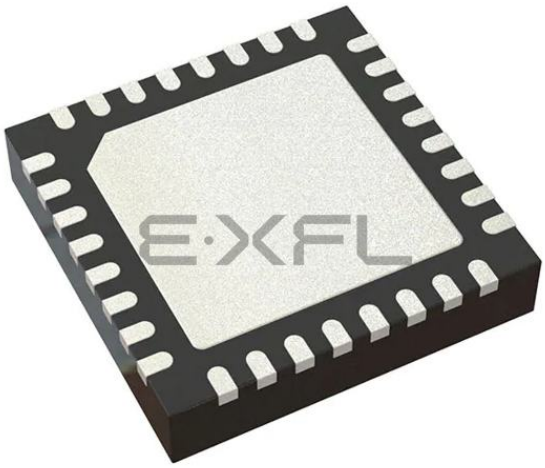


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

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

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	ARM® Cortex®-M23
Core Size	32-Bit Single-Core
Speed	32MHz
Connectivity	I²C, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, DMA, POR, PWM, WDT
Number of I/O	25
Program Memory Size	16KB (16K x 8)
Program Memory Type	FLASH
EEPROM Size	2K x 8
RAM Size	4K x 8
Voltage - Supply (Vcc/Vdd)	1.62V ~ 3.63V
Data Converters	A/D 10x12b; D/A 1x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°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/atsaml10e14a-mft

Table 4-4. Serial Wire Debug Interface Pinout

Signal	Supply	I/O pin
SWCLK	VDDIO	PA30
SWDIO	VDDIO	PA31

4.4 SERCOM Configurations

The following table lists the supported features for each SERCOM instance:

Table 4-5. SERCOM Features Summary

	SERCOM Instance		
Protocol	**SERCOM0**	**SERCOM1**	**SERCOM2**
SPI	Yes	Yes	Yes
I^2C (1)	Yes High-speed mode (≤ 3,4 Mbit/s)	Yes Fast plus Mode (≤ 1 Mbit/s)	No
USART	Yes including: Hardware Handshaking IrDA	Yes including: Hardware Handshaking IrDA	Yes including: Hardware Handshaking IrDA RS-485 Auto-baud mode LIN Slave ISO7816
USART/SPI Receive Buffer Size	Two-level	Four-level	Two-level
Secure Pin Multiplexing (**SAM L11 only**)	No	Yes	No

Note:

1. I^2C is not supported on all SERCOM pins. Refer to the SERCOM I^2C Pins table for more details.

4.4.1 SERCOM I2C Pins

The following table lists the SERCOM pins which support I^2C:

Table 4-6. SERCOM I^2C Pins

Pin Name	SERCOM0 I^2C Pad Name	SERCOM1 I^2C Pad Name
PA16	SERCOM0/PAD[0]	SERCOM1/PAD[0]
PA17	SERCOM0/PAD[1]	SERCOM1/PAD[1]
PA22	SERCOM0/PAD[0]	N/A
PA23	SERCOM0/PAD[1]	N/A

4.4.2 Secure Pin Multiplexing (on SERCOM) Pins

The Secure Pin Multiplexing feature can be used on dedicated SERCOM I/O pins to isolate a secure communication with external devices from the non-secure application.

Refer to 13.6 Secure Pin Multiplexing on SERCOM for more details.

The following table lists the SERCOM pins that support the Secure Pin Multiplexing feature:

14.4.5.7 Random Session Key Generation (CMD_DCEK) - SAM L11 only

This command allows using a challenge-response scheme to prevent exposure of the keys in clear text on the debugger communication lines.

The different keys sent by the debugger during the Boot ROM for Chip Erase (CMD_CEx) and CRC (CMD_CRC) commands execution are:

- CRCKEY for CMD_CRC command
- CEKEYx for CMD_CEx commands

Note: The CMD_DCEK command has no effect on the SAM L10, the key derivation will not be enabled.

The random challenge value is generated using the TRNG of the device. It is generated once the CMD_DCEK is received and communicated to the debugger.

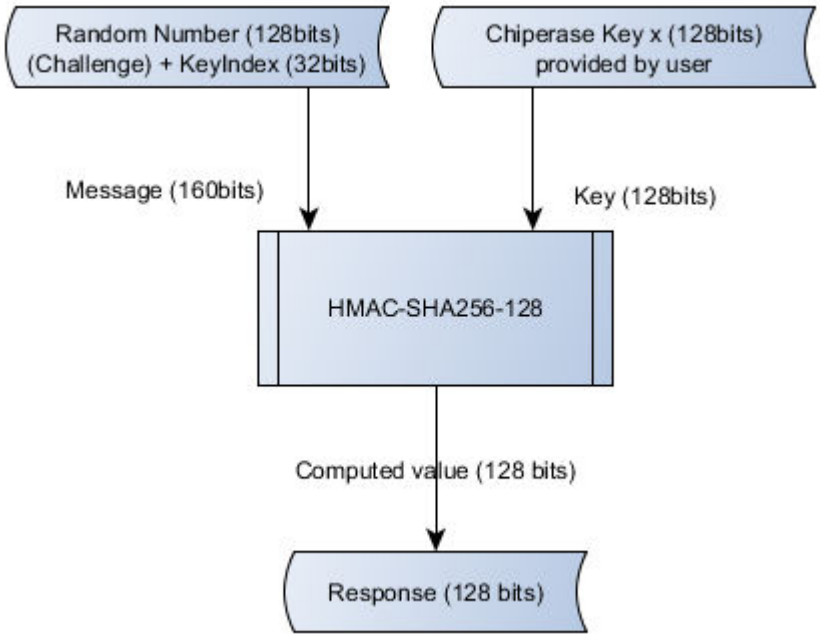
The next CMD_CEx or CMD_CRC commands will expect the key value to be replaced by the computed response corresponding to the challenge.

The challenge value is valid only for the next CMD_CEx/ CMD_CRC command.

Before sending a new CMD_CEx/ CMD_CRC command, a CMD_DCEK shall be used to re-enable the challenge-response scheme a get a new challenge value.

On the debugger side, the response shall be computed using the following algorithm:

Figure 14-13. Debugger Algorithm



Where KeyIndex is:

- 0 for ChipErase_NS
- 1 for ChipErase_S
- 2 for ChipErase_ALL
- 3 for CRC Command

Note:

Field	Size	Description	Location
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)	PID2

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

16.9.2 Chip Identification Method

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

- Processor identification
- Product family identification
- Product series identification
- Device select

16.10 Functional Description

16.10.1 Principle of Operation

The DSU provides memory services, such as CRC32 or MBIST that require almost the same interface. Hence, the Address, Length and Data registers (ADDR, LENGTH, DATA) are shared. These shared registers must be configured first; then a command can be issued by writing the Control register. When a command is ongoing, other commands are discarded until the current operation is completed. Hence, the user must wait for the STATUSA.DONE bit to be set prior to issuing another one.

16.10.2 Basic Operation

16.10.2.1 Initialization

The module is enabled by enabling its clocks. For more details, refer to 16.5.3 Clocks. The DSU registers can be PAC write-protected.

Related Links

15. PAC - Peripheral Access Controller

16.10.2.2 Operation From a Debug Adapter

Debug adapters should access the DSU registers in the external address range [0x100 – 0x1FFF].

If STATUSB.DAL is equal to 0x0, accessing the first 0x100 bytes causes the DSU security filter to return an error to the DAP.

(**SAM L11 only**): If STATUSB.DAL is equal to 0x1, debug accesses will go through the DSU security filter but will be forced as non-secure, therefore the DSU internal address space will not be accessible and any access in this case is discarded (writes are ignored, reads return 0) and raise STATUSA.PERR.

Writing in this bitfield has no effect.

16.12.4 Address

Name: ADDR
Offset: 0x0004
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
	ADDR[29:22]							
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
	ADDR[21:14]							
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
	ADDR[13:6]							
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
	ADDR[5:0]						AMOD[1:0]	
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 31:2 – ADDR[29:0] Address
Initial word start address needed for memory operations.

Bits 1:0 – AMOD[1:0] Address Mode
The functionality of these bits is dependent on the operation mode.

Bit description when testing on-16.10.6 Testing of On-Board Memories MBISTboard memories (MBIST): refer to

The DMAC is configured to operate in standby sleep mode by using its respective RUNSTDBY bit. A DMAC channel is configured to set the DMA destination. The Run in Standby bit of this DMAC channel is written to '1' to allow it running in Standby Sleep mode.

Entering Standby mode: The Power Manager peripheral sets PDSW to retention state. The VDDCORE is supplied by the low-power regulator.

Dynamic SleepWalking: based on RTC conditions, an RTC output signal (DMA request for timestamp) triggers DMAC to put timestamp value at configured DMA destination.

This event is detected by the Power Manager which sets the PDSW power domain to active state and starts the main voltage regulator.

This DMA transfer request is detected by the PM, which sets PDSW (containing the DMAC) to active state. The DMAC requests the CLK_DMAC_AHB clock and transfer the timestamp value to the memory. When the DMA beat transfer is completed, the CLK_DMAC_AHB clock is stopped again.

The low-power regulator starts again and the PDSW power domain is set back to retention state by the PM. Note that during this dynamic SleepWalking period, the CPU is still sleeping.

Exiting Standby mode: during SleepWalking with Dynamic Power Gating sequence, if conditions are met, the DMAC generates an interrupt to wake up the device.

Related Links

27. RTC – Real-Time Counter
33. EVSYS – Event System

22.6.7 DMA Operation

Not applicable.

22.6.8 Interrupts

The peripheral has the following interrupt sources:

- Performance Level Ready (PLRDY)
  This interrupt is a synchronous wake-up source. See Table 22-1 for details.

Each interrupt source has an interrupt flag associated with it. The interrupt flag in the Interrupt Flag Status and Clear (INTFLAG) register is set when the interrupt condition occurs. Each interrupt can be individually enabled by writing a '1' to the corresponding bit in the Interrupt Enable Set (INTENSET) register, and disabled by writing a '1' to the corresponding bit in the Interrupt Enable Clear (INTENCLR) register.

An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until the interrupt flag is cleared, the interrupt is disabled or the peripheral is reset.

An interrupt flag is cleared by writing a '1' to the corresponding bit in the INTFLAG register. Each peripheral can have one interrupt request line per interrupt source or one common interrupt request line for all the interrupt sources. Refer to the Nested Vector Interrupt Controller (NVIC) for details. If the peripheral has one common interrupt request line for all the interrupt sources, the user must read the INTFLAG register to determine which interrupt condition is present.

22.6.9 Events

Not applicable.

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

27.8 Register Description - Mode 0 - 32-Bit Counter

This Register Description section is valid if the RTC is in COUNT32 mode (CTRLA.MODE=0).

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

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

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

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

On **SAM L11** devices, this peripheral has different access permissions depending on PAC Security Attribution (Secure or Non-Secure):

- If the peripheral is configured as Non-Secure in the PAC:
  - Secure access and Non-Secure access are granted
- If the peripheral is configured as Secure in the PAC:
  - Secure access is granted
  - Non-Secure access is discarded (Write is ignored, read 0x0) and a PAC error is triggered

Refer to *Peripherals Security Attribution* for more information.

28. DMAC – Direct Memory Access Controller

28.1 Overview

The Direct Memory Access Controller (DMAC) contains both a Direct Memory Access engine and a Cyclic Redundancy Check (CRC) engine. The DMAC can transfer data between memories and peripherals, and thus off-load these tasks from the CPU. It enables high data transfer rates with minimum CPU intervention, and frees up CPU time. With access to all peripherals, the DMAC can handle automatic transfer of data between communication modules.

The DMA part of the DMAC has several DMA channels which all can receive different types of transfer triggers to generate transfer requests from the DMA channels to the arbiter, see also the Block Diagram. The arbiter will grant one DMA channel at a time to act as the active channel. When an active channel has been granted, the fetch engine of the DMAC will fetch a transfer descriptor from the SRAM and store it in the internal memory of the active channel, which will execute the data transmission.

An ongoing data transfer of an active channel can be interrupted by a higher prioritized DMA channel. The DMAC will write back the updated transfer descriptor from the internal memory of the active channel to SRAM, and grant the higher prioritized channel to start transfer as the new active channel. Once a DMA channel is done with its transfer, interrupts and events can be generated optionally.

The DMAC has four bus interfaces:

- The *data transfer bus* is used for performing the actual DMA transfer.
- The *AHB/APB Bridge bus* is used when writing and reading the I/O registers of the DMAC.
- The *descriptor fetch bus* is used by the fetch engine to fetch transfer descriptors before data transfer can be started or continued.
- The *write-back bus* is used to write the transfer descriptor back to SRAM.

All buses are AHB master interfaces but the AHB/APB Bridge bus, which is an APB slave interface.

The CRC engine can be used by software to detect an accidental error in the transferred data and to take corrective action, such as requesting the data to be sent again or simply not using the incorrect data.

28.2 Features

- Data transfer from:
  - Peripheral to peripheral
  - Peripheral to memory
  - Memory to peripheral
  - Memory to memory
- Transfer trigger sources
  - Software
  - Events from Event System
  - Dedicated requests from peripherals
- SRAM based transfer descriptors
  - Single transfer using one descriptor
  - Multi-buffer or circular buffer modes by linking multiple descriptors

30.8.6 Interrupt Enable Set

Name: INTENSET
Offset: 0x10
Reset: 0x00
Property: PAC Write-Protection, Write-Mix-Secure

Important: For **SAM L11 Non-Secure** accesses, write accesses (W*) are allowed only if Non-Secure Write is set in the NONSEC register.

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

Bit	7	6	5	4	3	2	1	0
			NSCHK	KEYE	NVME	LOCKE	PROGE	DONE
Access			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

Bit 5 – NSCHK Non-secure Check Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit sets the NSCHK interrupt enable.

This bit will read as the current value of the NSCHK interrupt enable.

Bit 4 – KEYE Key Error Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit sets the KEYE interrupt enable.

This bit will read as the current value of the KEYE interrupt enable.

Bit 3 – NVME NVM internal Error Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit sets the NVME interrupt enable.

This bit will read as the current value of the NVME interrupt enable.

Bit 2 – LOCKE Lock Error Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit sets the LOCKE interrupt enable.

This bit will read as the current value of the LOCKE interrupt enable.

Bit 1 – PROGE Programming Error Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit sets the PROGE interrupt enable.

This bit will read as the current value of the PROGE interrupt enable.

PRM can be cleared in three ways: through AHB access to the NVM block, through the command interface (SPRM and CPRM) or when exiting sleep with SLEEPPRM set accordingly.

Value	Description
0	NVM is not in power reduction mode.
1	NVM is in power reduction mode.

30.8.10 Secure Region Unlock Bits

Name: SULCK
Offset: 0x20
Reset: x initially determined from NVM User Row after reset
Property: PAC Write-Protection, Write-Secure

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

Bit	15	14	13	12	11	10	9	8
	SLKEY[7:0]							
Access	W/-/W	W/-/W	W/-/W	W/-/W	W/-/W	W/-/W	W/-/W	W/-/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
						DS	AS	BS
Access						RW/R/RW	RW/R/RW	RW/R/RW
Reset						x	x	x

Bits 15:8 – SLKEY[7:0] Secure Unlock Key
When this bit group is written to the key value 0xA5, the write will be performed. If a value different from the key value is tried, the write will be discarded and INTFLAG.KEYE set.

Bit 2 – DS DATA Secure Unlock Bit
Default state after erase will be unlocked (0x1).

Value	Description
0	The DS region is locked.
1	The DS region is not locked.

Bit 1 – AS Application Secure Unlock Bit
Default state after erase will be unlocked (0x1).

Value	Description
0	The AS region is locked.
1	The AS region is not locked.

Bit 0 – BS BOOT Secure Unlock Bit
Default state after erase will be unlocked (0x1).

Value	Description
0	The BS region is locked.
1	The BS region is not locked.

The tamper full erase routine operates at the highest priority. If a remanence routine executing when a tamper full erase occurs, the remanence routine is immediately terminated. If the CPU attempts to write a new scramble key at the same time the tamper key erase routine is active, the CPU data is ignored, but no bus error will occur. If a CPU security routine access is requested during a tamper full erase, the CPU transaction will be ignored and treated as a bus error similar to accessing the module during a software reset.

Important: In STANDBY low power mode, it is mandatory to enable the dynamic power gating feature (STDBYCFG.DPGPDSW) to ensure TrustRAM erasing when the power domain PDSW is in a retention state.

31.6.3 Interrupts

The TRAM has the following interrupt sources:

- Data Remanence Prevention (DRP): Indicates that the data remanence prevention routine has ended.
- Data Read Error (ERR): Indicates when there is a RAM readout error.

Each interrupt source has an interrupt flag associated with it. The interrupt flag in the Interrupt Flag Status and Clear (INTFLAG) register is set when the interrupt condition occurs. Each interrupt can be individually enabled by writing a one to the corresponding bit in the Interrupt Enable Set (INTENSET) register, and disabled by writing a one to the corresponding bit in the Interrupt Enable Clear (INTENCLR) register.

An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until the interrupt flag is cleared, the interrupt is disabled, or the TRAM is reset. See 22.8.6 INTFLAG for details on how to clear interrupt flags. All interrupt requests from the peripheral are ORed together on system level to generate one combined interrupt request to the NVIC. Refer to *Nested Vector Interrupt Controller* for details. The user must read the INTFLAG register to determine which interrupt condition is present.

Note that interrupts must be globally enabled for interrupt requests to be generated. Refer to *Nested Vector Interrupt Controller* for details.

31.6.4 Sleep Mode Operation

The TRAM continues to operate during sleep. When it receives events from the Event System, it will request its own clock in order to perform the requested operation.

An interrupt request will be generated after the wake-up if the Interrupt Controller is configured accordingly. Otherwise the CPU will wake up directly, without triggering an interrupt. In this case, the CPU will continue executing from the instruction following the entry into sleep.

The periodic events can also wake up the CPU through the interrupt function of the Event System. In this case, the event must be enabled and connected to an event channel with its interrupt enabled. See *EVSYS – Event System* for more information.

31.6.5 Synchronization

Due to the asynchronicity between event sources and CLK_TRAM_APB some registers must be synchronized when accessed. A register can require:

- Synchronization when written

Bit	31	30	29	28	27	26	25	24
	OUT[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
	OUT[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
	OUT[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
	OUT[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 – OUT[31:0] PORT Data Output Value
For pins configured as outputs via the Data Direction register (DIR), these bits set the logical output drive level.

For pins configured as inputs via the Data Direction register (DIR) and with pull enabled via the Pull Enable bit in the Pin Configuration register (PINCFG.PULLEN), these bits will set the input pull direction.

Value	Description
0	The I/O pin output is driven low, or the input is connected to an internal pull-down.
1	The I/O pin output is driven high, or the input is connected to an internal pull-up.

This functionality is automatically configured, depending on the selected operating mode.

Figure 34-3. Baud Rate Generator

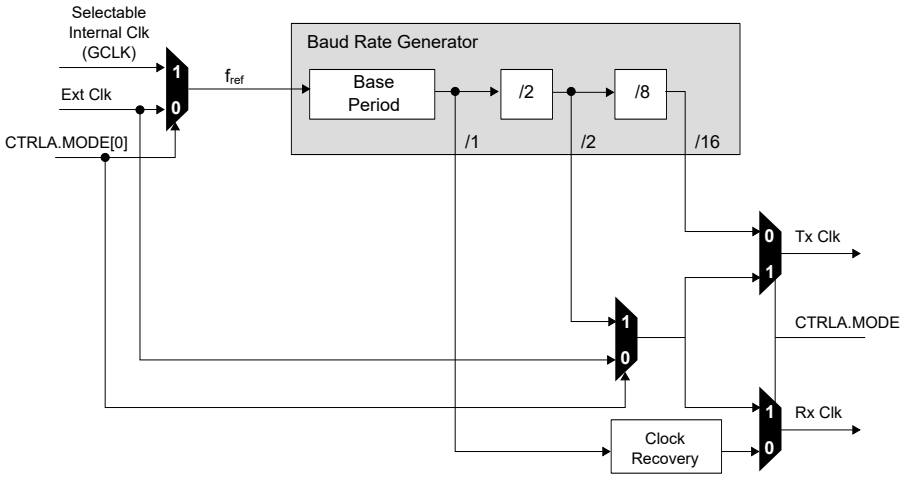


Table 34-2 contains equations for the baud rate (in bits per second) and the BAUD register value for each operating mode.

For asynchronous operation, there is one mode: *arithmetic mode*, the BAUD register value is 16 bits (0 to 65,535).

For synchronous operation, the BAUD register value is 8 bits (0 to 255).

Table 34-2. Baud Rate Equations

Operating Mode	Condition	Baud Rate (Bits Per Second)	BAUD Register Value Calculation
Asynchronous Arithmetic	$f_{BAUD} \leq \frac{f_{ref}}{16}$	$f_{BAUD} = \frac{f_{ref}}{16}\left(1 - \frac{BAUD}{65536}\right)$	$BAUD = 65536 \cdot \left(1 - 16 \cdot \frac{f_{BAUD}}{f_{ref}}\right)$
Asynchronous Fractional	$f_{BAUD} \leq \frac{f_{ref}}{S}$	$f_{BAUD} = \frac{f_{ref}}{S \cdot \left(BAUD + \frac{FP}{8}\right)}$	$BAUD = \frac{f_{ref}}{S \cdot f_{BAUD}} - \frac{FP}{8}$
Synchronous	$f_{BAUD} \leq \frac{f_{ref}}{2}$	$f_{BAUD} = \frac{f_{ref}}{2 \cdot (BAUD + 1)}$	$BAUD = \frac{f_{ref}}{2 \cdot f_{BAUD}} - 1$

S - Number of samples per bit, which can be 16, 8, or 3.

The Asynchronous Fractional option is used for auto-baud detection.

The baud rate error is represented by the following formula:

$$\text{Error} = 1 - \left(\frac{\text{ExpectedBaudRate}}{\text{ActualBaudRate}}\right)$$

34.6.2.3.1 Asynchronous Arithmetic Mode BAUD Value Selection

The formula given for f_{BAUD} calculates the average frequency over 65536 f_{ref} cycles. Although the BAUD register can be set to any value between 0 and 65536, the actual average frequency of f_{BAUD} over a single frame is more granular. The BAUD register values that will affect the average frequency over a single frame lead to an integer increase in the cycles per frame (CPF)

$$CPF = \frac{f_{ref}}{f_{BAUD}}(D + S)$$

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

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

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

On **SAM L11** devices, this peripheral has different access permissions depending on PAC Security Attribution (Secure or Non-Secure):

- If the peripheral is configured as Non-Secure in the PAC:
  - Secure access and Non-Secure access are granted
- If the peripheral is configured as Secure in the PAC:
  - Secure access is granted
  - Non-Secure access is discarded (Write is ignored, read 0x0) and a PAC error is triggered

Refer to *Peripherals Security Attribution* for more information.

If CMD 0x1 is issued, a repeated start will be issued followed by the transmission of the current address in ADDR.ADDR. If another address is desired, ADDR.ADDR must be written instead of the CMD bits. This will trigger a repeated start followed by transmission of the new address.

Issuing a command will set the System Operation bit in the Synchronization Busy register (SYNCBUSY.SYSOP).

Table 37-4. Command Description

CMD[1:0]	Direction	Action
0x0	X	(No action)
0x1	X	Execute acknowledge action succeeded by repeated Start
0x2	0 (Write)	No operation
	1 (Read)	Execute acknowledge action succeeded by a byte read operation
0x3	X	Execute acknowledge action succeeded by issuing a stop condition

These bits are not enable-protected.

Bit 9 – QCEN Quick Command Enable
This bit is not write-synchronized.

Value	Description
0	Quick Command is disabled.
1	Quick Command is enabled.

Bit 8 – SMEN Smart Mode Enable
When smart mode is enabled, acknowledge action is sent when DATA.DATA is read.

This bit is not write-synchronized.

Value	Description
0	Smart mode is disabled.
1	Smart mode is enabled.

37.10.11 Debug Control

Name: DBGCTRL
Offset: 0x30
Reset: 0x00
Property: PAC Write-Protection

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

Bit 0 – DBGSTOP Debug Stop Mode
This bit controls functionality when the CPU is halted by an external debugger.

Value	Description
0	The baud-rate generator continues normal operation when the CPU is halted by an external debugger.
1	The baud-rate generator is halted when the CPU is halted by an external debugger.

38.7.2.8 Status

Name: STATUS
Offset: 0x0B
Reset: 0x01
Property: Read-Synchronized

Bit	7	6	5	4	3	2	1	0
				CCBUFVx	PERBUFV		SLAVE	STOP
Access				R/W	R/W		R	R
Reset				0	0		0	1

Bit 4 – CCBUFVx Channel x Compare or Capture Buffer Valid
For a compare channel x, the bit x is set when a new value is written to the corresponding CCBUFx register.

The bit x is cleared by writing a '1' to it when CTRLB.LUPD is set, or it is cleared automatically by hardware on UPDATE condition.

For a capture channel x, the bit x is set when a valid capture value is stored in the CCBUFx register. The bit x is cleared automatically when the CCx register is read.

Bit 3 – PERBUFV Period Buffer Valid
This bit is set when a new value is written to the PERBUF register. The bit is cleared by writing '1' to the corresponding location when CTRLB.LUPD is set, or automatically cleared by hardware on UPDATE condition. This bit is available only in 8-bit mode and will always read zero in 16- and 32-bit modes.

Bit 1 – SLAVE Slave Status Flag
This bit is only available in 32-bit mode on the slave TC (i.e., TC1 and/or TC3). The bit is set when the associated master TC (TC0 and TC2, respectively) is set to run in 32-bit mode.

Bit 0 – STOP Stop Status Flag
This bit is set when the TC is disabled, on a Stop command, or on an overflow/underflow condition when the One-Shot bit in the Control B Set register (CTRLBSET.ONESHOT) is '1'.

Value	Description
0	Counter is running.
1	Counter is stopped.

Symbol	Parameters	Conditions		Measurements			Unit
				Min	Typ	Max	
			Vref=3V Vddana=1.6V to 3.6V	-	+/-1.82	+/-14.9	
			Bandgap Reference	-	+/-2.07	+/-15.8	
			Vref=Vddana=1.6V to 3.6V	-	+/-1.82	+/-15.3	
SFDR	Spurious Free Dynamic Range	Fs = 1MHz / Fin = 13 kHz / Full range Input signal	Vref=2.0V Vddana=3.0V	58.1	70.5	77.5	dB
SINAD	Signal to Noise and Distortion ratio			56.7	63.4	66.5	
SNR	Signal to Noise ratio			56.5	64.4	67.1	
THD	Total Harmonic Distortion			-74.7	-68.7	-57.7	
-	Noise RMS	External Reference voltage		-	0.42	-	mV

Note:

1. These are given without any ADC oversampling and decimation features enabled.

Table 46-25. Single-Ended Mode (1)

Symbol	Parameters	Conditions		Measurements			Unit
				Min	Typ	Max	
ENOB	Effective Number of bits	Fadc = 1Msps	Vref=2.0V Vddana=3.0V	8.0	9.3	9.7	bits
			Vref=1.0V Vddana=1.6V to 3.6V	7.9	8.2	9.4	
			Vref=Vddana=1.6V to 3.6V	8.6	9.2	9.9	
			Bandgap Reference, Vddana=1.6V to 3.6V	7.8	8.4	8.9	
TUE	Total Unadjusted Error	without offset and gain compensation	Vref=2.0V Vddana=3.0V	-	12	63	LSB
INL	Integral Non Linearity	without offset and gain	Vref=2.0V Vddana=3.0V	-	+/-3.4	+/-8.9	

Table 46-37. Analog Gain Settings (1)

Symbol	Setting	Average
Gain	GAIN_1	1.0
	GAIN_2	2.0
	GAIN_4	3.9
	GAIN_8	8.1

Note:

1. Analog Gain is a parameter of the QTouch Library. Refer to the QTouch Library Peripheral Touch Controller User Guide.

Power Consumption

The values in the Power Consumption table below are measured values of power consumption under the following conditions:

Operating Conditions

- VDD = 3.3V

Clocks

- OSC16M divided to 4MHz used as main clock source
- CPU is running on flash with 0 wait states, at 4MHz
- PTC running at 4MHz
- Voltage Regulator mode: LPEFF enabled

PTC Configuration

- Mutual-capacitance mode
- One touch channel

System Configuration

- Standby sleep mode enabled
- RTC running on OSCULP32K: used to define the PTC scan rate, through the event system
- Drift Calibration disabled: no interrupts, PTC scans are performed in standby mode
- Drift Calibration enabled: RTC interrupts (wakeup) the CPU to perform PTC scans. PTC drift calibration is performed every 1.5 sec.

Table 46-38. Power Consumption (1)

Symbol	Parameters	Drift Calibration	PTC scan rate (msec)	Oversamples	Ta	Typ.	Max.	Units
IDD	Current Consumption	Disabled	10	4	Max. 85°C Typ. 25°C	6.2	49.2	µA
				16		12.7	58.1	
			50	4		2.3	43.7	
				16		3.7	45.5	