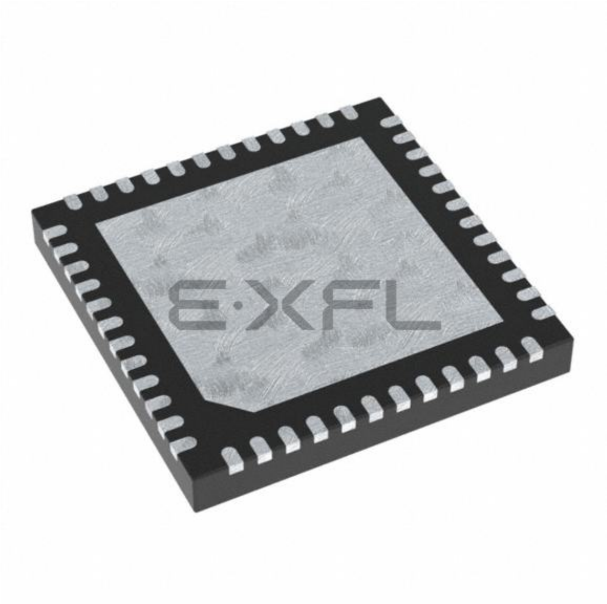


Welcome to E-XFL.COM

## 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 |
| Peripherals | Brown-out Detect/Reset, DMA, POR, WDT |
| Number of I/O | 38 |
| Program Memory Size | 64KB (64K x 8) |
| Program Memory Type | FLASH |
| EEPROM Size | - |
| RAM Size | 8K x 8 |
| Voltage - Supply (Vcc/Vdd) | 2.7V ~ 5.5V |
| Data Converters | A/D 12x12b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 105°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 48-VFQFN Exposed Pad |
| Supplier Device Package | 48-QFN (7x7) |
| Purchase URL | https://www.e-xfl.com/product-detail/microchip-technology/atsamc20g16a-mnt |

- VDDANA: Powers I/O lines and the ADC, AC, PTC, OSCULP32K, OSC32K, and XOSC32K. Voltage is 2.70V to 5.50V.
- VDDCORE: Internal regulated voltage output. Powers the core, memories, peripherals, and FDPLL96M. Voltage is 1.2V typical.

The same voltage must be applied to both VDDIN and VDDANA. This common voltage is referred to as $V_{DD}$ in the datasheet. VDDIO must always be less than or equal to VDDIN.

The ground pins, GND, are common to VDDCORE, VDDIO and VDDIN. The ground pin for VDDANA is GNDANA.

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

### 7.2.2 Voltage Regulator

The SAM C20/C21 voltage regulators have these two modes:

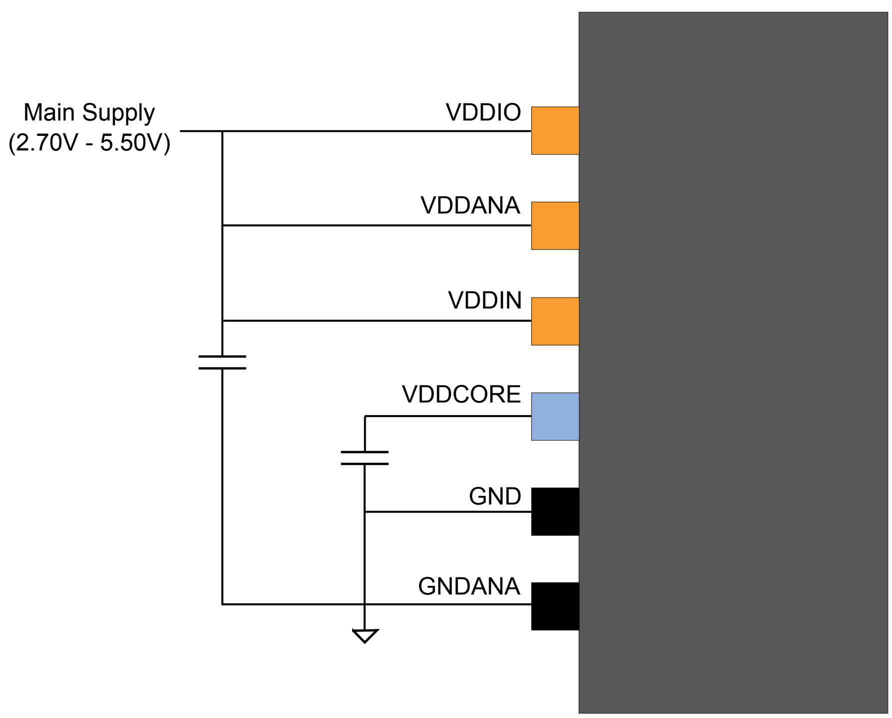
- Normal mode: This is the default mode when CPU and peripherals are running.
- Low Power (LP) mode: This default mode is used when the chip is in standby mode.

### 7.2.3 Typical Powering Schematics

The SAM C20/C21 use a single supply from 2.70V to 5.50V or dual supply mode where VDDIO is supplied separately from VDDIN.

The following figures show the recommended power supply connections.

**Figure 7-3. Power Supply Connection for Single Supply Mode Only**



arriving interrupts. Refer to Nested Vector Interrupt Controller and the Cortex-M0+ Technical Reference Manual for details (http://www.arm.com).

- System Timer (SysTick)
  – The System Timer is a 24-bit timer clocked by CLK_CPU that extends the functionality of both the processor and the NVIC. Refer to the Cortex-M0+ Technical Reference Manual for details (http://www.arm.com).
- System Control Block (SCB)
  – The System Control Block provides system implementation information, and system control. This includes configuration, control, and reporting of the system exceptions. Refer to the Cortex-M0+ Devices Generic User Guide for details (http://www.arm.com).
- Micro Trace Buffer (MTB)
  – The CoreSight MTB-M0+ (MTB) provides a simple execution trace capability to the Cortex-M0+ processor. Refer to section Micro Trace Buffer and the CoreSight MTB-M0+ Technical Reference Manual for details (http://www.arm.com).
- Memory Protection Unit (MPU)
  – The Memory Protection Unit divides the memory map into a number of regions, and defines the location, size, access permissions and memory attributes of each region. Refer to the Cortex-M0+ Devices Generic User Guide for details (http://www.arm.com)

### 10.1.3 Cortex-M0+ Address Map

**Table 10-2. Cortex-M0+ Address Map**

| Address | Peripheral |
|---|---|
| 0xE000E000 | System Control Space (SCS) |
| 0xE000E010 | System Timer (SysTick) |
| 0xE000E100 | Nested Vectored Interrupt Controller (NVIC) |
| 0xE000ED00 | System Control Block (SCB) |
| 0x41008000 | Micro Trace Buffer (MTB) |

**Related Links**

Product Mapping

### 10.1.4 I/O Interface

#### 10.1.4.1 Overview

Because accesses to the AMBA® AHB-Lite™ and the single cycle I/O interface can be made concurrently, the Cortex-M0+ processor can fetch the next instructions while accessing the I/Os. This enables single cycle I/O accesses to be sustained for as long as needed.

#### 10.1.4.2 Description

Direct access to PORT registers and DIVAS registers.

| Peripheral Name | Base Address | IRQ Line | AHB Clock | | APB Clock | | Generic Clock | PAC | | Events | | DMA | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | Index | Enabled at Reset | Index | Enabled at Reset | Index | Index | Prot at Reset | User | Generator | Index | Sleep Walking |
| DAC | 0x42005400 | 28 | | | 21 | N | 36 | 21 | N | 38: START | 78: EMPTY | 45: EMPTY | Y |
| PTC | 0x42005800 | 30 | | | 22 | N | 37 | 22 | N | 39: STCONV | 79: EOC<br>80: WCOMP | EOC: 46<br>WCOMP: 47<br>SEQ: 48 | |
| CCL | 0x42005C00 | | | | 23 | N | 38 | 23 | N | 40-43 : LUTIN0-3 | 781-84: LUTOUT0-3 | | Y |
| DIVAS | 0x48000000 | | 12 | Y | | | | | | | | | N/A |

## Table 12-4. Peripherals Configuration Summary SAM C20 E/G/J

| Peripheral Name | Base Address | IRQ Line | AHB Clock | | APB Clock | | Generic Clock | PAC | | Events | | DMA | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | Index | Enabled at Reset | Index | Enabled at Reset | Index | Index | Prot at Reset | User | Generator | Index | Sleep Walking |
| AHB-APB Bridge A | 0x40000000 | | 0 | Y | | | | | | | | | N/A |
| PAC | 0x44000000 | 0 | 10 | Y | 0 | Y | | 0 | N | | 85 : ACCERR | | N/A |
| PM | 0x40000400 | 0 | | | 1 | Y | | 1 | N | | | | N/A |
| MCLK | 0x40000800 | 0 | | | 2 | Y | | 2 | N | | | | Y |
| RSTC | 0x40000C00 | | | | 3 | Y | | 3 | N | | | | N/A |
| OSCCTRL | 0x40001000 | 0 | | | 4 | Y | 0: FDPLL96M clk source<br>1: FDPLL96M 32kHz | 4 | N | | 0: XOSC_FAIL | | Y |
| OSC32KCTRL | 0x40001400 | 0 | | | 5 | Y | | 5 | N | | 1: XOSC32K_FAIL | | Y |
| SUPC | 0x40001800 | 0 | | | 6 | Y | | 6 | N | | | | N/A |
| GCLK | 0x40001C00 | | | | 7 | Y | | 7 | N | | | | N/A |
| WDT | 0x40002000 | 1 | | | 8 | Y | | 8 | N | | | | Y |
| RTC | 0x40002400 | 2 | | | 9 | Y | | 9 | N | | 2: CMP0/ALARM0<br>3: CMP1<br>4: OVF<br>5-12: PER0-7 | | Y |
| EIC | 0x40002800 | 3, NMI | | | 10 | Y | 2 | 10 | N | | 13-28: EXTINT0-15 | | Y |
| FREQM | 0x40002C00 | 4 | | | 11 | Y | 3: Measure<br>4: Reference | 11 | N | | | | N/A |
| AHB-APB Bridge B | 0x41000000 | | 1 | Y | | | | | | | | | N/A |
| PORT | 0x41000000 | | | | 0 | Y | | 0 | N | 1-4 : EV0-3 | | | Y |
| DSU | 0x41002000 | | 3 | Y | 1 | Y | | 1 | Y | | | | N/A |
| NVMCTRL | 0x41004000 | 6 | 5 | Y | 2 | Y | 39 | 2 | N | | | | Y |
| DMAC | 0x41006000 | 7 | 7 | Y | | | | 3 | N | 5-8: CH0-3 | 30-33: CH0-3 | | Y |
| MTB | 0x41008000 | | | | | | | | N | 44: START<br>45: STOP | | | N/A |
| AHB-APB Bridge C | 0x42000000 | | 2 | Y | | | | | | | | | N/A |
| EVSYS | 0x42000000 | 8 | | | 0 | N | 6-17: one per CHANNEL | 0 | N | | | | Y |
| SERCOM0 | 0x42000400 | 9 | | | 1 | N | 19: CORE<br>18: SLOW | 1 | N | | | 2: RX<br>3: TX | Y |
| SERCOM1 | 0x42000800 | 10 | | | 2 | N | 20: CORE<br>18: SLOW | 2 | N | | | 4: RX<br>5: TX | Y |
| SERCOM2 | 0x42000C00 | 11 | | | 3 | N | 21: CORE<br>18: SLOW | 3 | N | | | 6: RX<br>7: TX | Y |

NVMCTRL – Non-Volatile Memory Controller
Security Bit

## 13.10 Device Identification

Device identification relies on the ARM CoreSight component identification scheme, which allows the chip to be identified as a SAM device implementing a DSU. The DSU contains identification registers to differentiate the device.

### 13.10.1 CoreSight Identification

A system-level ARM CoreSight ROM table is present in the device to identify the vendor and the chip identification method. Its address is provided in the MEM-AP BASE register inside the ARM Debug Access Port. The CoreSight ROM implements a 64-bit conceptual ID composed as follows from the PID0 to PID7 CoreSight ROM Table registers:

**Figure 13-5. Conceptual 64-bit Peripheral ID**

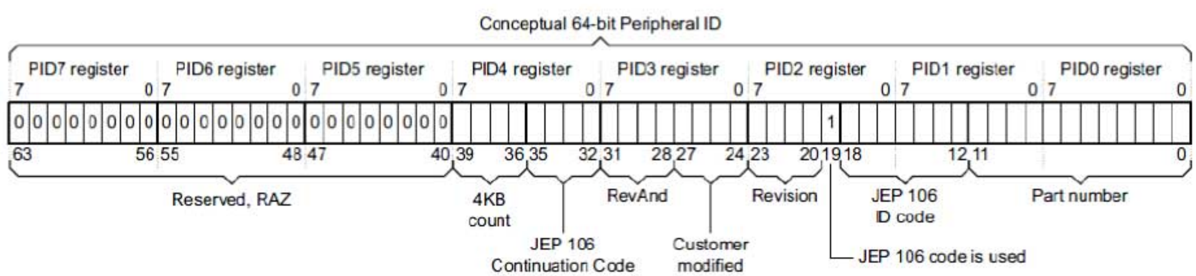



**Table 13-2. Conceptual 64-Bit Peripheral ID Bit Descriptions**

| Field | Size | Description | Location |
|---|---|---|---|
| JEP-106 CC code | 4 | Continuation code: 0x0 | PID4 |
| JEP-106 ID code | 7 | Device ID: 0x1F | PID1+PID2 |
| 4KB count | 4 | Indicates that the CoreSight component is a ROM: 0x0 | PID4 |
| RevAnd | 4 | Not used; read as 0 | PID3 |
| CUSMOD | 4 | Not used; read as 0 | PID3 |
| PARTNUM | 12 | Contains 0xCD0 to indicate that DSU is present | PID0+PID1 |
| REVISION | 4 | DSU revision (starts at 0x0 and increments by 1 at both major and minor revisions). Identifies DSU identification method variants. If 0x0, this indicates that device identification can be completed by reading the Device Identification register (DID) | PID3 |

For more information, refer to the ARM Debug Interface Version 5 Architecture Specification.

### 13.10.2 Chip Identification Method

The DSU DID register identifies the device by implementing the following information:

- Processor identification
- Product family identification
- Product series identification

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

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

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

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | SMEMP |
| Access | | | | | | | | R |
| Reset | | | | | | | | x |

**Bit 0 – SMEMP: System Memory Present**
This bit indicates whether system memory is present on the bus that connects to the ROM table.

This bit is set at power-up if the device is not protected, indicating that the system memory is accessible from a debug adapter.

This bit is cleared at power-up if the device is protected, indicating that the system memory is not accessible from a debug adapter.

### 13.13.14 Peripheral Identification 4

**Name:** PID4
**Offset:** 0x1FD0
**Reset:** 0x00000000
**Property:** -

## 15.6 Power Consumption vs. Speed

Due to the nature of the asynchronous clocking of the peripherals there are some considerations that needs to be taken if either targeting a low-power or a fast-acting system. If clocking a peripheral with a very low clock, the active power consumption of the peripheral will be lower. At the same time the synchronization to the synchronous (CPU) clock domain is dependent on the peripheral clock speed, and will be longer with a slower peripheral clock; giving lower response time and more time waiting for the synchronization to complete.

## 15.7 Clocks after Reset

On any reset the synchronous clocks start to their initial state:

- OSC48M is enabled and divided by 12
- GCLK_MAIN uses OSC48M as source
- CPU and BUS clocks are undivided

On a power reset the GCLK starts to their initial state:

- All generic clock generators disabled except:
  - The generator 0 (GCLK_MAIN) using OSC48M as source, with no division
- All generic clocks disabled

On a user reset the GCLK starts to their initial state, except for:

- Generic clocks that are write-locked (WRTLOCK is written to one prior to reset)

On any reset the clock sources are reset to their initial state except the 32KHz clock sources which are reset only by a power reset.

# 19. PM – Power Manager

**Related Links**

Sleep Mode Operation

## 19.1 Overview

The Power Manager (PM) controls the sleep modes of the device.

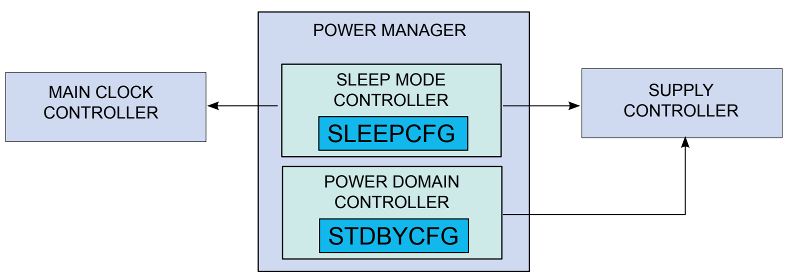
Various sleep modes are provided in order to fit power consumption requirements. This enables the PM to stop unused modules in order to save power. In active mode, the CPU is executing application code. When the device enters a sleep mode, program execution is stopped and some modules and clock domains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

## 19.2 Features

- Power management control
  - Sleep modes: Idle, Standby

## 19.3 Block Diagram

**Figure 19-1. PM Block Diagram**




## 19.4 Signal Description

Not applicable.

## 19.5 Product Dependencies

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

### 19.5.1 I/O Lines

Not applicable.

### 19.5.2 Clocks

The PM bus clock (CLK_PM_APB) can be enabled and disabled in the Main Clock module. If this clock is disabled, it can only be re-enabled by a system reset.

**Table 23-1. WDT Operating Modes**

| CTRLA.ENABLE | CTRLA.WEN | Interrupt Enable | Mode |
|---|---|---|---|
| 0 | x | x | Stopped |
| 1 | 0 | 0 | Normal mode |
| 1 | 0 | 1 | Normal mode with Early Warning interrupt |
| 1 | 1 | 0 | Window mode |
| 1 | 1 | 1 | Window mode with Early Warning interrupt |

### 23.6.2 Basic Operation

#### 23.6.2.1 Initialization

The following bits are enable-protected, meaning that they can only be written when the WDT is disabled (CTRLA.ENABLE=0):

- Control A register (CTRLA), except the Enable bit (CTRLA.ENABLE)
- Configuration register (CONFIG)
- Early Warning Interrupt Control register (EWCTRL)

Enable-protected bits in the CTRLA register can be written at the same time as CTRLA.ENABLE is written to '1', but not at the same time as CTRLA.ENABLE is written to '0'.

The WDT can be configured only while the WDT is disabled. The WDT is configured by defining the required Time-Out Period bits in the Configuration register (CONFIG.PER). If Window mode operation is desired, the Window Enable bit in the Control A register must be set (CTRLA.WEN=1) and the Window Period bits in the Configuration register (CONFIG.WINDOW) must be defined.

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

#### 23.6.2.2 Configurable Reset Values

After a Power-on Reset, some registers will be loaded with initial values from the NVM User Row.

This includes the following bits and bit groups:

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

**Related Links**

NVM User Row Mapping

#### 23.6.2.3 Enabling, Disabling, and Resetting

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

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

**Name:** SYNCBUSY
**Offset:** 0x08 [ID-0000067a]
**Reset:** 0x00000000
**Property:** -

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

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

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

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | WEN | ENABLE | |
| Access | | | | | | R | R | |
| Reset | | | | | | 0 | 0 | |

**Bit 2 – WEN: Window Enable Synchronization Busy**

| Value | Description |
|---|---|
| 0 | Write synchronization of the CTRLA.WEN bit is complete. |
| 1 | Write synchronization of the CTRLA.WEN bit is ongoing. |

**Bit 1 – ENABLE: Enable Synchronization Busy**

| Value | Description |
|---|---|
| 0 | Write synchronization of the CTRLA.ENABLE bit is complete. |
| 1 | Write synchronization of the CTRLA.ENABLE bit is ongoing. |

### 23.8.8 Clear

**Name:** CLEAR
**Offset:** 0x0C [ID-0000067a]
**Reset:** 0x00
**Property:** Write-Synchronized

| Value | Name | Description |
|---|---|---|
| 0xA | DIV512 | CLK_RTC_CNT = GCLK_RTC/512 |
| 0xB | DIV1024 | CLK_RTC_CNT = GCLK_RTC/1024 |
| 0xC-0xF | - | Reserved |

**Bit 7 – MATCHCLR: Clear on Match**
This bit is valid only in Mode 0 (COUNT32) and Mode 2 (CLOCK). This bit can be written only when the peripheral is disabled. This bit is not synchronized.

| Value | Description |
|---|---|
| 0 | The counter is not cleared on a Compare/Alarm 0 match |
| 1 | The counter is cleared on a Compare/Alarm 0 match |

**Bit 6 – CLKREP: Clock Representation**
This bit is valid only in Mode 2 and determines how the hours are represented in the Clock Value (CLOCK) register. This bit can be written only when the peripheral is disabled. This bit is not synchronized.

| Value | Description |
|---|---|
| 0 | 24 Hour |
| 1 | 12 Hour (AM/PM) |

**Bits 3:2 – MODE[1:0]: Operating Mode**
This field defines the operating mode of the RTC. This bit is not synchronized.

| Value | Name | Description |
|---|---|---|
| 0x0 | COUNT32 | Mode 0: 32-bit counter |
| 0x1 | COUNT16 | Mode 1: 16-bit counter |
| 0x2 | CLOCK | Mode 2: Clock/calendar |
| 0x3 | - | Reserved |

**Bit 1 – ENABLE: Enable**
Due to synchronization there is delay from writing CTRLA.ENABLE until the peripheral is enabled/ disabled. The value written to CTRLA.ENABLE will read back immediately and the Enable bit in the Synchronization Busy register (SYNCBUSY.ENABLE) will be set. SYNCBUSY.ENABLE will be cleared when the operation is complete.

| Value | Description |
|---|---|
| 0 | The peripheral is disabled |
| 1 | The peripheral is enabled |

**Bit 0 – SWRST: Software Reset**
Writing a '0' to this bit has no effect.

Writing a '1' to this bit resets all registers in the RTC, except DBGCTRL, to their initial state, and the RTC will be disabled.

Writing a '1' to CTRLA.SWRST will always take precedence, meaning that all other writes in the same write-operation will be discarded.

Due to synchronization there is a delay from writing CTRLA.SWRST until the reset is complete. CTRLA.SWRST will be cleared when the reset is complete.

### 24.12.10 Alarm Value in Clock/Calendar mode (CTRLA.MODE=2)

The 32-bit value of ALARM is continuously compared with the 32-bit CLOCK value, based on the masking set by MASK.SEL. When a match occurs, the Alarm n interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.ALARM) is set on the next counter cycle, and the counter is cleared if CTRLA.MATCHCLR is '1'.

**Name:** ALARM
**Offset:** 0x20
**Reset:** 0x00000000
**Property:** PAC Write-Protection, Write-Synchronized

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | YEAR[5:0] | | | | | | MONTH[3:2] | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | MONTH[1:0] | | DAY[4:0] | | | | | HOUR[4:4] |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | HOUR[3:0] | | | | MINUTE[5:2] | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

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

**Bits 31:26 – YEAR[5:0]: Year**
The alarm year. Years are only matched if MASK.SEL is 6

**Bits 25:22 – MONTH[3:0]: Month**
The alarm month. Months are matched only if MASK.SEL is greater than 4.

**Bits 21:17 – DAY[4:0]: Day**
The alarm day. Days are matched only if MASK.SEL is greater than 3.

**Bits 16:12 – HOUR[4:0]: Hour**
The alarm hour. Hours are matched only if MASK.SEL is greater than 2.

**Bits 11:6 – MINUTE[5:0]: Minute**
The alarm minute. Minutes are matched only if MASK.SEL is greater than 1.

**Bits 5:0 – SECOND[5:0]: Second**
The alarm second. Seconds are matched only if MASK.SEL is greater than 0.

**Bits 23:22 – TRIGACT[1:0]: Trigger Action**
These bits define the trigger action used for a transfer.

| TRIGACT[1:0] | Name | Description |
|---|---|---|
| 0x0 | BLOCK | One trigger required for each block transfer |
| 0x1 | - | Reserved |
| 0x2 | BEAT | One trigger required for each beat transfer |
| 0x3 | TRANSACTION | One trigger required for each transaction |

**Bits 13:8 – TRIGSRC[5:0]: Trigger Source**
These bits define the peripheral trigger which is source of the transfer. For details on trigger selection and trigger modes, refer to Transfer Triggers and Actions and CHCTRLB.TRIGACT.

**Table 25-2. Peripheral Trigger Source**

| Value | Name | Description |
|---|---|---|
| 0x00 | DISABLE | Only software/event triggers |
| 0x01 | TSENS | TSENS Result Ready Trigger |
| 0x02 | SERCOM0 RX | SERCOM0 RX Trigger |
| 0x03 | SERCOM0 TX | SERCOM0TX Trigger |
| 0x04 | SERCOM1 RX | SERCOM1 RX Trigger |
| 0x05 | SERCOM1 TX | SERCOM1 TX Trigger |
| 0x06 | SERCOM2 RX | SERCOM2 RX Trigger |
| 0x07 | SERCOM2 TX | SERCOM2 TX Trigger |
| 0x08 | SERCOM3 RX | SERCOM3 RX Trigger |
| 0x09 | SERCOM3 TX | SERCOM3 TX Trigger |
| 0x0A | SERCOM4 RX- | SERCOM4 RX TriggerReserved |
| 0x0B | SERCOM4 TX- | SERCOM4 TX TriggerReserved |
| 0x0C | SERCOM5 RX- | SERCOM5 RX TriggerReserved |
| 0x0D | SERCOM5 TX- | SERCOM5 TX TriggerReserved |
| 0x0E | CAN0 DEBUG- | CAN0 Debug TriggerReserved |
| 0x0F | CAN1 DEBUG- | CAN1 Debug TriggerReserved |
| 0x10 | TCC0 OVF | TCC0 Overflow Trigger |
| 0x11 | TCC0 MC0 | TCC0 Match/Compare 0 Trigger |
| 0x12 | TCC0 MC1 | TCC0 Match/Compare 1 Trigger |
| 0x13 | TCC0 MC2 | TCC0 Match/Compare 2 Trigger |
| 0x14 | TCC0 MC3 | TCC0 Match/Compare 3 Trigger |
| 0x15 | TCC1 OVF | TCC1 Overflow Trigger |

**Name:** SYNCBUSY
**Offset:** 0x04
**Reset:** 0x00000000
**Property:** –

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

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

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

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | ENABLE | SWRST |
| Access | | | | | | | R | R |
| Reset | | | | | | | 0 | 0 |

**Bit 1 – ENABLE: Enable Synchronization Busy Status**

| Value | Description |
|---|---|
| 0 | Write synchronization for CTRLA.ENABLE bit is complete. |
| 1 | Write synchronization for CTRLA.ENABLE bit is ongoing. |

**Bit 0 – SWRST: Software Reset Synchronization Busy Status**

| Value | Description |
|---|---|
| 0 | Write synchronization for CTRLA.SWRST bit is complete. |
| 1 | Write synchronization for CTRLA.SWRST bit is ongoing. |

### 26.8.5 Event Control

**Name:** EVCTRL
**Offset:** 0x08
**Reset:** 0x00000000
**Property:** PAC Write-Protection, Enable-Protected

**Name:** PINSTATE
**Offset:** 0x38
**Reset:** 0x00000000

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

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

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

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

**Bits 31:0 – PINSTATE[31:0]: Pin State**
These bits return the valid pin state of the debounced external interrupt pin EXTINTx.

#### 27.6.4.2 RWWEE Read

Reading from the RWW EEPROM address space is performed via the AHB bus by addressing the RWWEE address space directly.

Read timings are similar to regular NVM read timings when access size is Byte or half-Word. The AHB data phase is twice as long in case of full-Word-size access.

It is not possible to read the RWWEE area while the NVM main array is being written or erased, whereas the RWWEE area can be written or erased while the main array is being read.

The RWWEE address space is not cached, therefore it is recommended to limit access to this area for performance and power consumption considerations.

#### 27.6.4.3 NVM Write

The NVM Controller requires that an erase must be done before programming. The entire NVM main address space and the RWWEE address space can be erased by a debugger Chip Erase command. Alternatively, rows can be individually erased by the Erase Row command or the RWWEE Erase Row command to erase the NVM main address space or the RWWEE address space, respectively.

After programming the NVM main array, the region that the page resides in can be locked to prevent spurious write or erase sequences. Locking is performed on a per-region basis, and so, locking a region will lock all pages inside the region.

Data to be written to the NVM block are first written to and stored in an internal buffer called the *page buffer*. The page buffer contains the same number of bytes as an NVM page. Writes to the page buffer must be 16 or 32 bits. 8-bit writes to the page buffer are not allowed and will cause a system exception.

Internally, writes to the page buffer are on a 64-bit basis through the page buffer load data register (PBLDATA1 and PBLDATA0). The PBLDATA register is a holding register for writes to the same 64-bit page buffer section. Data within a 64-bit section can be written in any order. Crossing a 64-bit boundary will reset the PBLDATA register to all ones. The following example assumes startup from reset where the current address is 0 and PBLDATA is all ones. Only 64 bits of the page buffer are written at a time, but 128 bits are shown for reference.

**Sequential 32-bit Write Example:**

- 32-bit 0x1 written to address 0
  - Page buffer[127:0] = {0xFFFFFFFF_FFFFFFFF, PBLDATA[63:32], 0x00000001}
  - PBLDATA[63:0] = {PBLDATA[63:32], 0x00000001}
- 32-bit 0x2 written to address 1
  - Page buffer[127:0] = {0xFFFFFFFF_FFFFFFFF, 0x00000002, PBLDATA[31:0]
  - PBLDATA[63:0] = 0x00000002, PBLDATA[31:0]}
- 32-bit 0x3 written to address 2 (crosses 64-bit boundary)
  - Page buffer[127:0] = 0xFFFFFFFF_00000003_00000002_00000001
  - PBLDATA[63:0] = 0xFFFFFFFF_00000003

Random access writes to 32-bit words within the page buffer will overwrite the opposite word within the same 64-bit section with ones. In the following example, notice that 0x00000001 is overwritten with 0xFFFFFFFF from the third write due to the 64-bit boundary crossing. Only 64 bits of the page buffer are written at a time, but 128 bits are shown for reference.

**Random Access 32-bit Write Example:**

- 32-bit 0x1 written to address 2
  - Page buffer[127:0] = 0xFFFFFFFF_00000001_FFFFFFFF_FFFFFFFF
  - PBLDATA[63:0] = 0xFFFFFFFF_00000001

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

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

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

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

**Bits 31:0 – IN[31:0]: PORT Data Input Value**
These bits are cleared when the corresponding I/O pin input sampler detects a logical low level on the input pin.

These bits are set when the corresponding I/O pin input sampler detects a logical high level on the input pin.

### 28.9.10 Control

**Tip:** The I/O pins are assembled in pin groups (”PORT groups”) with up to 32 pins. Group 0 consists of the PA pins, group 1 is for the PB pins, etc. Each pin group has its own PORT registers. For example, the register address offset for the Data Direction (DIR) register for group 0 (PA00 to PA31) is 0x00, and the register address offset for the DIR register for group 1 (PB00 to PB31) is 0x80.

**Name:** CTRL
**Offset:** 0x24
**Reset:** 0x00000000
**Property:** PAC Write-Protection

4. In the event user peripheral, enable event input by writing a '1' to the respective Event Input Enable bit ("EI") in the peripheral's Event Control register (e.g., AC.EVCTRL.IVEI0, ADC.EVCTRL.STARTEI).

#### 29.6.2.2 Enabling, Disabling, and Resetting

The EVSYS is always enabled.

The EVSYS is reset by writing a ‘1’ to the Software Reset bit in the Control A register (CTRLA.SWRST). All registers in the EVSYS will be reset to their initial state and all ongoing events will be canceled.

Refer to CTRLA.SWRST register for details.

#### 29.6.2.3 User Multiplexer Setup

The user multiplexer defines the channel to be connected to which event user. Each user multiplexer is dedicated to one event user. A user multiplexer receives all event channels output and must be configured to select one of these channels, as shown in Block Diagram section. The channel is selected with the Channel bit group in the User register (USERm.CHANNEL).

The user multiplexer must always be configured before the channel. A list of all user multiplexers is found in the User (USERm) register description.

**Related Links**

USERm

#### 29.6.2.4 Event System Channel

An event channel can select one event from a list of event generators. Depending on configuration, the selected event could be synchronized, resynchronized or asynchronously sent to the users. When synchronization or resynchronization is required, the channel includes an internal edge detector, allowing the Event System to generate internal events when rising, falling or both edges are detected on the selected event generator.

An event channel is able to generate internal events for the specific software commands. A channel block diagram is shown in *Block Diagram* section.

#### 29.6.2.5 Event Generators

Each event channel can receive the events form all event generators. All event generators are listed in the Event Generator bit field in the Channel n register (CHANNELn.EVGEN). For details on event generation, refer to the corresponding module chapter. The channel event generator is selected by the Event Generator bit group in the Channel register (CHANNELn.EVGEN). By default, the channels are not connected to any event generators (ie, CHANNELn.EVGEN = 0)

#### 29.6.2.6 Channel Path

There are three different ways to propagate the event from an event generator:

- Asynchronous path
- Synchronous path
- Resynchronized path

The path is decided by writing to the Path Selection bit group of the Channel register (CHANNELn.PATH).

**Asynchronous Path**

When using the asynchronous path, the events are propagated from the event generator to the event user without intervention from the Event System. The GCLK for this channel (GCLK_EVSYS_CHANNEL_n) is not mandatory, meaning that an event will be propagated to the user without any clock latency.

**Bit 8 – OVFEO: Overflow/Underflow Event Output Enable**

This bit enables the Overflow/Underflow event. When enabled, an event will be generated when the counter overflows/underflows.

| Value | Description |
|---|---|
| 0 | Overflow/Underflow event is disabled and will not be generated. |
| 1 | Overflow/Underflow event is enabled and will be generated for every counter overflow/underflow. |

**Bit 5 – TCEI: TC Event Enable**

This bit is used to enable asynchronous input events to the TC.

| Value | Description |
|---|---|
| 0 | Incoming events are disabled. |
| 1 | Incoming events are enabled. |

**Bit 4 – TCINV: TC Inverted Event Input Polarity**

This bit inverts the asynchronous input event source.

| Value | Description |
|---|---|
| 0 | Input event source is not inverted. |
| 1 | Input event source is inverted. |

**Bits 2:0 – EVACT[2:0]: Event Action**

These bits define the event action the TC will perform on an event.

| Value | Name | Description |
|---|---|---|
| 0x0 | OFF | Event action disabled |
| 0x1 | RETRIGGER | Start, restart or retrigger TC on event |
| 0x2 | COUNT | Count on event |
| 0x3 | START | Start TC on event |
| 0x4 | STAMP | Time stamp capture |
| 0x5 | PPW | Period captured in CC0, pulse width in CC1 |
| 0x6 | PWP | Period captured in CC1, pulse width in CC0 |
| 0x7 | PW | Pulse width capture |

### 35.7.2.5 Interrupt Enable Clear

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set register (INTENSET).

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

- Dead-time insertion
- Swap
- Pattern generation

See also Figure 36-1.

The output matrix (OTMX) can distribute and route out the TCC waveform outputs across the port pins in different configurations, each optimized for different application types. The Dead-Time Insertion (DTI) unit splits the four lower OTMX outputs into two non-overlapping signals: the non-inverted low side (LS) and inverted high side (HS) of the waveform output with optional dead-time insertion between LS and HS switching. The SWAP unit can swap the LS and HS pin outputs, and can be used for fast decay motor control.

The pattern generation unit can be used to generate synchronized waveforms with constant logic level on TCC UPDATE conditions. This is useful for easy stepper motor and full bridge control.

The non-recoverable fault module enables event controlled fault protection by acting directly on the generated waveforms of the timer/counter compare channel outputs. When a non-recoverable fault condition is detected, the output waveforms are forced to a preconfigured value that is safe for the application. This is typically used for instant and predictable shut down and disabling high current or voltage drives.

The count event sources (TCE0 and TCE1) are shared with the non-recoverable fault extension. The events can be optionally filtered. If the filter options are not used, the non-recoverable faults provide an immediate asynchronous action on waveform output, even for cases where the clock is not present. For further details on how to configure asynchronous events routing, refer to section *EVSYS – Event System*.

**Related Links**

EVSYS – Event System

### 36.6.2 Basic Operation

#### 36.6.2.1 Initialization

The following registers are enable-protected, meaning that they can only be written when the TCC is disabled(CTRLA.ENABLE=0):

- Control A (CTRLA) register, except Run Standby (RUNSTDBY), Enable (ENABLE) and Software Reset (SWRST) bits
- Recoverable Fault n Control registers (FCTRLA and FCTRLB)
- Waveform Extension Control register (WEXCTRL)
- Drive Control register (DRVCTRL)
- Event Control register (EVCTRL)

Enable-protected bits in the CTRLA register can be written at the same time as CTRLA.ENABLE is written to '1', but not at the same time as CTRLA.ENABLE is written to '0'. Enable-protection is denoted by the “Enable-Protected” property in the register description.

Before the TCC is enabled, it must be configured as outlined by the following steps:

1. Enable the TCC bus clock (CLK_TCCx_APB).
2. If Capture mode is required, enable the channel in capture mode by writing a '1' to the Capture Enable bit in the Control A register (CTRLA.CPTEN).

Optionally, the following configurations can be set before enabling TCC:

1. Select PRESCALER setting in the Control A register (CTRLA.PRESCALER).
2. Select Prescaler Synchronization setting in Control A register (CTRLA.PRESCSYNC).

| Value | Description |
|---|---|
| 0 | There is no reset operation ongoing. |
| 1 | The reset operation is ongoing. |

### 38.8.2 Control B

**Name:** CTRLB
**Offset:** 0x01 [ID-0000120e]
**Reset:** 0x00
**Property:** PAC Write-Protection, Enable-Protected

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

**Bits 2:0 – PRESCALER[2:0]: Prescaler Configuration**
This field defines the ADC clock relative to the peripheral clock.

This field is not synchronized. For the slave ADC, these bits have no effect when the SLAVEEN bit is set (CTRLA.SLAVEEN= 1).

| Value | Name | Description |
|---|---|---|
| 0x0 | DIV2 | Peripheral clock divided by 2 |
| 0x1 | DIV4 | Peripheral clock divided by 4 |
| 0x2 | DIV8 | Peripheral clock divided by 8 |
| 0x3 | DIV16 | Peripheral clock divided by 16 |
| 0x4 | DIV32 | Peripheral clock divided by 32 |
| 0x5 | DIV64 | Peripheral clock divided by 64 |
| 0x6 | DIV128 | Peripheral clock divided by 128 |
| 0x7 | DIV256 | Peripheral clock divided by 256 |

### 38.8.3 Reference Control

**Name:** REFCTRL
**Offset:** 0x02 [ID-0000120e]
**Reset:** 0x00
**Property:** PAC Write-Protection, Enable-Protected

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

**Bit 7 – REFCOMP: Reference Buffer Offset Compensation Enable**
The gain error can be reduced by enabling the reference buffer offset compensation. This will decrease the input impedance and thus increase the start-up time of the reference.