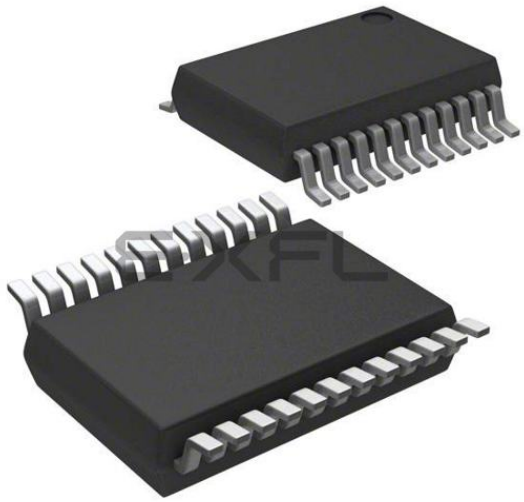


**Welcome to E-XFL.COM**

## 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®-M23 |
| Core Size | 32-Bit Single-Core |
| Speed | 32MHz |
| Connectivity | I²C, LINbus, SPI, UART/USART |
| Peripherals | Brown-out Detect/Reset, DMA, POR, PWM, WDT |
| Number of I/O | 17 |
| Program Memory Size | 16KB (16K x 8) |
| Program Memory Type | FLASH |
| EEPROM Size | 2K x 8 |
| RAM Size | 4K x 8 |
| Voltage - Supply (Vcc/Vdd) | 1.62V ~ 3.63V |
| Data Converters | A/D 5x12b; D/A 1x10b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 125°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 24-SSOP (0.209", 5.30mm Width) |
| Supplier Device Package | 24-SSOP |
| Purchase URL | https://www.e-xfl.com/product-detail/microchip-technology/atsaml10d14a-yft |

**Bit 5 – OSC32KCTRL** Interrupt Flag for OSC32KCTRL

**Bit 4 – OSCCTRL** Interrupt Flag for OSCCTRL

**Bit 3 – RSTC** Interrupt Flag for RSTC

**Bit 2 – MCLK** Interrupt Flag for MCLK

**Bit 1 – PM** Interrupt Flag for PM

**Bit 0 – PAC** Interrupt Flag for PAC

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).

## 17.3 Register Synchronization

### 17.3.1 Overview

All 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.

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).
**Note:** For registers requiring both read- and write-synchronization, the corresponding bit in SYNCBUSY is shared.

| index(m) | Name | Description |
|---|---|---|
| 9 | GCLK_EVSYS_CHANNEL_3 | EVSYS_CHANNEL_3 |
| 10 | GCLK_SERCOM[0,1,2]_SLOW | SERCOM[0,1,2]_SLOW |
| 11 | GCLK_SERCOM0_CORE | SERCOM0_CORE |
| 12 | GCLK_SERCOM1_CORE | SERCOM1_CORE |
| 13 | GCLK_SERCOM2_CORE | SERCOM2_CORE |
| 14 | GCLK_TC0, GCLK_TC1 | TC0,TC1 |
| 15 | GCLK_TC2 | TC2 |
| 16 | GCLK_ADC | ADC |
| 17 | GCLK_AC | AC |
| 18 | GCLK_DAC | DAC |
| 19 | GCLK_PTC | PTC |
| 20 | GCLK_CCL | CCL |

### 19.5.7 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.

### 19.5.8 Register Access Protection

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

- Interrupt Flag 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.

**Related Links**

15. PAC - Peripheral Access Controller

### 19.5.9 SAM L11 TrustZone Specific Register Access Protection

On **SAM L11** 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 *Peripherals Security Attribution* for more information.

### 19.5.10 Analog Connections

Not applicable.

## 19.6 Functional Description

### 19.6.1 Principle of Operation

The CLK_MAIN clock signal from the GCLK module or the DFLLULP is the source for the main clock, which in turn is the common root for the synchronous clocks for the CPU, APBx, and AHBx modules. The CLK_MAIN is divided by an 8-bit prescaler. 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) can be changed on the fly to respond to variable load in the application. 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.

### 19.6.2 Basic Operation

#### 19.6.2.1 Initialization

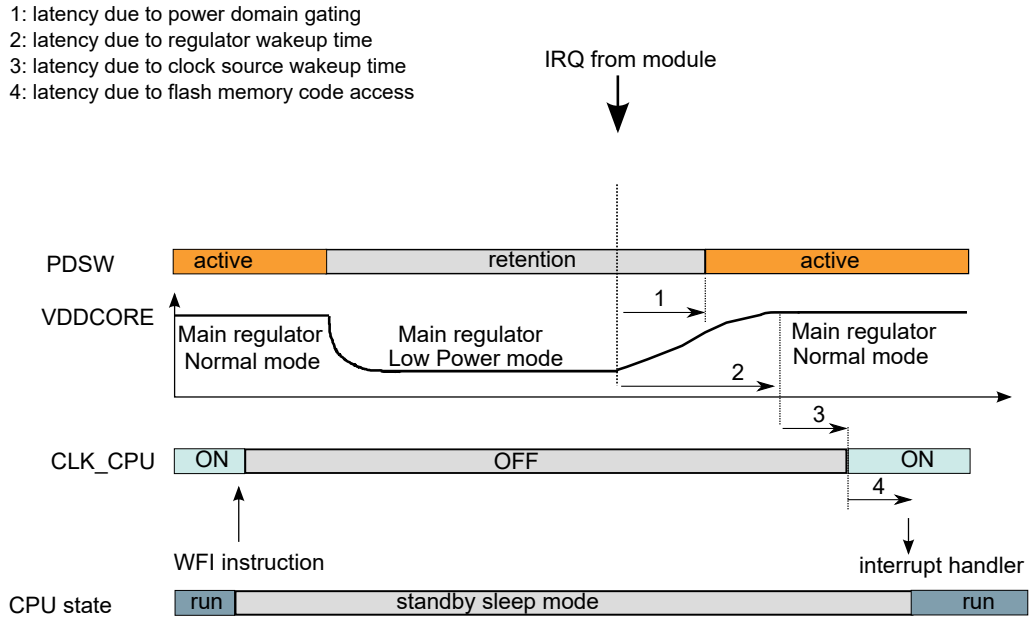
After a Reset, the default clock source of the CLK_MAIN clock (GCLK_MAIN) is started and calibrated before the CPU starts running. The GCLK_MAIN clock is selected as the main clock without any prescaler division.

By default, only the necessary clocks are enabled.

amount of time for the main regulator to transition to the voltage level corresponding to PL2, causing additional wake-up time.

- Latency due to the CPU clock source wake-up time.
- Latency due to the NVM memory access.
- Latency due to Switchable Power Domain back-bias wake-up time:
  If back-bias is enabled, and the device wakes up from retention, it takes a certain amount of time for the regulator to settle.

**Figure 22-5. Total Wake-up Time from Standby Sleep Mode**




### 22.6.5 Standby with Static Power Domain Gating in Details

In Standby Sleep mode, the switchable power domain (PDSW) of a peripheral can remain in active state to perform the peripheral's tasks. This Static Power Domain Gating feature is supported by all peripherals. For some peripherals it must be enabled by writing a Run in Standby bit in the respective Control A register (CTRLA.RUNSTDBY) to '1'. Refer to each peripheral chapter for details.

The following examples illustrate Standby with static Power Domain Gating:

**TC0 Standby with Static Power Domain Gating**

TC0 peripheral is used in counter operation mode. An interrupt is generated to wake-up the device based on the TC0 peripheral configuration. To make the TC0 peripheral continue to run in Standby Sleep mode, the RUNSTDBY bit is written to '1'.

- Entering Standby mode: As shown in Figure 22-6, PDSW remains active. Refer to 22.6.3.5 Power Domain Controller for details.
- Exiting Standby mode: When conditions are met, the TC0 peripheral generates an interrupt to wake-up the device, and the CPU is able to operate normally and execute the TC0 interrupt handler accordingly.
- Wake-up time:
  - The required time to set PDSW to active state has to be considered for the global wake-up time, refer to 22.6.4.6 Wake-Up Time for details.

### 22.8.1 Sleep Configuration

**Name:** SLEEPCFG
**Offset:** 0x01
**Reset:** 0x2
**Property:** PAC Write-Protection

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

**Bits 2:0 – SLEEPMODE[2:0]** Sleep Mode
**Note:** A small latency happens between the store instruction and actual writing of the SLEEPCFG register due to bridges. Software has to make sure the SLEEPCFG register reads the wanted value before issuing WFI instruction.

| Value | Name | Definition |
|---|---|---|
| 0x0 | Reserved | Reserved |
| 0x1 | Reserved | Reserved |
| 0x2 | IDLE | CPU, AHBx, and APBx clocks are OFF |
| 0x3 | Reserved | Reserved |
| 0x4 | STANDBY | ALL clocks are OFF, unless requested by sleepwalking peripheral |
| 0x5 | Reserved | Reserved |
| 0x6 | OFF | All power domains are powered OFF |
| 0x7 | Reserved | Reserved |

### 27.8.13 Tamper Control

**Name:** TAMPCTRL
**Offset:** 0x60
**Reset:** 0x00000000
**Property:** PAC Write-Protection, Enable-Protected

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | DEBNC3 | DEBNC2 | DEBNC1 | DEBNC0 |
| Access | | | | | | | | |
| Reset | | | | | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | | | TAMLVL3 | TAMLVL2 | TAMLVL1 | TAMLVL0 |
| Access | | | | | | | | |
| Reset | | | | | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | IN3ACT[1:0] | | IN2ACT[1:0] | | IN1ACT[1:0] | | IN0ACT[1:0] | |
| Access | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 24, 25, 26, 27 – DEBNC** Debounce Enable of Tamper Input INn

| Value | Description |
|---|---|
| 0 | Debouncing is disabled for Tamper input INn |
| 1 | Debouncing is enabled for Tamper input INn |

**Bits 16, 17, 18, 19 – TAMLVL** Tamper Level Select of Tamper Input INn

| 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 – INACT** Tamper Channel n Action
These bits determine the action taken by Tamper Channel n.

| Value | Name | Description |
|---|---|---|
| 0x0 | OFF | Off (Disabled) |
| 0x1 | WAKE | Wake and set Tamper flag |
| 0x2 | CAPTURE | Capture timestamp and set Tamper flag |
| 0x3 | ACTL | Compare RTC signal routed between INn and OUT pins . When a mismatch occurs, capture timestamp and set Tamper flag |

| Offset | Name | Bit Pos. | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | 31:24 | GP[31:24] | | | | | | | |
| 0x44 | GP1 | 7:0 | GP[7:0] | | | | | | | |
| | | 15:8 | GP[15:8] | | | | | | | |
| | | 23:16 | GP[23:16] | | | | | | | |
| | | 31:24 | GP[31:24] | | | | | | | |
| 0x48 ... 0x5F | Reserved | | | | | | | | | |
| 0x60 | TAMPCTRL | 7:0 | IN3ACT[1:0] | | IN2ACT[1:0] | | IN1ACT[1:0] | | IN0ACT[1:0] | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | TAMLVL3 | TAMLVL2 | TAMLVL1 | TAMLVL0 |
| | | 31:24 | | | | | DEBNC3 | DEBNC2 | DEBNC1 | DEBNC0 |
| 0x64 | TIMESTAMP | 7:0 | COUNT[7:0] | | | | | | | |
| | | 15:8 | COUNT[15:8] | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x68 | TAMPID | 7:0 | | | | | TAMPID3 | TAMPID2 | TAMPID1 | TAMPID0 |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | TAMPEVT | | | | | | | |
| 0x6C | TAMPCTRLB | 7:0 | | | | | ALSI3 | ALSI2 | ALSI1 | ALSI0 |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |

## 27.10 Register Description - Mode 1 - 16-Bit Counter

This Register Description section is valid if the RTC is in COUNT16 mode (CTRLA.MODE=1).

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

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

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

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

On **SAM L11** 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

| Offset | Name | Bit Pos. | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x44 | GP1 | 7:0 | GP[7:0] | | | | | | | |
| | | 15:8 | GP[15:8] | | | | | | | |
| | | 23:16 | GP[23:16] | | | | | | | |
| | | 31:24 | GP[31:24] | | | | | | | |
| 0x48 ... 0x5F | Reserved | | | | | | | | | |
| 0x60 | TAMPCTRL | 7:0 | IN3ACT[1:0] | | IN2ACT[1:0] | | IN1ACT[1:0] | | IN0ACT[1:0] | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | TAMLVL3 | TAMLVL2 | TAMLVL1 | TAMLVL0 |
| | | 31:24 | | | | | DEBNC3 | DEBNC2 | DEBNC1 | DEBNC0 |
| 0x64 | TIMESTAMP | 7:0 | MINUTE[1:0] | | SECOND[5:0] | | | | | |
| | | 15:8 | HOUR[3:0] | | | | MINUTE[5:2] | | | |
| | | 23:16 | MONTH[1:0] | | DAY[4:0] | | | | | HOUR[4:4] |
| | | 31:24 | YEAR[5:0] | | | | | | MONTH[3:2] | |
| 0x68 | TAMPID | 7:0 | | | | | TAMPID3 | TAMPID2 | TAMPID1 | TAMPID0 |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | TAMPEVT | | | | | | | |
| 0x6C | TAMPCTRLB | 7:0 | | | | | ALSI3 | ALSI2 | ALSI1 | ALSI0 |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |

## 27.12 Register Description - Mode 2 - Clock/Calendar

This Register Description section is valid if the RTC is in Clock/Calendar mode (CTRLA.MODE=2).

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

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

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

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

On **SAM L11** 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

| Value | Name | Description |
|---|---|---|
| 0x33 | CHN | DMA channel 19 |
| 0x34 | CHN | DMA channel 20 |
| 0x35 | CHN | DMA channel 21 |
| 0x36 | CHN | DMA channel 22 |
| 0x37 | CHN | DMA channel 23 |
| 0x38 | CHN | DMA channel 24 |
| 0x39 | CHN | DMA channel 25 |
| 0x3A | CHN | DMA channel 26 |
| 0x3B | CHN | DMA channel 27 |
| 0x3C | CHN | DMA channel 28 |
| 0x3D | CHN | DMA channel 29 |
| 0x3E | CHN | DMA channel 30 |
| 0x3F | CHN | DMA channel 31 |

**Bits 3:2 – CRCPOLY[1:0]** CRC Polynomial Type
These bits define the size of the data transfer for each bus access when the CRC is used with I/O interface, as shown in the table below.

| Value | Name | Description |
|---|---|---|
| 0x0 | CRC16 | CRC-16 (CRC-CCITT) |
| 0x1 | CRC32 | CRC32 (IEEE 802.3) |
| 0x2-0x3 | | Reserved |

**Bits 1:0 – CRCBEATSIZE[1:0]** CRC Beat Size
These bits define the size of the data transfer for each bus access when the CRC is used with I/O interface.

| Value | Name | Description |
|---|---|---|
| 0x0 | BYTE | 8-bit bus transfer |
| 0x1 | HWORD | 16-bit bus transfer |
| 0x2 | WORD | 32-bit bus transfer |
| 0x3 | | Reserved |

### 31.8.5 Status

**Name:** STATUS
**Offset:** 0x007
**Reset:** 0x00
**Property:** Read-Only

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | DRP | RAMINV |
| Access | | | | | | | R | R |
| Reset | | | | | | | 0 | 0 |

**Bit 1 – DRP** Data Remanence Prevention Routine
This bit identifies if the data remanence prevention routine is running.

| Value | Description |
|---|---|
| 0 | The data remanence prevention routine is not running. |
| 1 | The data remanence prevention routine is running. |

**Bit 0 – RAMINV** RAM Inversion Bit
This bit identifies if the TrustRAM bit values are inverted.

| Value | Description |
|---|---|
| 0 | The TrustRAM physical bit information is normal. |
| 1 | The TrustRAM physical bit information is inverted. |

### 31.8.10 Security RAM n

**Name:** RAM
**Offset:** 0x0100 + n*0x04 [n=0..63]
**Reset:** 0x00000000
**Property:** PAC Write-Protected, Enable-Protected

Access to the Security RAM is only permitted when CTRLA.ENABLE=1.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | DATA[31:24] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | DATA[23:16] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

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

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

**Bits 31:0 – DATA[31:0]** RAM Data

- Status register (STATUS)
- Data register (DATA)

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

Write-protection does not apply to accesses through an external debugger.

**Related Links**

15. PAC - Peripheral Access Controller

### 36.5.9 SAM L11 TrustZone Specific Register Access Protection

On **SAM L11** 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 *Peripherals Security Attribution* for more information.

### 36.5.10 Analog Connections

Not applicable.

## 36.6 Functional Description

### 36.6.1 Principle of Operation

The SPI is a high-speed synchronous data transfer interface It allows high-speed communication between the device and peripheral devices.

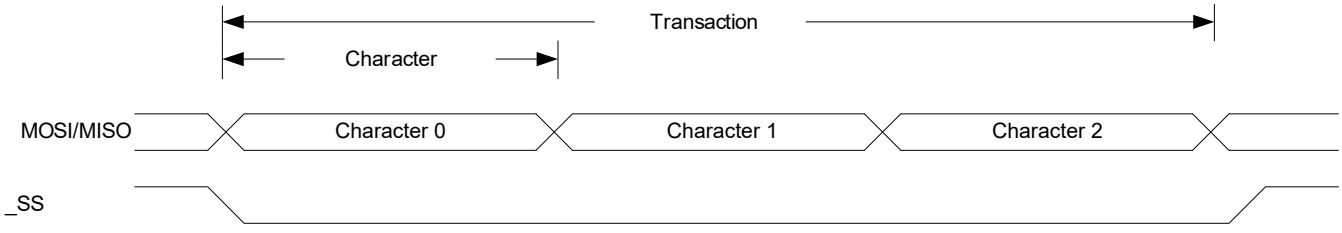
The SPI can operate as master or slave. As master, the SPI initiates and controls all data transactions. The SPI is single buffered for transmitting and double buffered for receiving.

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 or four-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 36-2. SPI Transaction Format**



# 37. SERCOM I²C – SERCOM Inter-Integrated Circuit

## 37.1 Overview

The inter-integrated circuit ( I²C) interface is one of the available modes in the serial communication interface (SERCOM).

The I²C interface uses the SERCOM transmitter and receiver configured as shown in Figure 37-1. Labels in capital letters are registers accessible by the CPU, while lowercase labels are internal to the SERCOM.

A SERCOM instance can be configured to be either an I²C master or an I²C slave. Both master and slave have an interface containing a shift register, a transmit buffer and a receive buffer. In addition, the I²C master uses the SERCOM baud-rate generator, while the I²C slave uses the SERCOM address match logic.

**Related Links**

34. SERCOM – Serial Communication Interface

## 37.2 Features

SERCOM I²C includes the following features:

- Master or slave operation
- Can be used with DMA
- Philips I²C compatible
- SMBus™ compatible
- PMBus compatible
- Support of 100kHz and 400kHz, 1MHz and 3.4MHz I²C mode
- 4-Wire operation supported
- Physical interface includes:
  - Slew-rate limited outputs
  - Filtered inputs
- Slave 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

**Related Links**

34.2 Features

**Note:** The I²C standard *Fm+* (Fast-mode plus) requires a nominal high to low SCL ratio of 1:2, and BAUD should be set accordingly. At a minimum, BAUD.BAUD and/or BAUD.BAUDLOW must be non-zero.

**Startup Timing** The minimum time between SDA transition and SCL rising edge is 6 APB cycles when the DATA register is written in smart mode. If a greater startup time is required due to long rise times, the time between DATA write and IF clear must be controlled by software.
**Note:** When timing is controlled by user, the Smart Mode cannot be enabled.

***Master Clock Generation (High-Speed Mode)***
For I²C *Hs* transfers, there is no SCL synchronization. Instead, the SCL frequency is determined by the GCLK_SERCOMx_CORE frequency ($f_{GCLK}$) and the High-Speed Baud setting in the Baud register (BAUD.HSBAUD). When BAUD.HSBAUDLOW=0, the HSBAUD value will determine both SCL high and SCL low. In this case the following formula determines the SCL frequency.

$$f_{SCL} = \frac{f_{GCLK}}{2 + 2 \cdot HS\ BAUD}$$

When HSBAUDLOW is non-zero, the following formula determines the SCL frequency.

$$f_{SCL} = \frac{f_{GCLK}}{2 + HS\ BAUD + HSBAUDLOW}$$

**Note:** The I²C standard *Hs* (High-speed) requires a nominal high to low SCL ratio of 1:2, and HSBAUD should be set accordingly. At a minimum, BAUD.HSBAUD and/or BAUD.HSBAUDLOW must be non-zero.

#### 37.6.2.4.2 Transmitting Address Packets

The I²C master starts a bus transaction by writing the I²C slave address to ADDR.ADDR and the direction bit, as described in 37.6.1 Principle of Operation. If the bus is busy, the I²C master will wait until the bus becomes idle before continuing the operation. When the bus is idle, the I²C master will issue a start condition on the bus. The I²C master will then transmit an address packet using the address written to ADDR.ADDR. After the address packet has been transmitted by the I²C master, 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 Master on Bus bit in the Interrupt Flag Status and Clear register (INTFLAG.MB) and the Arbitration Lost bit in the Status register (STATUS.ARBLOST) are both set. Serial data output to SDA is disabled, and the SCL is released, which disables clock stretching. In effect the I²C master 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 Master Bus Error bit in the Status register (STATUS.BUSERR) are both set in addition to STATUS.ARBLOST.

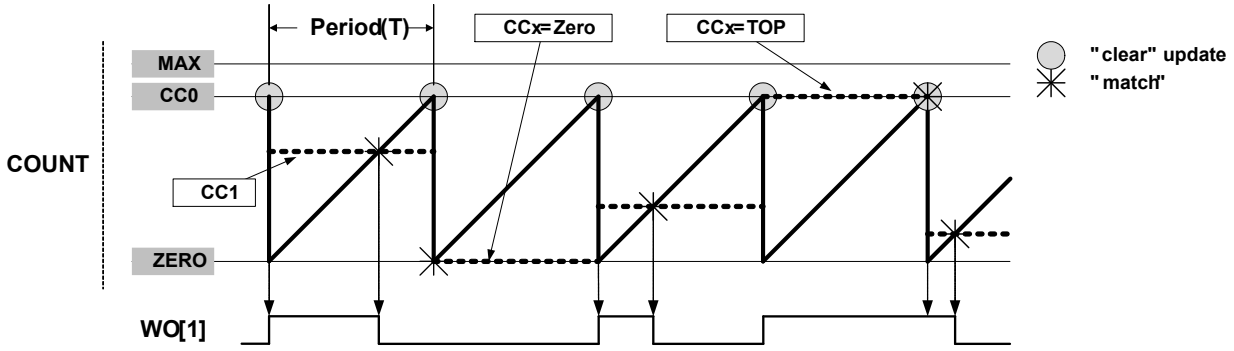
The Master Received Not Acknowledge bit in the Status register (STATUS.RXNACK) will always contain the last successfully received acknowledge or not acknowledge indication.

In this case, software will typically inform the application code of the condition and then clear the interrupt flag before exiting the interrupt routine. No other flags have to be cleared at this moment, because all flags will be cleared automatically the next time the ADDR.ADDR register is written.

**Case 2: Address packet transmit complete – No ACK received**

If there is no I²C slave 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.

**Figure 38-6. Match PWM Operation**




The table below shows the update counter and overflow event/interrupt generation conditions in different operation modes.

**Table 38-3. Counter Update and Overflow Event/interrupt Conditions in TC**

| Name | Operation | TOP | Update | Output Waveform | | OVFIF/Event | |
|---|---|---|---|---|---|---|---|
| | | | | On Match | On Update | Up | Down |
| NFRQ | Normal Frequency | PER | TOP/ ZERO | Toggle | Stable | TOP | ZERO |
| MFRQ | Match Frequency | CC0 | TOP/ ZERO | Toggle | Stable | TOP | ZERO |
| NPWM | Single-slope PWM | PER | TOP/ ZERO | See description above. | | TOP | ZERO |
| MPWM | Single-slope PWM | CC0 | TOP/ ZERO | Toggle | Toggle | TOP | ZERO |

**Related Links**

32. PORT - I/O Pin Controller

#### 38.6.2.7 Double Buffering

The Compare Channels (CCx) registers, and the Period (PER) register in 8-bit mode are double buffered. Each buffer register has a buffer valid bit (CCBUFVx or PERBUFV) in the STATUS register, which indicates that the buffer register contains a new valid value that can be copied into the corresponding register. As long as the respective buffer valid status flag (PERBUFV or CCBUFVx) are set to '1', related syncbusy bits are set (SYNCBUSY.PER or SYNCBUSY.CCx), a write to the respective PER/PERBUF or CCx/CCBUFx registers will generate a PAC error, and access to the respective PER or CCx register is invalid.

When the buffer valid flag bit in the STATUS register is '1' and the Lock Update bit in the CTRLB register is set to '0', (writing CTRLBCLR.LUPD to '1'), double buffering is enabled: the data from buffer registers will be copied into the corresponding register under hardware UPDATE conditions, then the buffer valid flags bit in the STATUS register are automatically cleared by hardware.
**Note:** The software update command (CTRLBSET.CMD=0x3) is acting independently of the LUPD value.

A compare register is double buffered as in the following figure.

## 46.7 Power Consumption

The values in this section are measured values of power consumption under the following conditions, except where noted:

- **Operating Conditions**
  - $V_{DDIO}$ = 3.3V or 1.8V
  - CPU is running on Flash with required Wait states, as recommended in the NVM Characteristics section
  - Low-power cache is enabled
  - BOD33 is disabled
  - I/Os are configured with digital input trigger disabled (default Reset configuration)
- **Oscillators**
  - XOSC (crystal oscillator) stopped
  - XOSC32K (32.768 kHz crystal oscillator) running with external 32.768 kHz crystal
  - When in Active mode with Performance Level 2 (PL2), DPLL is running at 32 MHz and using XOSC32K as reference
  - When in Active mode on DFLLULP, the DFLLULP is configured in Closed Loop mode using XOSC32K as reference clock and MCLK.CTRLA.CKSEL = 1

**Table 46-8. Active Current Consumption**

| Mode | Conditions | Regulator | PL | CPU Clock | Vcc | Ta | Typ. | Max | Units |
|---|---|---|---|---|---|---|---|---|---|
| ACTIVE | COREMARK/ FIBONACCI | LDO | PL0 | DFLLULP at 8MHz | 1.8V | Max at 85°C Typ at 25° | 64.1 | 82 | µA/Mhz |
| | | | | | 3.3V | | 64.4 | 84 | |
| | | | | OSC 8MHz | 1.8V | | 66.6 | 81 | |
| | | | | | 3.3V | | 70.3 | 83 | |
| | | | | OSC 4MHz | 1.8V | | 74.1 | 102 | |
| | | | | | 3.3V | | 77.8 | 106 | |
| | | | PL2 | FDPLL96M at 32MHz | 1.8V | | 82.0 | 89 | |
| | | | | | 3.3V | | 82.5 | 89 | |
| | | | | DFLLULP at 32MHz | 1.8V | | 75.8 | 99 | |
| | | | | | 3.3V | | 75.8 | 96 | |
| | | BUCK | PL0 | DFLLULP at 8MHz | 1.8V | | 40.0 | 53 | |
| | | | | | 3.3V | | 25.3 | 34 | |
| | | | | OSC 8MHz | 1.8V | | 43.8 | 53 | |
| | | | | | 3.3V | | 32.1 | 39 | |
| | | | | OSC 4MHz | 1.8V | | 50.3 | 68 | |
| | | | | | 3.3V | | 38.9 | 52 | |
| | | | PL2 | FDPLL96M at 32MHz | 1.8V | | 59.9 | 66 | |

| Symbol | Parameter | Conditions | | Min. | Typ. | Max. | Units |
|---|---|---|---|---|---|---|---|
| | | Master, VDD>1,62V | | 0 | - | - | |
| tMOV | MOSI output valid after SCK | Master, VDD>2,70V | | - | - | 34.5 | ns |
| | | Master, VDD>1,62V | | - | - | 38.6 | |
| tMOH | MOSI hold after SCK | Master, VDD>2,70V | | 9.7 | - | - | |
| | | Master, VDD>1,62V | | 9.7 | - | - | |
| tSSCK | Slave SCK Period when tMIS=0 on the master side | Slave | Reception | 2*(tSIS +tMASTER_OUT) (5) | - | - | |
| | | Slave | Transmission | 2*(tSOV +tMASTER_IN) (6) | - | - | |
| tSSCKW | SCK high/low width | Slave | | - | 0,5*tSCK | - | |
| tSSCKR | SCK rise time(2) | Slave | | - | 0,25*tSCK | - | |
| tSSCKF | SCK fall time(2) | Slave | | - | 0,25*tSCK | - | |
| tSIS | MOSI setup to SCK | Slave, VDD>2,70V | | 25.6 | - | - | ns |
| | | Slave, VDD>1,62V | | 26.2 | - | - | |
| tSIH | MOSI hold after SCK | Slave, VDD>2,70V | | 13.2 | - | - | |
| | | Slave, VDD>1,62V | | 13.9 | - | - | |
| tSSS | SS setup to SCK | Slave | PRELOADEN=1 | tSOSS+tEXT_MIS +2*tAPBC (8) (9) | - | - | |
| | | | PRELOADEN=0 | tSOSS+tEXT_MIS (8) | - | - | |
| tSSH | SS hold after SCK | Slave | | 0.5*tSSCK | - | - | |
| tSOV | MISO output valid after SCK | Slave, VDD>2,70V | | - | - | 69 | |
| | | Slave, VDD>1,62V | | - | - | 78.4 | |
| tSOH | MISO hold after SCK | Slave, VDD>2,70V | | 20.2 | - | - | |
| | | Slave, VDD>1,62V | | 20.2 | - | - | |
| tSOSS | MISO setup after SS low | Slave, VDD>2,70V | | - | - | 1* tSCK | |
| | | Slave, VDD>1,62V | | - | - | 1* tSCK | |

## 47.5 Oscillators Characteristics

### 47.5.1 Crystal Oscillator (XOSC) Characteristics

**Table 47-18. Power Consumption**

| Symbol | Parameter | Conditions | | Ta | Min. | Typ. | Max. | Units |
|---|---|---|---|---|---|---|---|---|
| $I_{DD}$ | Current consumption | F=2MHz - CL=20pF XOSC,GAIN=0, VCC=3.3V | AMPGC=OFF | Max 125°C Typ 25°C | - | 66 | 106 | µA |
| | | | AMPGC=ON | | - | 62 | 107 | |
| | | F=4MHz - CL=20pF XOSC,GAIN=1, VCC=3.3V | AMPGC=OFF | | - | 107 | 164 | |
| | | | AMPGC=ON | | - | 70 | 132 | |
| | | F=8MHz - CL=20pF XOSC,GAIN=2, VCC=3.3V | AMPGC=OFF | | - | 200 | 307 | |
| | | | AMPGC=ON | | - | 118 | 180 | |
| | | F=16MHz - CL=20pF XOSC,GAIN=3, VCC=3.3V | AMPGC=OFF | | - | 436 | 630 | |
| | | | AMPGC=ON | | - | 247 | 382 | |
| | | F=32MHz - CL=20pF XOSC,GAIN=4, VCC=3.3V | AMPGC=OFF | | - | 1303 | 2251 | |
| | | | AMPGC=ON | | - | 627 | 1116 | |

### 47.5.2 External 32KHz Crystal Oscillator (XOSC32K) Characteristics

**Table 47-19. Power Consumption**

| Symbol | Parameter | Conditions | Ta | Min. | Typ. | Max. | Units |
|---|---|---|---|---|---|---|---|
| $I_{DD}$ | Current consumption | VCC=3.3V | Max 125°C Typ 25°C | - | 309 | 767 | nA |

### 47.5.3 Ultra Low-Power Internal 32 kHz RC Oscillator (OSCULP32K) Characteristics

**Table 47-20. Ultra Low-Power Internal 32 kHz RC Oscillator Electrical Characteristics**

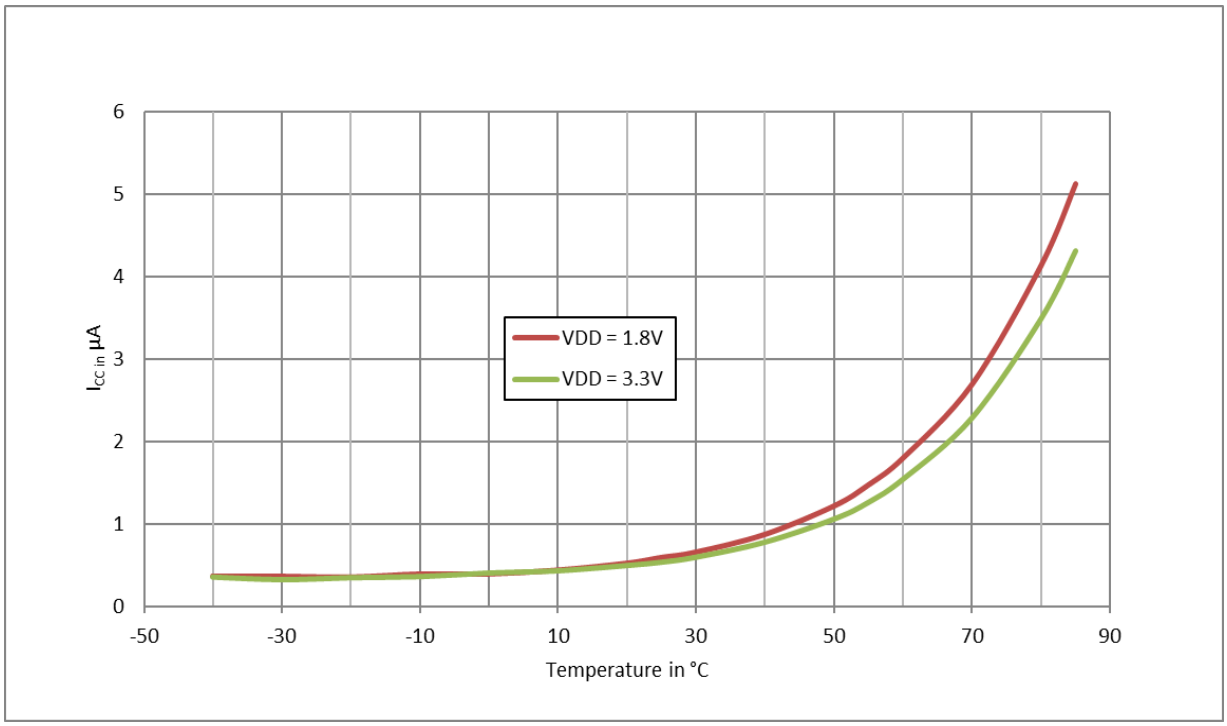
| Symbol | Parameter | Conditions | Min. | Typ. | Max. | Units |
|---|---|---|---|---|---|---|
| $F_{OUT}$ | Output frequency | at 25°C, at $V_{DDIO}$=3.3V | 30.84 | 32.768 | 34.51 | kHz |
| | | at 25°C, over [1.62, 3.63]V | 30.84 | 32.768 | 34.74 | kHz |
| | | over[-40,+125]°C, over [1.62, 3.63]V | 25.17 | 32.768 | 41.76 | kHz |
| Duty | Duty Cycle | | - | 50 | - | % |

### 47.5.4 16 MHz RC Oscillator (OSC16M) Characteristics

**Table 47-21. Multi-RC Oscillator Electrical Characteristics**

| Symbol | Parameter | Conditions | Min. | Typ. | Max. | Units |
|---|---|---|---|---|---|---|
| $F_{OUT}$ | Output frequency | $V_{DD}$=3.3V, T=25°C | 3.96 | 4.00 | 4.04 | MHz |

**Figure 48-2. Power Consumption over Temperature in Standby Sleep Mode with PDSW in Retention state**




**Power Consumption in Off Sleep Mode**

Operating conditions:

- VDDIO = 3.3V or 1.8V

**Figure 48-3. Power Consumption over Temperature in Off Sleep Mode**