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, SCI, SPI, UART/USART, USB |
| Peripherals | DMA, POR, PWM, WDT |
| Number of I/O | 25 |
| Program Memory Size | 128KB (128K x 8) |
| Program Memory Type | FLASH |
| EEPROM Size | - |
| RAM Size | 16K x 8 |
| Voltage - Supply (Vcc/Vdd) | 1.62V ~ 3.63V |
| Data Converters | A/D 10x12b; D/A 2x12b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 105°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 32-VFQFN Exposed Pad |
| Supplier Device Package | 32-VQFN (5x5) |
| Purchase URL | https://www.e-xfl.com/product-detail/microchip-technology/atsaml21e17b-mnt |

### 13.7.1. Write Control

**Name:** WRCTRL
**Offset:** 0x0
**Reset:** 0x00000000
**Property:** –

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | KEY[7:0] | | | | | | | |
| Access | R/W | R/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. The Peripheral Identifier is calculated following formula:

$$PERID = 32 * \mathrm{BridgeNumber} + \mathrm{N}$$

Where BridgeNumber represents the Peripheral Bridge Number (0 for Peripheral Bridge A, 1 for Peripheral Bridge B, etc). N represents the peripheral index from the respective Bridge Number:

### 18.8.5. CPU Clock Division

**Name:** CPUDIV
**Offset:** 0x05
**Reset:** 0x01
**Property:** PAC Write-Protection

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

**Bits 7:0 – CPUDIV[7:0]: CPU Clock Division Factor**
These bits define the division ratio of the main clock prescaler related to the CPU clock domain.

To ensure correct operation, frequencies must be selected so that $F_{CPU} \geq F_{LP}$ (i.e. LPDIV ≥CPUDIV).

Frequencies must never exceed the specified maximum frequency for each clock domain.

| Value | Name | Description |
|---|---|---|
| 0x01 | DIV1 | Divide by 1 |
| 0x02 | DIV2 | Divide by 2 |
| 0x04 | DIV4 | Divide by 4 |
| 0x08 | DIV8 | Divide by 8 |
| 0x10 | DIV16 | Divide by 16 |
| 0x20 | DIV32 | Divide by 32 |
| 0x40 | DIV64 | Divide by 64 |
| 0x80 | DIV128 | Divide by 128 |
| others | - | Reserved |

When the Automatic Power Switch configuration is selected, the Automatic Power Switch Ready bit in the Status register (STATUS.APWSRDY) is set when the Automatic Power Switch is ready to operate. The Automatic Power Switch Ready bit in the Interrupt Flag Status and Clear (INTFLAG.APSWRDY) will be set at the same time.

**Related Links**

Electrical Characteristics on page 1144

**BOD33 Power Switch**

When the Configuration bit field in the Battery Backup Power Switch register (BBPS.CONF) are selecting the BOD33, BOD33 will function as Battery Backup Power Switch. In this case, when the VDD voltage is below the BOD33 threshold, the backup domain supply is switched to VBAT.

**Main Power Supply OK (PSOK) Pin Enable**

The state of the Main Power VDD can be used to switch between supply sources as long as the Battery Backup Power Switch is not configured as Automatic Power Switch (i.e., BBPS.CONF not set to APWS): when the Main Power Supply OK Pin Enable bit in the BBPS register is written to '1' (BBPS.PSOKEN), restoring VDD will form a low-to-high transition on the PSOK pin. This low-to-high transition will switch the Backup Power Supply back to VDD.

**Note:** With BBPS.PSOKEN=0 and BBPS.CONF not configured to APWS, the device can not be restarted.

**Backup Battery Power Switch Status**

The Battery Backup Power Switch bit in the Status register (STATUS.BBPS) indicates whether the backup domain is currently powered by VDD or VBAT.

#### 23.6.3.4. Sleep Mode Operation

The Battery Backup Power Switch is not stopped in any sleep mode.

**Entering Battery Backup Mode**

Entering backup mode can be triggered by either:

- Wait-for-interrupt (WFI) instruction.
- Automatic Power Switch (BBPS.CONF=APWS). When the Automatic Power Switch detects loss of Main Power, the Backup Domain will be powered by battery and the device will enter the backup mode.
- BOD33 detection: When the BOD33 detects loss of Main Power, the Backup Domain will be powered by battery and the device will enter the backup mode. For this trigger, the following register configuration is required: BOD33.ACTION=BKUP, BOD33.VMON=VDD, and BBPS.CONF=BOD33.

**Related Links**

PM – Power Manager on page 192

**Leaving Battery Backup Mode**

Leaving backup mode can be triggered by either:

- RTC requests and externally triggered RSTC requests, under one of these conditions:
  - The Backup Domain is supplied by Main Power, and the Battery Backup Power Switch is *not* forced (BBPS.CONF not set to FORCED)
  - The Battery Backup Power Switch *is* forced (BBPS.CONF is FORCED)

  The device is kept in battery-powered backup mode until Main Power is restored to supply the device. Then, the backup domain will be powered by Main Power.

## 24.7. Register Summary

| Offset | Name | Bit Pos. | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x00 | CTRLA | 7:0 | ALWAYSON | | | | | 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 | 7:0 | | | | CLEAR | ALWAYSON | WEN | ENABLE | |
| 0x09 | | 15:8 | | | | | | | | |
| 0x0A | | 23:16 | | | | | | | | |
| 0x0B | | 31:24 | | | | | | | | |
| 0x0C | CLEAR | 7:0 | CLEAR[7:0] | | | | | | | |

## 24.8. Register Description

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

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). Optional PAC write-protection is denoted by the "PAC Write-Protection" property in each individual register description. For details, refer to Register Access Protection.

Some registers are synchronized when read and/or written. Synchronization is denoted by the "Write-Synchronized" or the "Read-Synchronized" property in each individual register description. For details, refer to Synchronization.

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

**Figure 25-2. RTC Block Diagram (Mode 1 — 16-Bit Counter)**

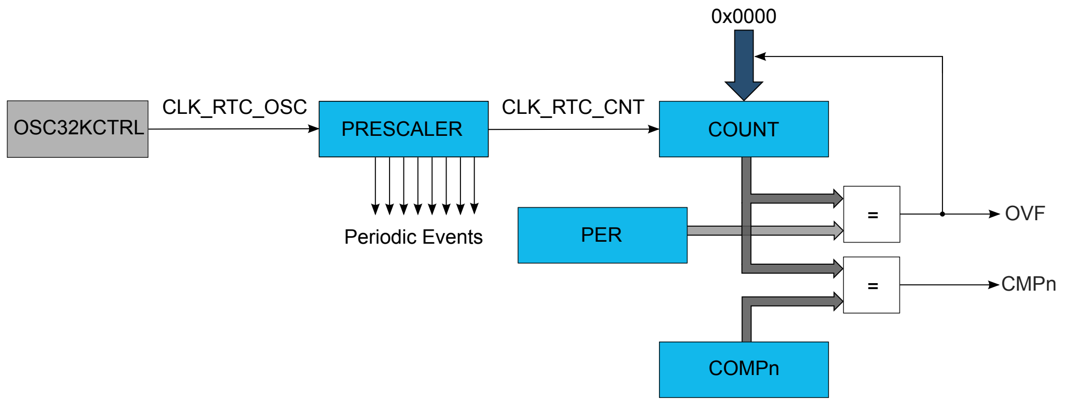
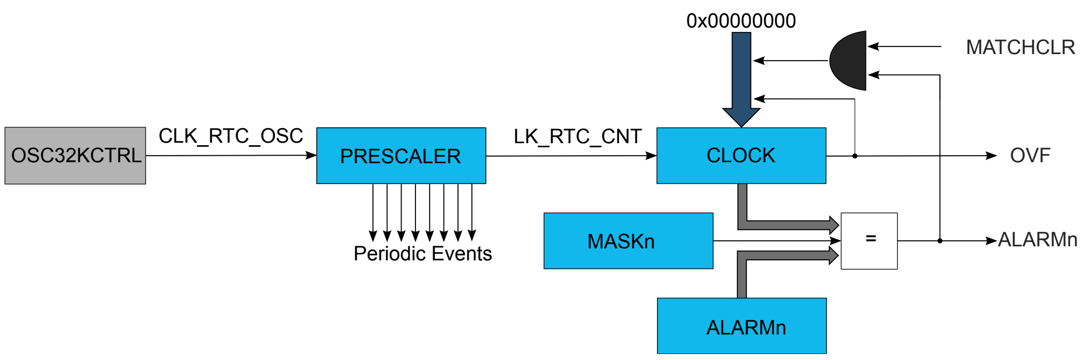



**Figure 25-3. RTC Block Diagram (Mode 2 — Clock/Calendar)**




## 25.4. Signal Description

Not applicable.

**Related Links**

I/O Multiplexing and Considerations on page 30

## 25.5. Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

### 25.5.1. I/O Lines

Not applicable.

### 25.5.2. Power Management

The RTC will continue to operate in any sleep mode where the selected source clock is running. The RTC interrupts can be used to wake up the device from sleep modes. Events connected to the event system can trigger other operations in the system without exiting sleep modes. Refer to the *Power Manager* for details on the different sleep modes.

The RTC will be reset only at power-on (POR) or by setting the Software Reset bit in the Control A register (CTRLA.SWRST=1).

**Related Links**

PM – Power Manager on page 192

### 25.10.4. Interrupt Enable Set in COUNT16 mode (CTRLA.MODE=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.

**Name:** INTENSET
**Offset:** 0x0A
**Reset:** 0x0000
**Property:** PAC Write-Protection

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | OVF | | | | | | CMP1 | CMP0 |
| Access | R/W | | | | | | R/W | R/W |
| Reset | 0 | | | | | | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | PER7 | PER6 | PER5 | PER4 | PER3 | PER2 | PER1 | PER0 |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 15 – OVF: Overflow Interrupt Enable**

Writing a '0' to this bit has no effect. Writing a '1' to this bit will set the Overflow Interrupt Enable bit, which enables the Overflow interrupt.

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

**Bits 9:8 – CMPn: Compare n Interrupt Enable [n = 1..0]**

Writing a '0' to this bit has no effect. Writing a '1' to this bit will set the Compare n Interrupt Enable bit, which and enables the Compare n interrupt.

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

**Bits 7:0 – PERn: Periodic Interval n Interrupt Enable [n = 7..0]**

Writing a '0' to this bit has no effect. Writing a '1' to this bit will set the Periodic Interval n Interrupt Enable bit, which enables the Periodic Interval n interrupt.

| Value | Description |
|---|---|
| 0 | Periodic Interval n interrupt is disabled. |
| 1 | Periodic Interval n interrupt is enabled. |

### 25.10.12. Compare n Value in COUNT16 mode (CTRLA.MODE=1)

**Name:** COMPn
**Offset:** 0x20 + n*0x02 [n=0..1]
**Reset:** 0x0000
**Property:** PAC Write-Protection, Write-Synchronized

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

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

**Bits 15:0 – COMP[15:0]: Compare Value**
The 16-bit value of COMPn is continuously compared with the 16-bit COUNT value. When a match occurs, the Compare n interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.CMPn) is set on the next counter cycle.

The I/O pins of the device are controlled by PORT peripheral registers. Each port pin has a corresponding bit in the Data Direction (DIR) and Data Output Value (OUT) registers to enable that pin as an output and to define the output state.

The direction of each pin in a PORT group is configured by the DIR register. If a bit in DIR is set to '1', the corresponding pin is configured as an output pin. If a bit in DIR is set to '0', the corresponding pin is configured as an input pin.

When the direction is set as output, the corresponding bit in the OUT register will set the level of the pin. If bit y in OUT is written to '1', pin y is driven HIGH. If bit y in OUT is written to '0', pin y is driven LOW. Pin configuration can be set by Pin Configuration (PINCFGy) registers, with y=00, 01, ..31 representing the bit position.

The Data Input Value (IN) is set as the input value of a port pin with resynchronization to the PORT clock. To reduce power consumption, these input synchronizers are clocked only when system requires reading the input value. The value of the pin can always be read, whether the pin is configured as input or output. If the Input Enable bit in the Pin Configuration registers (PINCFGy.INEN) is '0', the input value will not be sampled.

In PORT, the Peripheral Multiplexer Enable bit in the PINCFGy register (PINCFGy.PMUXEN) can be written to '1' to enable the connection between peripheral functions and individual I/O pins. The Peripheral Multiplexing n (PMUXn) registers select the peripheral function for the corresponding pin. This will override the connection between the PORT and that I/O pin, and connect the selected peripheral signal to the particular I/O pin instead of the PORT line bundle.

### 29.6.2. Basic Operation

#### 29.6.2.1. Initialization

After reset, all standard function device I/O pads are connected to the PORT with outputs tri-stated and input buffers disabled, even if there is no clock running.

However, specific pins, such as those used for connection to a debugger, may be configured differently, as required by their special function.

#### 29.6.2.2. Operation

Each I/O pin y can be controlled by the registers in PORT. Each PORT group has its own set of PORT registers, the base address of the register set for pin y is at byte address PORT + ([y] * 0x4). The index within that register set is [y].

To use pin number y as an *output*, write bit y of the DIR register to '1'. This can also be done by writing bit y in the DIRSET register to '1' - this will avoid disturbing the configuration of other pins in that group. The y bit in the OUT register must be written to the desired output value.

Similarly, writing an OUTSET bit to '1' will set the corresponding bit in the OUT register to '1'. Writing a bit in OUTCLR to '1' will set that bit in OUT to zero. Writing a bit in OUTTGL to '1' will toggle that bit in OUT.

To use pin y as an *input*, bit y in the DIR register must be written to '0'. This can also be done by writing bit y in the DIRCLR register to '1' - this will avoid disturbing the configuration of other pins in that group. The input value can be read from bit y in register IN as soon as the INEN bit in the Pin Configuration register (PINCFGy.INEN) is written to '1'.

Refer to *I/O Multiplexing and Considerations* for details on pin configuration and PORT groups.

By default, the input synchronizer is clocked only when an input read is requested. This will delay the read operation by two CLK_PORT cycles. To remove the delay, the input synchronizers for each PORT group of eight pins can be configured to be always active, but this will increase power consumption. This enabled by writing '1' to the corresponding bit in the Interrupt Enable Set register (INTENSET), and disabled by writing '1' to the corresponding bit in the Interrupt Enable Clear register (INTENCLR).

An interrupt request is generated when the interrupt flag is set and if the corresponding interrupt is enabled. The interrupt request remains active until either the interrupt flag is cleared, the interrupt is disabled, or the SPI is reset. For details on clearing interrupt flags, refer to the INTFLAG register description.

The SPI has one common interrupt request line for all the interrupt sources. 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.

**Related Links**

Nested Vector Interrupt Controller on page 52

#### 33.6.4.3. Events

Not applicable.

### 33.6.5. Sleep Mode Operation

The behavior in sleep mode is depending on the master/slave configuration and the Run In Standby bit in the Control A register (CTRLA.RUNSTDBY):

- Master 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.
- Master operation, CTRLA.RUNSTDBY=0: GLK_SERCOMx_CORE will be disabled after the ongoing transaction is finished. Any interrupt can wake up the device.
- Slave operation, CTRLA.RUNSTDBY=1: The Receive Complete interrupt can wake up the device.
- Slave operation, CTRLA.RUNSTDBY=0: All reception will be dropped, including the ongoing transaction.

### 33.6.6. Synchronization

Due to asynchronicity between the main clock domain and the peripheral clock domains, some registers need to be synchronized when written or read.

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)

**Note:** CTRLB.RXEN is write-synchronized somewhat differently. See also CTRLB for details.

Required write-synchronization is denoted by the "Write-Synchronized" property in the register description.

### 34.8.9. Data

**Name:** DATA
**Offset:** 0x28
**Reset:** 0x0000
**Property:** Write-Synchronized, Read-Synchronized

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

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

**Bits 7:0 – DATA[7:0]: Data**
The slave data register I/O location (DATA.DATA) provides access to the master 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 slave (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).

Writing or reading DATA.DATA when not in smart mode does not require synchronization.

| Value | Description |
|---|---|
| 0 | Automatic transfer length disabled. |
| 1 | Automatic transfer length enabled. |

**Bits 10:8 – ADDR[2:0]: 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 $I^2C$ master will await further operation until the bus becomes IDLE.

IDLE: The $I^2C$ master 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.

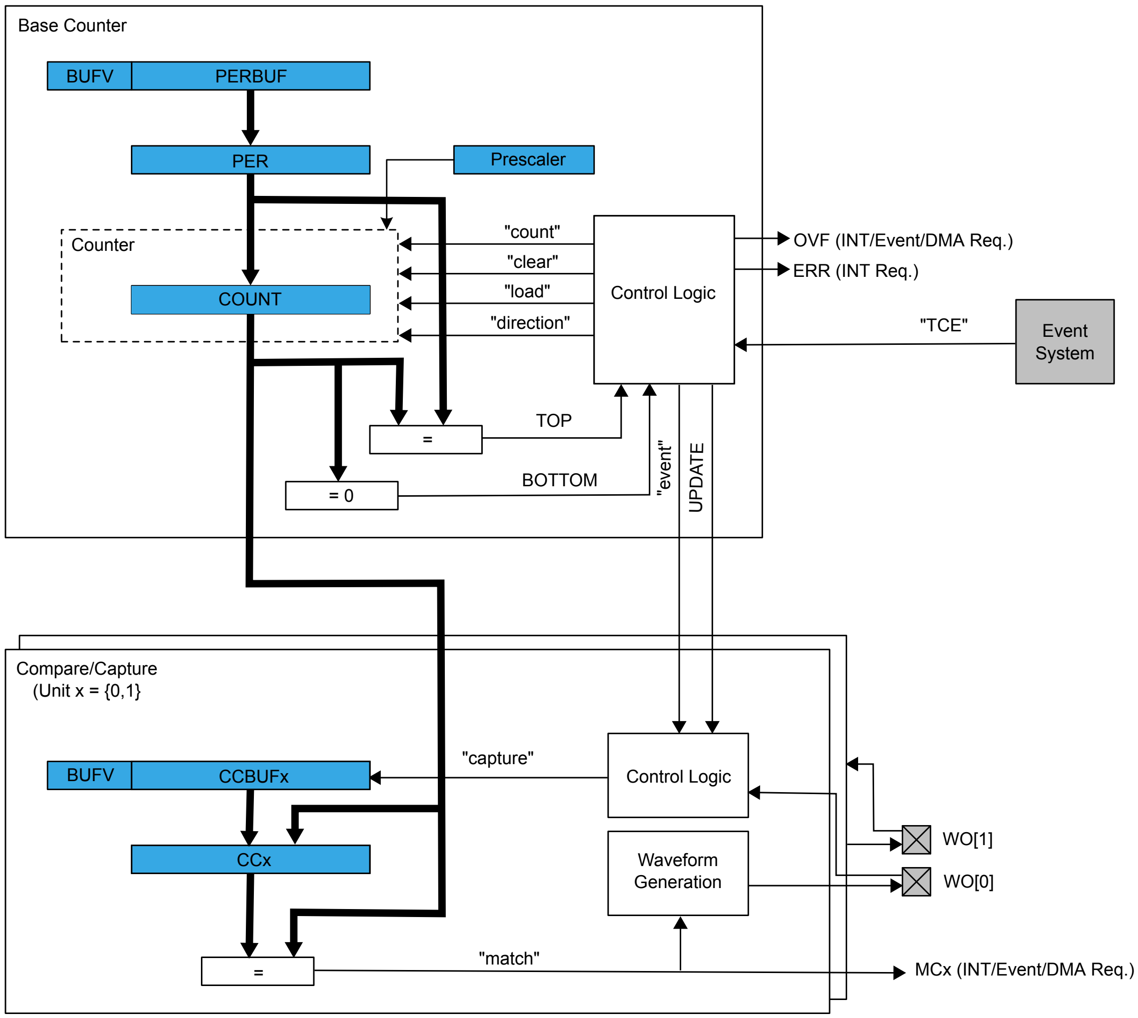
STATUS.BUSERR, STATUS.ARBLOST, INTFLAG.MB and INTFLAG.SB will be cleared when ADDR is written.

The ADDR register can be read at any time without interfering with ongoing bus activity, as a read access does not trigger the master logic to perform any bus protocol related operations.

The $I^2C$ master control logic uses bit 0 of ADDR as the bus protocol's read/write flag (R/W); 0 for write and 1 for read.

## 35.3. Block Diagram

**Figure 35-1. Timer/Counter Block Diagram**




## 35.4. Signal Description

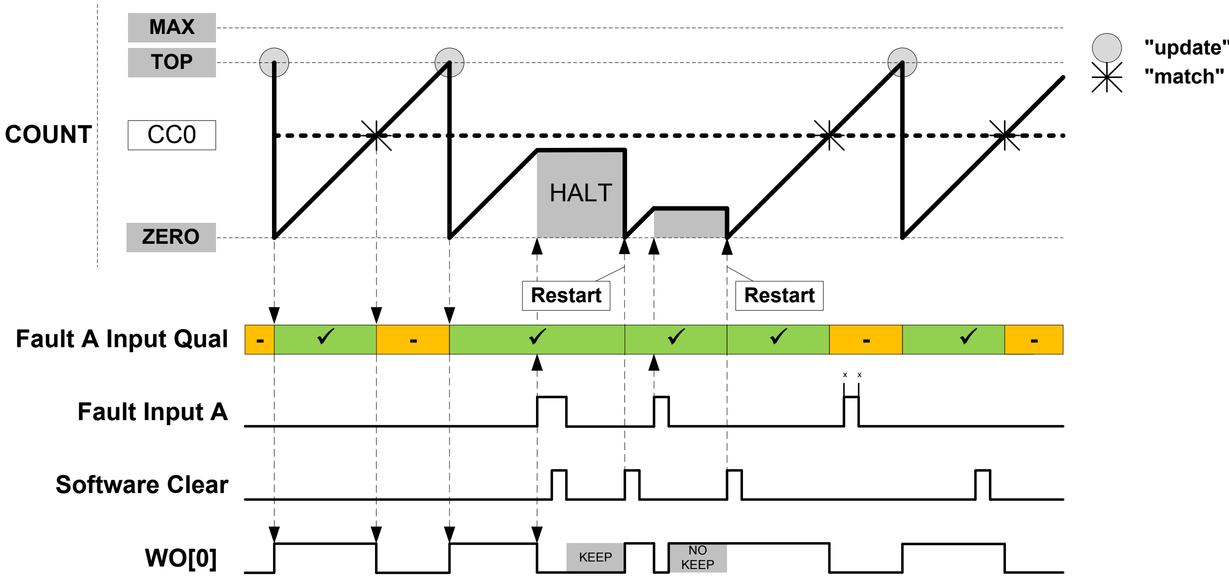
**Table 35-1. Signal Description for TC.**

| Signal Name | Type | Description |
|---|---|---|
| WO[1:0] | Digital output | Waveform output |
| | Digital input | Capture input |

Refer to *I/O Multiplexing and Considerations* for details on the pin mapping for this peripheral. One signal can be mapped on several pins.

**Related Links**

**Figure 36-32. Waveform Generation with Software Halt, Fault Qualification, Keep and Restart Actions**




#### 36.6.3.6. Non-Recoverable Faults

The non-recoverable fault action will force all the compare outputs to a pre-defined level programmed into the Driver Control register (DRVCTRL.NRE and DRVCTRL.NRV). The non-recoverable fault input (EV0 and EV1) actions are enabled in Event Control register (EVCTRL.EVACT0 and EVCTRL.EVACT1).

To avoid false fault detection on external events (e.g. a glitch on an I/O port) a digital filter can be enabled using Non-Recoverable Fault Input x Filter Value bits in the Driver Control register (DRVCTRL.FILTERVALn). Therefore, the event detection is synchronous, and event action is delayed by the selected digital filter value clock cycles.

When the Fault Detection on Debug Break Detection bit in Debug Control register (DGBCTRL.FDDBD) is written to '1', a non-recoverable Debug Faults State and an interrupt (DFS) is generated when the system goes in debug operation.

In RAMP2, RAMP2A, or DSBOTH operation, when the Lock Update bit in the Control B register is set by writing CTRLBSET.LUPD=1 and the ramp index or counter direction changes, a non-recoverable Update Fault State and the respective interrupt (UFS) are generated.

#### 36.6.3.7. Time-Stamp Capture

This feature is enabled when the Capture Time Stamp (STAMP) Event Action in Event Control register (EVCTRL.EVACT) is selected. The counter TOP value must be smaller than MAX.

When a capture event is detected, the COUNT value is copied into the corresponding Channel x Compare/Capture Value (CCx) register. In case of an overflow, the MAX value is copied into the corresponding CCx register.

When a valid captured value is present in the capture channel register, the corresponding Capture Channel x Interrupt Flag (INTFLAG.MCx) is set.

The timer/counter can detect capture overflow of the input capture channels: When a new capture event is detected while the Capture Channel interrupt flag (INTFLAG.MCx) is still set, the new time-stamp will not be stored and INTFLAG.ERR will be set.

**Related Links**

Nested Vector Interrupt Controller on page 52
Sleep Mode Controller on page 198

#### 36.6.4.3. 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: MCx

Writing a '1' ('0') to an Event Output bit in the Event Control Register (EVCTRL.xxEO) enables (disables) the corresponding output event. Refer also to *EVSYS – Event System*.

The TCC can take the following actions on a channel input event (MCx):

- Capture event
- Generate a recoverable or non-recoverable fault

The TCC can take the following actions on counter Event 1 (TCCx EV1):

- Counter re-trigger
- Counter direction control
- Stop the counter
- Decrement the counter on event
- Period and pulse width capture
- Non-recoverable fault

The TCC can take the following actions on counter Event 0 (TCCx EV0):

- Counter re-trigger
- Count on event (increment or decrement, depending on counter direction)
- Counter start - start counting on the event rising edge. Further events will not restart the counter; the counter will keep 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 and EVCTRL.EVACT1). For further details, refer to EVCTRL.

Writing a '1' ('0') to an Event Input bit in the Event Control register (EVCTRL.MCEIx or EVCTRL.TCEIx) enables (disables) the corresponding action on input event.

**Note:** When several events are connected to the TCC, the enabled action will apply for each of the incoming events. Refer to *EVSYS – Event System* for details on how to configure the event system.

**Related Links**

EVSYS – Event System on page 544

| Value | Description |
|---|---|
| 0 | The Debug Fault State interrupt is disabled. |
| 1 | The Debug Fault State 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 Compare 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. |

**Bits 19,18,17,16 – MCx: Match or Capture Channel x Interrupt Enable**
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the corresponding Match or Capture Channel x Interrupt Disable/Enable bit, which disables the Match or Capture Channel x interrupt.

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.

#### 39.6.2.15. 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 in INTFLAG(INTFLAG.LPMSUSP) bit which indicates that the USB transceiver is suspended, reducing power consumption. This suspend occurs 9 microseconds after the LPM transaction according to the specification.

To further reduce consumption, it is recommended to stop the USB clock while the device is suspended.

The MCU can also enter in one of the available sleep modes if the wakeup time latency of the selected sleep mode complies with the host latency constraint (see the BESL parameter in 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(INTFLAG.LPMNYET) is set. This generates an interrupt if the Link Power Management Not Yet Interrupt Enable bit in INTENCLR/SET (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.

Table 46-7. Active Current Consumption(1)

| Mode | Conditions | Regulator | PL | Clock | Vcc | Ta | Typ. | Max. | Units |
|---|---|---|---|---|---|---|---|---|---|
| ACTIVE | COREMARK | LDO | PL0 | OSC 12MHz | 1.8V | Max at 85°C Typ at 25°C | 77 | 106 | μA/MHz |
| | | | | | 3.3V | | 79 | 101 | |
| | | | | OSC 8MHz | 1.8V | | 80 | 111 | |
| | | | | | 3.3V | | 82 | 120 | |
| | | | | OSC 4MHz | 1.8V | | 89 | 153 | |
| | | | | | 3.3V | | 92 | 166 | |
| | | | PL2 | DFLL 48MHz | 1.8V | | 93 | 103 | |
| | | | | | 3.3V | | 95 | 105 | |
| | | BUCK | PL0 | OSC 12MHz | 1.8V | | 47 | 60 | |
| | | | | | 3.3V | | 32 | 43 | |
| | | | | OSC 8MHz | 1.8V | | 50 | 71 | |
| | | | | | 3.3V | | 34 | 56 | |
| | | | | OSC 4MHz | 1.8V | | 57 | 98 | |
| | | | | | 3.3V | | 42 | 81 | |
| | | | PL2 | DFLL 48MHz | 1.8V | | 67 | 76 | |
| | | | | | 3.3V | | 40 | 45 | |
| | FIBO | LDO | PL0 | OSC 12MHz | 1.8V | | 76 | 95 | |
| | | | | | 3.3V | | 78 | 98 | |
| | | | | OSC 8MHz | 1.8V | | 79 | 111 | |
| | | | | | 3.3V | | 81 | 119 | |
| | | | | OSC 4MHz | 1.8V | | 89 | 155 | |
| | | | | | 3.3V | | 91 | 173 | |
| | | | PL2 | DFLL 48MHz | 1.8V | | 94 | 104 | |
| | | | | | 3.3V | | 95 | 104 | |
| | | BUCK | PL0 | OSC 12MHz | 1.8V | | 47 | 62 | |
| | | | | | 3.3V | | 31 | 42 | |
| | | | | OSC 8MHz | 1.8V | | 50 | 71 | |
| | | | | | 3.3V | | 34 | 55 | |
| | | | | OSC 4MHz | 1.8V | | 57 | 100 | |
| | | | | | 3.3V | | 42 | 88 | |
| | | | PL2 | DFLL 48MHz | 1.8V | | 67 | 76 | |
| | | | | | 3.3V | | 40 | 45 | |

Refer to the Atmel-ICE, SAM-ICE, JTAGICE3 or SAM L21 Xplained Pro user guides for details on debugging and programming connections and options. For connecting to any other programming or debugging tool, refer to that specific programmer or debugger's user guide.

The SAM L21 Xplained Pro evaluation board supports programming and debugging through the onboard embedded debugger so no external programmer or debugger is needed.

**Note:** A pull-up resistor on the SWCLK pin is critical for reliable operation. Refer to related link for more information.

**Figure 49-12. SWCLK Circuit Connections**

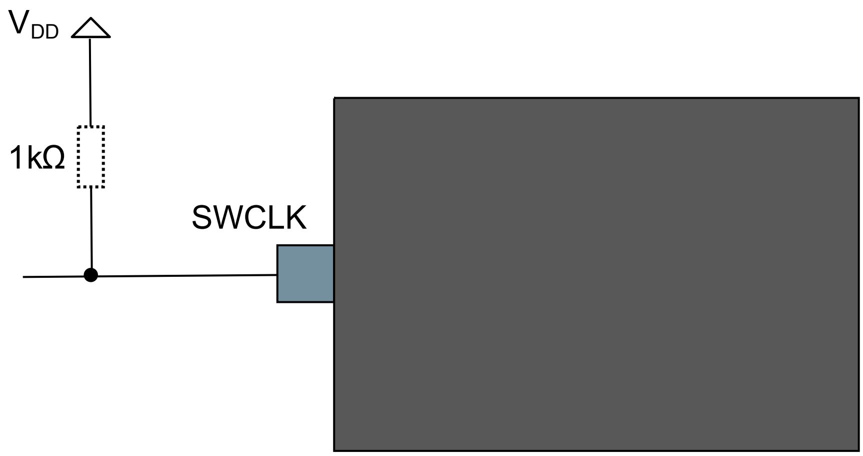



**Table 49-7. SWCLK Circuit Connections**

| Pin Name | Description | Recommended Pin Connection |
|---|---|---|
| SWCLK | Serial wire clock pin | Pull-up resistor 1kΩ |

**Related Links**

Operation in Noisy Environment on page 1201

### 49.7.1. Cortex Debug Connector (10-pin)

For debuggers and/or programmers that support the Cortex Debug Connector (10-pin) interface the signals should be connected as shown in Figure 49-13 with details described in Table 49-8.

### 50.1.8. TRNG

**1 – When TRNG is enabled with configuration CTRL.RUNSTDBY = 0, (disabled during sleep) it could still continue to operate resulting in over-consumption (~50uA) in standby mode.**
**Errata reference: 14827**
**Fix/Workaround:**
Disable the TRNG before entering standby mode.

### 50.1.9. RSTC

**1 – When a System Reset Request is applied, the OSC16MCTRL register is not reset.**
**Errata reference: 13416**
**Fix/Workaround:**
None.

### 50.1.10. Device

**1 – The H2LBRIDGE and H2LBRIDGE latency is high.**
**For a write the latency is 5-cycles (cycles needed for a transaction to propagate from the bridge slave endpoint to the master endpoint). As the bridge supports posted-write, no stall cycles is seen as long as the FIFO is not full.**
**For a read the latency is 4-cycles for the address phase and again 4-cycles for the data to propagate back leading to a total of 8-cycle stall when there is no LP clock division and the accessed slave does not stall.**
**Errata reference: 13414**
**Fix/Workaround:**
None

**2 – In Standby mode, when Power Domain 1 is power gated, devices can show higher consumption than expected.**
**Errata reference: 13599**
**Fix/Workaround:**
Force the Power Domain 1 to remain active by setting PM.PDCFG to 0x2

**3 – In IDLE sleep mode, the APB and AHB clocks are not stopped if the FDPLL is running as a GCLK clock source.**
**Errata reference: 13401**
**Fix/Workaround:**
Disable the FDPLL before entering IDLE sleep mode.

**4 – The low latency mode cannot be enabled by writing a one to address 0x41008120. This bit has no effect.**
**Errata reference: 13506**
**Fix/Workaround:**
None

**5 – XOSC32K contains load capacitor equivalent to Cload=7pf. External load capacitors must take this into account.**
**Errata reference: 13425**
**Fix/Workaround:**
None

| Abbreviation | Description |
|---|---|
| ULP | Ultra-low power |
| USART | Universal Synchronous and Asynchronous Serial Receiver and Transmitter |
| USB | Universal Serial Bus |
| $V_{DD}$ | Common voltage to be applied to VDDIO, VDDIN and VDDANA |
| $V_{DDIN}$ | Digital supply voltage |
| $V_{DDIO}$ | Digital supply voltage |
| $V_{DDANA}$ | Analog supply voltage |
| VREF | Voltage reference |
| WDT | Watchdog Timer |
| XOSC | Crystal Oscillator |