

Welcome to [E-XFL.COM](https://www.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	I ² C, 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	8K 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-VFQFN Exposed Pad
Supplier Device Package	24-VQFN (4x4)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atsaml11d14a-mf

15.7.13 Peripheral Non-Secure Status - Bridge B

Name: NONSECB
Offset: 0x58
Reset: x initially determined from NVM User Row after reset
Property: Write-Secure



Important: This register is only available for **SAM L11** and has no effect for **SAM L10**.

Reading NONSEC register returns peripheral security attribution status:

Value	Description
0	Peripheral is secured.
1	Peripheral is non-secured.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
Access				R/R/R	R/R/R	R/R/R	R/R/R	R/R/R
Reset				x	x	0	1	0

Bit 4 – HMATRIXHS Peripheral HMATRIXHS Non-Secure

Bit 3 – DMAC Peripheral DMAC Non-Secure

Bit 2 – NVMCTRL Peripheral NVMCTRL Non-Secure
 The NVMCTRL Peripheral is always secured.

Bit 1 – DSU Peripheral DSU Non-Secure
 The DSU Peripheral is always non-secured.

REGC (16-bit access) can be written without affecting REGA or REGB. If REGC is written to in two consecutive 8-bit accesses without waiting for synchronization, the second write attempt will be discarded and an error is generated through the PAC.

A 32-bit access to offset 0x00 will write all three registers. Note that REGA, REGB and REGC can be updated at different times because of independent write synchronization.

17.3.3 General Read Synchronization

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

Note: The corresponding bits in SYNCBUSY will automatically be set when the device wakes up from sleep because read-synchronized registers need to be synchronized. Therefore reading a read-synchronized register before its corresponding SYNCBUSY bit is cleared will return the last synchronized value before sleep mode.

Moreover, if a register is also write-synchronized, any write access while the SYNCBUSY bit is set will be discarded and generate an error.

17.3.4 Completion of Synchronization

In order to check if synchronization is complete, the user can either poll the relevant bits in SYNCBUSY or use the Synchronisation Ready interrupt (if available). The Synchronization Ready interrupt flag will be set when all ongoing synchronizations are complete, i.e. when all bits in SYNCBUSY are '0'.

17.3.5 Write Synchronization for CTRLA.ENABLE

Setting the Enable bit in a module's Control A register (CTRLA.ENABLE) will trigger write-synchronization and set SYNCBUSY.ENABLE.

CTRLA.ENABLE will read its new value immediately after being written.

SYNCBUSY.ENABLE will be cleared by hardware when the operation is complete.

The Synchronization Ready interrupt (if available) cannot be used to enable write-synchronization.

17.3.6 Write-Synchronization for Software Reset Bit

Setting the Software Reset bit in CTRLA (CTRLA.SWRST=1) will trigger write-synchronization and set SYNCBUSY.SWRST. When writing a '1' to the CTRLA.SWRST bit it will immediately read as '1'.

CTRLA.SWRST and SYNCBUSY.SWRST will be cleared by hardware when the peripheral has been reset.

Writing a '0' to the CTRLA.SWRST bit has no effect.

The Ready interrupt (if available) cannot be used for Software Reset write-synchronization.

Note: Not all peripherals have the SWRST bit in the respective CTRLA register.

17.3.7 Synchronization Delay

The synchronization will delay write and read accesses by a certain amount. This delay D is within the range of:

$$5 \times P_{GCLK} + 2 \times P_{APB} < D < 6 \times P_{GCLK} + 3 \times P_{APB}$$

Where P_{GCLK} is the period of the generic clock and P_{APB} is the period of the peripheral bus clock. A normal peripheral bus register access duration is $2 \times P_{APB}$.

SAM L10/L11 Family

GCLK - Generic Clock Controller

index(m)	Name	Description
9	GCLK_EVSYN_CHANNEL_3	EVSYN_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

23.8.11 DFLLULP Read Request

Name: DFLLULPRREQ
Offset: 0x1F
Reset: 0x00
Property: -

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

Bit 7 – RREQ Read Request

Writing a zero to this bit has no effect.

Writing a one to this bit requests synchronization of the DFLLULPDLY register with the current oscillator delay value and sets the Delay Busy bit in the Synchronization Busy register (DFLLULPSYNCBUSY.DELAY).

This bit is cleared automatically when synchronization is complete.

25.6.2.3 Selecting a Voltage Reference

The Voltage Reference Selection bit field in the VREF register (VREF.SEL) selects the voltage of INTREF to be applied to analog modules, e.g. the ADC.

25.6.2.4 Sleep Mode Operation

The Voltage Reference output and the Temperature Sensor output behavior during sleep mode can be configured using the Run in Standby bit and the On Demand bit in the Voltage Reference register (VREF.RUNSTDBY, VREF.ONDEMAND), see the following table:

Table 25-2. VREF Sleep Mode Operation

VREF.ONDEMAND	VREF.RUNSTDBY	Voltage Reference Sleep behavior
-	-	Disable
0	0	Always run in all sleep modes <i>except</i> standby sleep mode
0	1	Always run in all sleep modes <i>including</i> standby sleep mode
1	0	Only run if requested by the ADC, in all sleep modes <i>except</i> standby sleep mode
1	1	Only run if requested by the ADC, in all sleep modes <i>including</i> standby sleep mode

25.6.3 Brown-Out Detectors

25.6.3.1 Initialization

Before a Brown-Out Detector (BOD33) is enabled, it must be configured, as outlined by the following:

- Set the BOD threshold level (BOD33.LEVEL)
- Set the configuration in Active, Standby (BOD33.ACTION, BOD33.STDBYCFG)
- Set the prescaling value if the BOD will run in sampling mode (BOD33.PSEL)
- Set the action and hysteresis (BOD33.ACTION and BOD33.HYST)

The BOD33 register is Enable-Protected, meaning that they can only be written when the BOD is disabled (BOD33.ENABLE=0 and STATUS.B33SRDY=0). As long as the Enable bit is '1', any writes to Enable-Protected registers will be discarded, and an APB error will be generated. The Enable bits are not Enable-Protected.

25.6.3.2 Enabling, Disabling, and Resetting

After power or user reset, the BOD33 and BOD12 register values are loaded from the NVM User Page.

The BOD33 is enabled by writing a '1' to the Enable bit in the BOD control register (BOD33.ENABLE). The BOD33 is disabled by writing a '0' to the BOD33.ENABLE.

25.6.3.3 3.3V Brown-Out Detector (BOD33)

The 3.3V Brown-Out Detector (BOD33) is able to monitor the VDD supply and compares the voltage with the brown-out threshold level set in the BOD33 Level field (BOD33.LEVEL) in the BOD33 register.

When VDD crosses below the brown-out threshold level, the BOD33 can generate either an interrupt or a Reset, depending on the BOD33 Action bit field (BOD33.ACTION).

The BOD33 detection status can be read from the BOD33 Detection bit in the Status register (STATUS.BOD33DET).

At start-up or at Power-On Reset (POR), the BOD33 register values are loaded from the NVM User Row.

The counter value is continuously compared with the 16-bit Compare registers (COMPn, n=0..1). When a compare match occurs, the Compare n Interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.CMPn, n=0..1) is set on the next 0-to-1 transition of CLK_RTC_CNT.

27.6.2.5 Clock/Calendar (Mode 2)

When the RTC Operating Mode bits in the Control A register (CTRLA.MODE) are written to 0x2, the counter operates in Clock/Calendar mode, as shown in [Figure 27-3](#). When the RTC is enabled, the counter will increment on every 0-to-1 transition of CLK_RTC_CNT. The selected clock source and RTC prescaler must be configured to provide a 1Hz clock to the counter for correct operation in this mode.

The time and date can be read from or written to the Clock Value register (CLOCK) in a 32-bit time/date format. Time is represented as:

- Seconds
- Minutes
- Hours

Hours can be represented in either 12- or 24-hour format, selected by the Clock Representation bit in the Control A register (CTRLA.CLKREP). This bit can be changed only while the RTC is disabled.

The date is represented in this form:

- Day as the numeric day of the month (starting at 1)
- Month as the numeric month of the year (1 = January, 2 = February, etc.)
- Year as a value from 0x00 to 0x3F. This value must be added to a user-defined reference year. The reference year must be a leap year (2016, 2020 etc). Example: the year value 0x2D, added to a reference year 2016, represents the year 2061.

The RTC will increment until it reaches the top value of 23:59:59 December 31 of year value 0x3F, and then wrap to 00:00:00 January 1 of year value 0x00. This will set the Overflow Interrupt flag in the Interrupt Flag Status and Clear registers (INTFLAG.OVF).

The clock value is continuously compared with the 32-bit Alarm register (ALARM0). When an alarm match occurs, the Alarm 0 Interrupt flag in the Interrupt Flag Status and Clear registers (INTFLAG.ALARM0) is set on the next 0-to-1 transition of CLK_RTC_CNT. E.g. For a 1Hz clock counter, it means the Alarm 0 Interrupt flag is set with a delay of 1s after the occurrence of alarm match.

A valid alarm match depends on the setting of the Alarm Mask Selection bits in the Alarm 0 Mask register (MASK0.SEL). These bits determine which time/date fields of the clock and alarm values are valid for comparison and which are ignored.

If the Clear on Match bit in the Control A register (CTRLA.MATCHCLR) is set, the counter is cleared on the next counter cycle when an alarm match with ALARM0 occurs. This allows the RTC to generate periodic interrupts or events with longer periods than it would be possible with the prescaler events only (see [27.6.8.1 Periodic Intervals](#)).

Note: When CTRLA.MATCHCLR is 1, INTFLAG.ALARM0 and INTFLAG.OVF will both be set simultaneously on an alarm match with ALARM0.

27.6.3 DMA Operation

The RTC generates the following DMA request:

- Tamper (TAMPER): The request is set on capture of the timestamp. The request is cleared when the Timestamp register is read.

ongoing one is done. Only one pending transfer can be kept per channel. If the trigger source generates more transfer requests while one is already pending, the additional ones will be lost. All channels pending status flags are also available in the Pending Channels register (PENDCH).

When the transfer starts, the corresponding Channel Busy status flag is set in Channel Status register (CHSTATUS.BUSY). When the trigger action is complete, the Channel Busy status flag is cleared. All channel busy status flags are also available in the Busy Channels register (BUSYCH) in DMAC.

28.6.2.7 Addressing

Each block transfer needs to have both a source address and a destination address defined. The source address is set by writing the Transfer Source Address (SRCADDR) register, the destination address is set by writing the Transfer Destination Address (DSTADDR) register.

The addressing of this DMAC module can be static or incremental, for either source or destination of a block transfer, or both.

Incrementation for the source address of a block transfer is enabled by writing the Source Address Incrementation Enable bit in the Block Transfer Control register (BTCTRL.SRCINC=1). The step size of the incrementation is configurable and can be chosen by writing the Step Selection bit in the Block Transfer Control register (BTCTRL.STEPSEL=1) and writing the desired step size in the Address Increment Step Size bit group in the Block Transfer Control register (BTCTRL.STEPSIZE). If BTCTRL.STEPSEL=0, the step size for the source incrementation will be the size of one beat.

When source address incrementation is configured (BTCTRL.SRCINC=1), SRCADDR is calculated as follows:

If BTCTRL.STEPSEL=1:

$$\text{SRCADDR} = \text{SRCADDR}_{\text{START}} + \text{BTCNT} \cdot (\text{BEATSIZE} + 1) \cdot 2^{\text{STEPSIZE}}$$

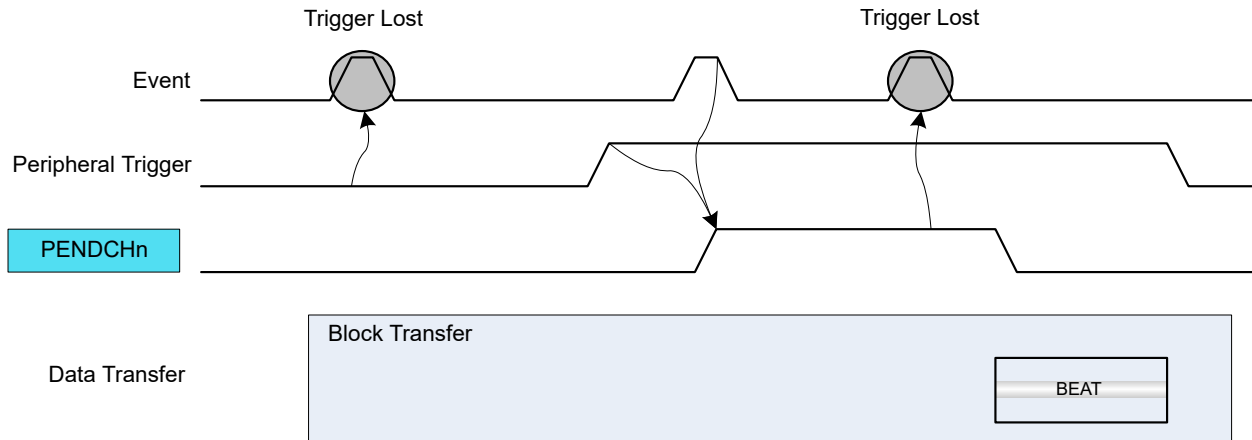
If BTCTRL.STEPSEL=0:

$$\text{SRCADDR} = \text{SRCADDR}_{\text{START}} + \text{BTCNT} \cdot (\text{BEATSIZE} + 1)$$

- SRCADDR_{START} is the source address of the first beat transfer in the block transfer
- BTCNT is the initial number of beats remaining in the block transfer
- BEATSIZE is the configured number of bytes in a beat
- STEPSIZE is the configured number of beats for each incrementation

The following figure shows an example where DMA channel 0 is configured to increment the source address by one beat after each beat transfer (BTCTRL.SRCINC=1), and DMA channel 1 is configured to increment the source address by two beats (BTCTRL.SRCINC=1, BTCTRL.STEPSEL=1, and BTCTRL.STEPSIZE=0x1). As the destination address for both channels are peripherals, destination incrementation is disabled (BTCTRL.DSTINC=0).

Figure 28-12. Periodic Event with Beat Peripheral Triggers



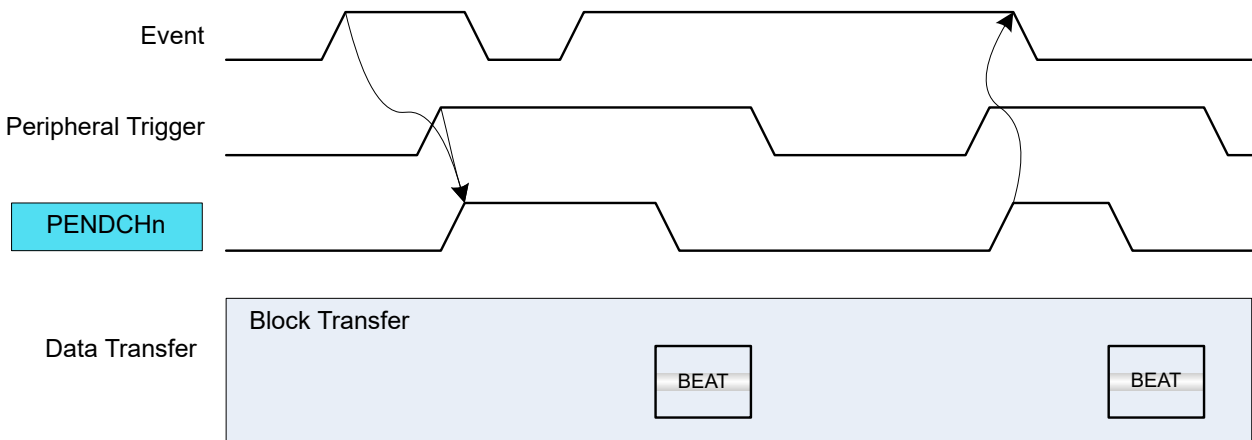
Conditional Transfer

The event input is used to trigger a conditional transfer on peripherals with pending transfer requests. As an example, this type of event can be used for peripheral-to-peripheral transfers, where one peripheral is the source of event and the second peripheral is the source of the trigger.

Each peripheral trigger is stored internally when the event is received. When the peripheral trigger is stored internally, the Channel Pending status bit is set ([CHSTATUS.PEND](#)), the respective Pending Channel n Bit in the Pending Channels register is set ([28.8.13 PENDCH.PENDCHn](#)), and the event is acknowledged. A software trigger will now trigger a transfer.

The figure below shows an example where conditional event is enabled with peripheral beat trigger requests.

Figure 28-13. Conditional Event with Beat Peripheral Triggers



Conditional Block Transfer

The event input is used to trigger a conditional block transfer on peripherals.

Before starting transfers within a block, an event must be received. When received, the event is acknowledged when the block transfer is completed. A software trigger will trigger a transfer.

The figure below shows an example where conditional event block transfer is started with peripheral beat trigger requests.

Each pin may be secured or non-secured, with secured pins only accessible by secure accesses.

Device-specific configurations may cause some lines (and the corresponding Pxy pin) not to be implemented.

32.5.2 Power Management

During Reset, all PORT lines are configured as inputs with input buffers, output buffers and pull disabled.

The PORT peripheral will continue operating in any sleep mode where its source clock is running.

32.5.3 Clocks

The PORT bus clock (CLK_PORT_APB) can be enabled and disabled in the Main Clock module, and the default state of CLK_PORT_APB can be found in the *Peripheral Clock Masking* section in *MCLK – Main Clock*.

The PORT requires an APB clock, which may be divided from the CPU main clock and allows the CPU to access the registers of PORT through the high-speed matrix and the AHB/APB bridge.

The priority of IOBUS accesses is higher than APB accesses. One clock cycle latency can be observed on the APB access in case of concurrent PORT accesses.

Related Links

[19. MCLK – Main Clock](#)

32.5.4 DMA

Not applicable.

32.5.5 Interrupts

Not applicable.

32.5.6 Events

The events of this peripheral are connected to the Event System.

Related Links

[33. EVSYS – Event System](#)

32.5.7 Debug Operation

When the CPU is halted in debug mode, this peripheral will continue normal operation.

32.5.8 Register Access Protection

All registers with write-access can be optionally write-protected by the Peripheral Access Controller (PAC).

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

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

Related Links

[15. PAC - Peripheral Access Controller](#)

32.5.9 Analog Connections

Analog functions are connected directly between the analog blocks and the I/O pads using analog buses. However, selecting an analog peripheral function for a given pin will disable the corresponding digital features of the pad.

32.8.8 Data Output Value Toggle

Name: OUTTGL
Offset: 0x1C
Reset: 0x00000000
Property: PAC Write-Protection, Mix-Secure



Important: For **SAM L11 Non-Secure** accesses, read and write accesses (RW*) are allowed only if the security attribution for the corresponding I/O pin is set as Non-Secured in the NONSEC register.

This register allows the user to toggle the drive level of one or more output I/O pins, without doing a read-modify-write operation. Changes in this register will also be reflected in the Data Output Value (OUT), Data Output Value Set (OUTSET) and Data Output Value Clear (OUTCLR) registers.



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

Bit	31	30	29	28	27	26	25	24
	OUTTGL[31:24]							
Access	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	OUTTGL[23:16]							
Access	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	OUTTGL[15:8]							
Access	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	OUTTGL[7:0]							
Access	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW	RW/RW*/RW
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – OUTTGL[31:0] PORT Data Output Value Toggle
 Writing '0' to a bit has no effect.

33. EVSYS – Event System

33.1 Overview

The Event System (EVSYS) allows autonomous, low-latency and configurable communication between peripherals.

Several peripherals can be configured to generate and/or respond to signals known as events. The exact condition to generate an event, or the action taken upon receiving an event, is specific to each peripheral. Peripherals that respond to events are called event users. Peripherals that generate events are called event generators. A peripheral can have one or more event generators and can have one or more event users. Channels and event users can be defined as secured or non-secured, where secured channels or event users can only be handled by secure code.

Communication is made without CPU intervention and without consuming system resources such as bus or RAM bandwidth. This reduces the load on the CPU and other system resources, compared to a traditional interrupt-based system.

33.2 Features

- 8 configurable event channels:
 - All channels can be connected to any event generator
 - All channels provide a pure asynchronous path
 - 4 channels (CHANNEL0 to CHANNEL3) provide a resynchronized or synchronous path using their dedicated generic clock (GCLK_EVSYS_CHANNEL_n)
- 49 event generators.
- 23 event users.
- Configurable edge detector.
- Peripherals can be event generators, event users, or both.
- SleepWalking and interrupt for operation in sleep modes.
- Software event generation.
- Each event user can choose which channel to respond to.
- Optional Static or Round-Robin interrupt priority arbitration.
- Each channel and each event user can be configured as secured or non-secured (**SAM L11**).

33.7.3 Priority Control

Name: PRICTRL
Offset: 0x08
Reset: 0x00
Property: PAC Write-Protection, Secure

	Bit	7	6	5	4	3	2	1	0
		RREN						PRI[1:0]	
Access		RW/RW/RW						RW/-RW	RW/-RW
Reset		0						0	0

Bit 7 – RREN Round-Robin Scheduling Enable

For details on scheduling schemes, refer to [Interrupt Status and Interrupts Arbitration](#)

Value	Description
0	Static scheduling scheme for channels with level priority
1	Round-robin scheduling scheme for channels with level priority

Bits 1:0 – PRI[1:0] Channel Priority Number

When round-robin arbitration is enabled (PRICTRL.RREN=1) for priority level, this register holds the channel number of the last EVSYS channel being granted access as the active channel with priority level. The value of this bit group is updated each time the INTPEND or any of CHINTFLAG registers are written.

When static arbitration is enabled (PRICTRL.RREN=0) for priority level, and the value of this bit group is nonzero, it will not affect the static priority scheme.

This bit group is not reset when round-robin scheduling gets disabled (PRICTRL.RREN written to zero).

33.7.12 Channel n Status

Name: CHSTATUSn
Offset: 0x27 + n*0x08 [n=0..7]
Reset: 0x01
Property: Mix-Secure



Important: For **SAM L11 Non-Secure** accesses, read accesses (R*) are allowed only if the security attribution for the corresponding channel (CHANNELx) is set as Non-Secured in the NONSECCHAN register.

	7	6	5	4	3	2	1	0
Access							R/R*/R	R/R*/R
Reset							0	0

Bit 1 – BUSYCH Busy Channel

This bit is cleared when channel is idle.

This bit is set if an event on channel has not been handled by all event users connected to channel.

When the event channel path is asynchronous, this bit is always read '0'.

Bit 0 – RDYUSR Ready User

This bit is cleared when at least one of the event users connected to the channel is not ready.

This bit is set when all event users connected to channel are ready to handle incoming events on the channel.

When the event channel path is asynchronous, this bit is always read zero.

35.8.5 Receive Pulse Length Register

Name: RXPL
Offset: 0x0E
Reset: 0x00
Property: Enable-Protected, PAC Write-Protection

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

Bits 7:0 – RXPL[7:0] Receive Pulse Length

When the encoding format is set to IrDA (CTRLB.ENC=1), these bits control the minimum pulse length that is required for a pulse to be accepted by the IrDA receiver with regards to the serial engine clock period SE_{per} .

$$PULSE \geq (RXPL + 2) \cdot SE_{per}$$

SAM L10/L11 Family

SERCOM SPI – SERCOM Serial Peripheral Interface

Bit 0 – SWRST Software Reset

Writing '0' to this bit has no effect.

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

Writing "1" to CTRLA.SWRST will always take precedence, meaning that all other writes in the same write-operation will be discarded. Any register write access during the ongoing reset will result in an APB error. Reading any register will return the reset value of the register.

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

This bit is not enable-protected.

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

SAM L10/L11 Family

SERCOM I2C – SERCOM Inter-Integrated Circ...

Writing '1' to CTRLA.SWRST will always take precedence, meaning that all other writes in the same write-operation will be discarded. Any register write access during the ongoing reset will result in an APB error. Reading any register will return the reset value of the register.

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

This bit is not enable-protected.

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

38.7.3.12 Synchronization Busy

Name: SYNCBUSY
Offset: 0x10
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
		CCx	PER	COUNT	STATUS	CTRLB	ENABLE	SWRST
Access		R	R	R	R	R	R	R
Reset		0	0	0	0	0	0	0

Bit 6 – CCx Compare/Capture Channel x Synchronization Busy
 For details on CC channels number, refer to each TC feature list.

This bit is set when the synchronization of CCx between clock domains is started.

This bit is also set when the CCBUFx is written, and cleared on update condition. The bit is automatically cleared when the STATUS.CCBUFx bit is cleared.

Bit 5 – PER PER Synchronization Busy

This bit is cleared when the synchronization of PER between the clock domains is complete.

This bit is set when the synchronization of PER between clock domains is started.

This bit is also set when the PER is written, and cleared on update condition. The bit is automatically cleared when the STATUS.PERBUF bit is cleared.

Bit 4 – COUNT COUNT Synchronization Busy

This bit is cleared when the synchronization of COUNT between the clock domains is complete.

This bit is set when the synchronization of COUNT between clock domains is started.

Bit 3 – STATUS STATUS Synchronization Busy

This bit is cleared when the synchronization of STATUS between the clock domains is complete.

38.7.3.15 Channel x Compare/Capture Value, 32-bit Mode

Name: CCx
Offset: 0x1C + x*0x04 [x=0..1]
Reset: 0x00000000
Property: Write-Synchronized

Bit	31	30	29	28	27	26	25	24
	CC[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
	CC[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
	CC[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
	CC[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 – CC[31:0] Channel x Compare/Capture Value

These bits contain the compare/capture value in 32-bit TC mode. In Match frequency (MFRQ) or Match PWM (MPWM) waveform operation (WAVE.WAVEGEN), the CC0 register is used as a period register.

39. TRNG – True Random Number Generator

39.1 Overview

The True Random Number Generator (TRNG) generates unpredictable random numbers that are not generated by an algorithm. It passes the American NIST Special Publication 800-22 and Diehard Random Tests Suites.

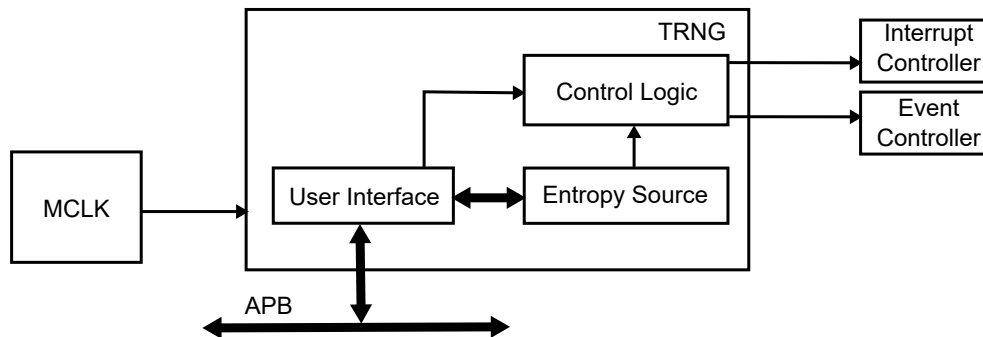
The TRNG may be used as an entropy source for seeding an NIST approved DRNG (Deterministic RNG) as required by FIPS PUB 140-2 and 140-3.

39.2 Features

- Passed NIST Special Publication 800-22 Tests Suite
- Passed Diehard Random Tests Suite
- May be used as Entropy Source for seeding an NIST approved DRNG (Deterministic RNG) as required by FIPS PUB 140-2 and 140-3
- Provides a 32-bit random number every 84 clock cycles

39.3 Block Diagram

Figure 39-1. TRNG Block Diagram.



39.4 Signal Description

Not applicable.

39.5 Product Dependencies

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

39.5.1 I/O Lines

Not applicable.

39.5.2 Power Management

The functioning of TRNG depends on the sleep mode of device.

42.5.2 Power Management

The AC will continue to operate in any sleep mode where the selected source clock is running. The AC's interrupts can be used to wake up the device from sleep modes. Events connected to the event system can trigger other operations in the system without exiting sleep modes.

Related Links

[22. PM – Power Manager](#)

42.5.3 Clocks

The AC bus clock (CLK_AC_APB) can be enabled and disabled in the Main Clock module, MCLK (see *MCLK - Main Clock*, and the default state of CLK_AC_APB can be found in *Peripheral Clock Masking*.

A generic clock (GCLK_AC) is required to clock the AC. This clock must be configured and enabled in the generic clock controller before using the AC. Refer to the Generic Clock Controller chapter for details.

This generic clock is asynchronous to the bus clock (CLK_AC_APB). Due to this asynchronicity, writes to certain registers will require synchronization between the clock domains. Refer to [Synchronization](#) for further details.

Related Links

[22. PM – Power Manager](#)

42.5.4 DMA

Not applicable.

42.5.5 Interrupts

The interrupt request lines are connected to the interrupt controller. Using the AC interrupts requires the interrupt controller to be configured first. Refer to *Nested Vector Interrupt Controller* for details.

42.5.6 Events

The events are connected to the Event System. Refer to *EVSYS – Event System* for details on how to configure the Event System.

Related Links

[33. EVSYS – Event System](#)

42.5.7 Debug Operation

When the CPU is halted in debug mode, the AC will halt normal operation after any on-going comparison is completed. The AC can be forced to continue normal operation during debugging. Refer to [DBGCTRL](#) for details. If the AC is configured in a way that requires it to be periodically serviced by the CPU through interrupts or similar, improper operation or data loss may result during debugging.

42.5.8 Register Access Protection

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

- Control B register (CTRLB)
- 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.