enables the Match or Capture Channel x interrupt.

| Value | Description |
|---|---|
| 0 | The Match or Capture Channel x interrupt is disabled. |
| 1 | The Match or Capture Channel x interrupt is enabled. |

**Bit 3 – SYNCRDY: Synchronization Ready Interrupt Enable**
Writing a '0' to this bit has no effect.

Writing a one to this bit will clear the Synchronization Ready Interrupt Disable/Enable bit, which disables the Synchronization Ready interrupt.

| Value | Description |
|---|---|
| 0 | The Synchronization Ready interrupt is disabled. |
| 1 | The Synchronization Ready interrupt is enabled. |

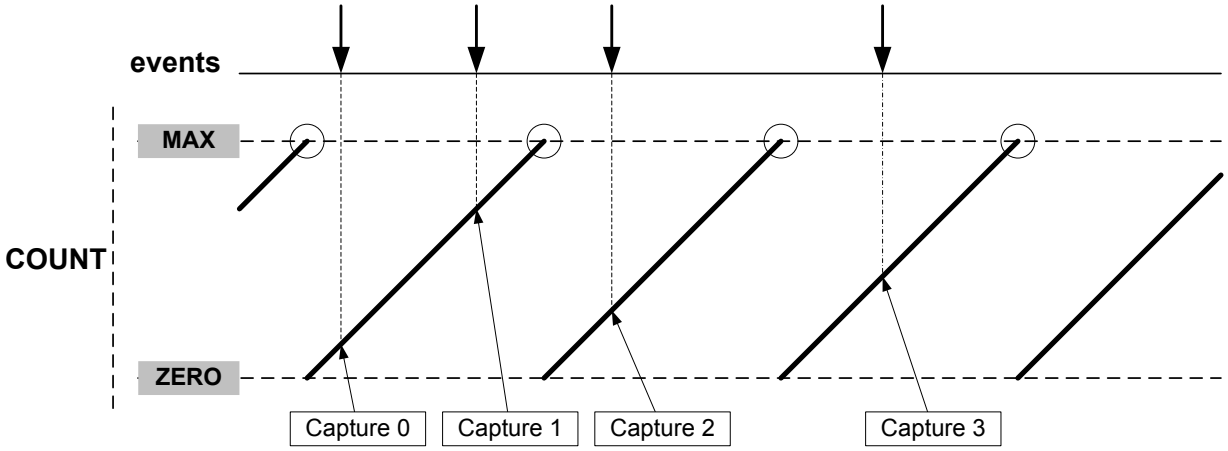
#### 31.6.2.7 Capture Operations

To enable and use capture operations, the Match or Capture Channel x Event Input Enable bit in the Event Control register (EVCTRL.MCEIx) must be written to '1'. The capture channels to be used must also be enabled in the Capture Channel x Enable bit in the Control A register (CTRLA.CPTENx) before capturing can be performed.

**Event Capture Action**

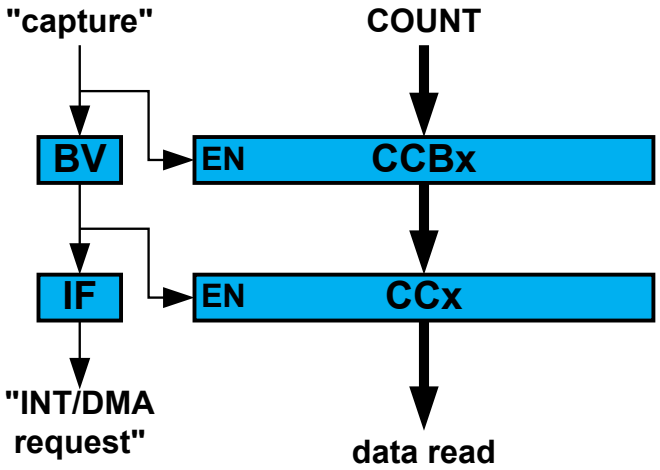
The compare/capture channels can be used as input capture channels to capture events from the Event System, and give them a timestamp. The following figure shows four capture events for one capture channel.

**Figure 31-14. Input Capture Timing**




For input capture, the buffer register and the corresponding CCx act like a FIFO. When CCx is empty or read, any content in CCBx is transferred to CCx. The buffer valid flag is passed to set the CCx interrupt flag (IF) and generate the optional interrupt, event or DMA request. CCBx register value can't be read, all captured data must be read from CCx register.

**Figure 31-15. Capture Double Buffering**




The TCC can detect capture overflow of the input capture channels: When a new capture event is detected while the Capture Buffer Valid flag (STATUS.CCBV) is still set, the new timestamp will not be stored and INTFLAG.ERR will be set.

**Period and Pulse-Width (PPW) Capture Action**

#### 32.6.3.15 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 section STATUS_PIPE register.

If one of the error field in the STATUS_PIPE register from the Host Pipe Descriptor is set and the STATUS_PIPE.ERCNT is less than the CTRL_PIPE.PERMAX, the STATUS_PIPE.ERCNT is incremented.

#### 32.6.3.16 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 EXTREG register.

When the pipe is configured and enabled, an EXTENDED TOKEN followed by a LPM TOKEN are transmitted. The device responds with a valid HANDSHAKE, corrupted HANDSHAKE or no HANDSHAKE (TIME-OUT).

If the valid HANDSHAKE is an ACK, the host will immediately proceed to L1 SLEEP and the PINTFLAG.TRCT0 is set. The minimum duration of the L1 SLEEP state will be the TL1RetryAndResidency as defined in the reference document "ENGINEERING CHANGE NOTICE, USB 2.0 Link Power Management Addendum". When entering the L1 SLEEP state, the CTRLB.SOFE is cleared, avoiding Start-of-Frame generation.

If the valid HANDSHAKE is a NYET PINTFLAG.TRFAIL is set.

If the valid HANDSHAKE is a STALL the PINTFLAG.STALL is set.

If there is no HANDSHAKE or corrupted HANDSHAKE, the EXTENDED/LPM pair of TOKENS will be transmitted again until reaching the maximum number of retries as defined by the CTRL_PIPE.PERMAX in the pipe descriptor.

If the last retry returns no valid HANDSHAKE, the PINTFLAGn.PERR is set, and the STATUS_BK is updated in the pipe descriptor.

All LPM transactions, should they end up with a ACK, a NYET, a STALL or a PERR, will set the PSTATUS.PFREEZE bit, freezing the pipe before a succeeding operation. The user should unfreeze the pipe to start a new LPM transaction.

To exit the L1 STATE, the user initiate a DOWNSTREAM RESUME by setting the bit CTRLB.RESUME or a L1 RESUME by setting the Send L1 Resume bit in CTRLB (CTRLB.L1RESUME). In the case of a L1 RESUME, the K STATE duration is given by the BESL bit field in the EXTREG.VARIABLE field. See EXTREG.

When the host is in the L1 SLEEP state after a successful LPM transmitted, the device can initiate an UPSTREAM RESUME. This will set the Upstream Resume Interrupt bit in INTFLAG (INTFLAG.UPRSM). The host should proceed then to a L1 RESUME as described above.

After resuming from the L1 SLEEP state, the bit CTRLB.SOFE is set, allowing Start-of-Frame generation.

**Errata reference: 13574**
**Fix/Workaround:**
Write CTRLB.ACKACT to 0 using the following sequence:
// If higher priority interrupts exist, then disable so that the
// following two writes are atomic.
SERCOM - STATUS.reg = 0;
SERCOM - CTRLB.reg = 0;
// Re-enable interrupts if applicable.
Write CTRLB.ACKACT to 1 using the following sequence:
// If higher priority interrupts exist, then disable so that the
// following two writes are atomic.
SERCOM - STATUS.reg = 0;
SERCOM - CTRLB.reg = SERCOM_I2CS_CTRLB_ACKACT;
// Re-enable interrupts if applicable.
Otherwise, only write to CTRLB in the AMATCH or DRDY interrupts if it is to close out a transaction.
When not closing a transaction, clear the AMATCH interrupt by writing a 1 to its bit position instead of using CTRLB.CMD. The DRDY interrupt is automatically cleared by reading/writing to the DATA register in smart mode. If not in smart mode, DRDY should be cleared by writing a 1 to its bit position.
Code replacements examples:
Current:
SERCOM - CTRLB.reg |= SERCOM_I2CS_CTRLB_ACKACT;
Change to:
// If higher priority interrupts exist, then disable so that the
// following two writes are atomic.
SERCOM - STATUS.reg = 0;
SERCOM - CTRLB.reg = SERCOM_I2CS_CTRLB_ACKACT;
// Re-enable interrupts if applicable.
Current:
SERCOM - CTRLB.reg &= ~SERCOM_I2CS_CTRLB_ACKACT;
Change to:
// If higher priority interrupts exist, then disable so that the
// following two writes are atomic.
SERCOM - STATUS.reg = 0;
SERCOM - CTRLB.reg = 0;
// Re-enable interrupts if applicable.
Current:
/* ACK or NACK address */
SERCOM - CTRLB.reg |= SERCOM_I2CS_CTRLB_CMD(0x3);
Change to:
// CMD=0x3 clears all interrupts, so to keep the result similar,
// PREC is cleared if it was set.
if (SERCOM - INTFLAG.bit.PREC) SERCOM - INTFLAG.reg = SERCOM_I2CS_INTFLAG_PREC;
SERCOM - INTFLAG.reg = SERCOM_I2CS_INTFLAG_AMATCH;

**15 – The SYSTICK calibration value is incorrect.**
**Errata reference: 14154**
**Fix/Workaround:**

**Fix/Workaround:**
None

**12 – Pulldown functionality is not available on GPIO pin PA24 and PA25**
**Errata reference: 13883**
**Fix/Workaround:**
None

**13 – The voltage regulator in low power mode is not functional at temperatures above 85C.**
**Errata reference: 12291**
**Fix/Workaround:**
Enable normal mode on the voltage regulator in standby sleep mode.
Example code:
// Set the voltage regulator in normal mode configuration in standby sleep mode
SYSCTRL->VREG.bit.RUNSTDBY = 1;

**14 – If the external XOSC32K is broken, neither the external pin RST nor the GCLK software reset can reset the GCLK generators using XOSC32K as source clock.**
**Errata reference: 12164**
**Fix/Workaround:**
Do a power cycle to reset the GCLK generators after an external XOSC32K failure.

#### 40.1.3.2 DSU

**1 – If a debugger has issued a DSU Cold-Plugging procedure and then released the CPU from the resulting ""CPU Reset Extension"", the CPU will be held in ""CPU Reset Extension"" after any upcoming reset event.**
**Errata reference: 12015**
**Fix/workaround:**
The CPU must be released from the ""CPU Reset Extension"" either by writing a one in the DSU STATUSA.CRSTEXT register or by applying an external reset with SWCLK high or by power cycling the device.

**2 – The MBIST ""Pause-on-Error"" feature is not functional on this device.**
**Errata reference: 14324**
**Fix/Workaround:**
Do not use the ""Pause-on-Error"" feature.

#### 40.1.3.3 PM

**1 – In debug mode, if a watchdog reset occurs, the debug session is lost.**
**Errata reference: 12196**
**Fix/Workaround:**
A new debug session must be restart after a watchdog reset.

#### 40.1.3.4 DFLL48M

**1 – The DFLL clock must be requested before being configured otherwise a write access to a DFLL register can freeze the device.**
**Errata reference: 9905**