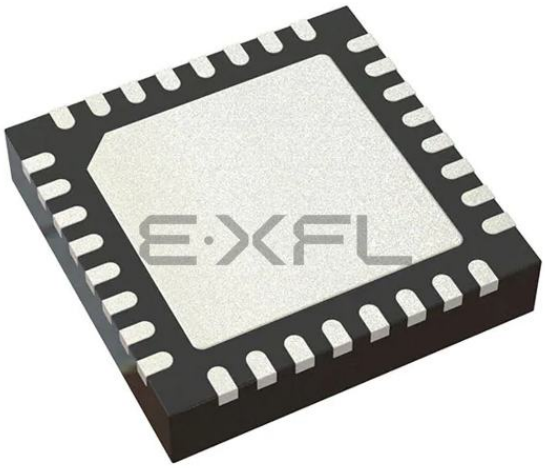


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

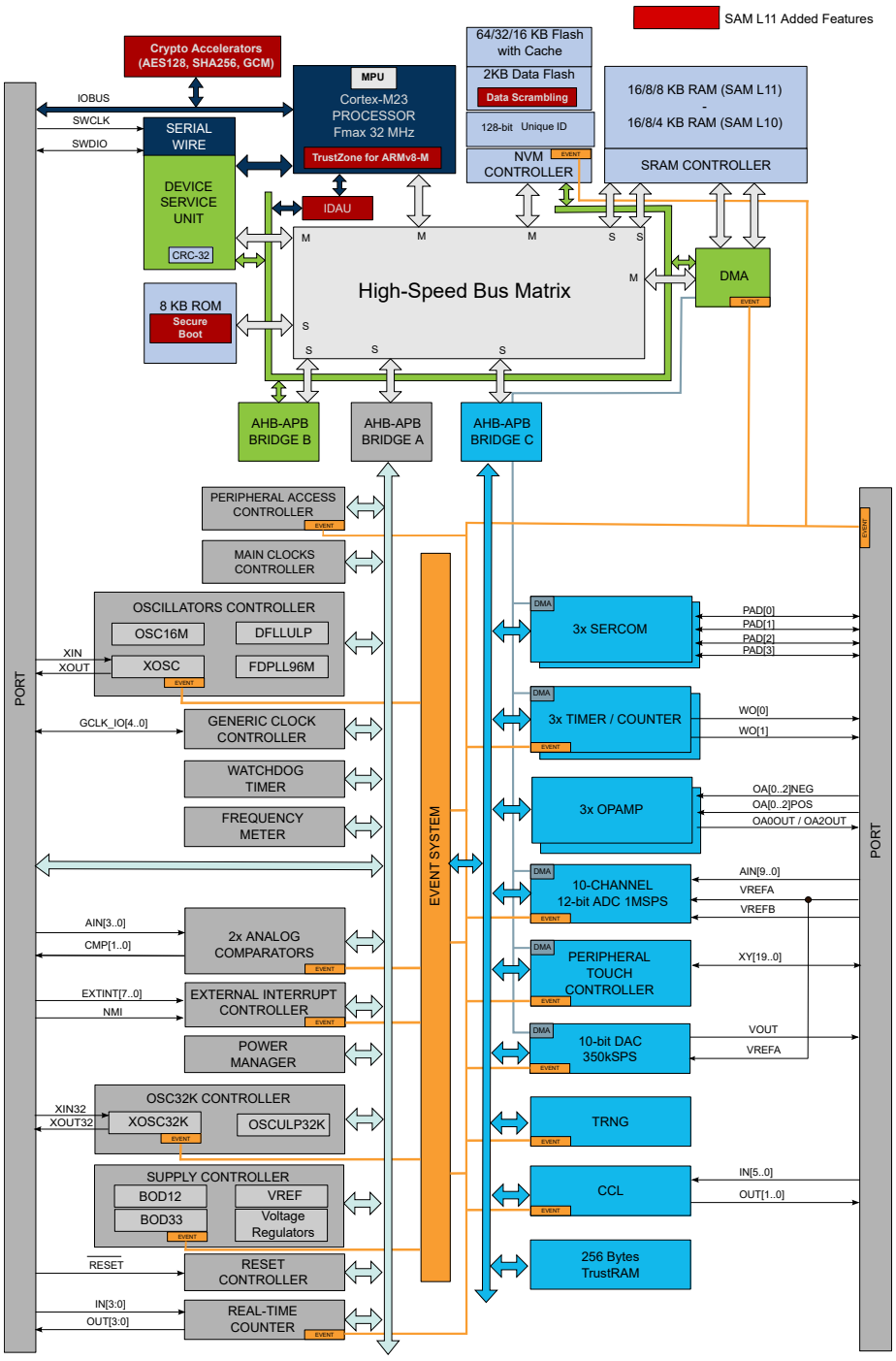
"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" 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	8K 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/atsaml11e14a-mf

3. Block Diagram

Figure 3-1. SAM L10/L11 Block Diagram



Note: Number of SERCOM instances, PTC/ADC channels, Tamper input pins, and Analog Compare inputs differ on the packages pinout.

The priority order for concurrent accesses are decided by two factors:

- As first priority, the QoS level for the master.
- As a second priority, a static priority given by the port ID. The lowest port ID has the highest static priority.

See the tables below for more details.

Table 11-8. HS SRAM Port Connections QoS

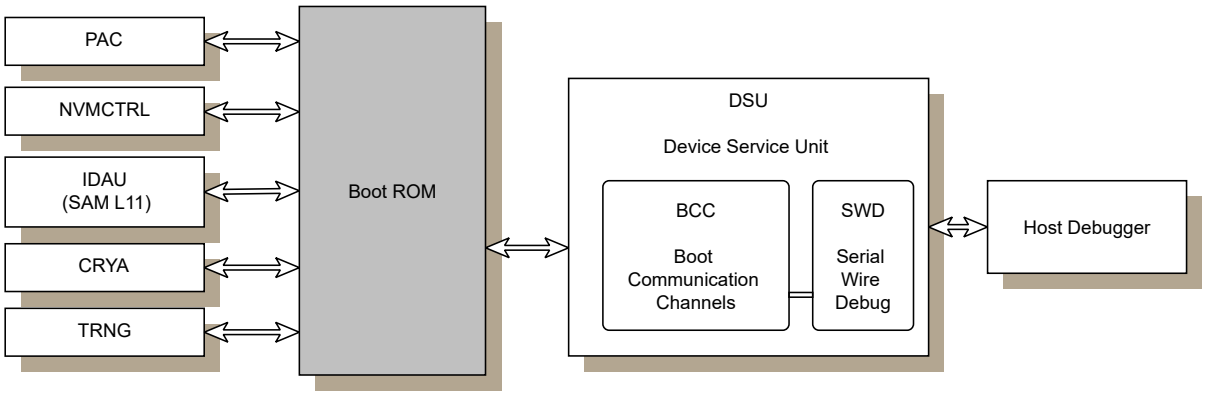
HS SRAM Port Connection	Port ID	Connection Type	QoS	default QoS
DMAC - Direct Memory Access Controller - Write-Back 1 Access	6	Direct	DMAC QOSCTRL.WRBQOS	0x2
DMAC - Direct Memory Access Controller - Write-Back 0 Access	5	Direct	DMAC QOSCTRL.WRBQOS	0x2
DMAC - Direct Memory Access Controller - Fetch 1 Access	4	Direct	DMAC QOSCTRL.FQOS	0x2
DMAC - Direct Memory Access Controller - Fetch 0 Access	3	Direct	DMAC QOSCTRL.FQOS	0x2
DMAC - Direct Memory Access Controller - Data Access	2	Bus Matrix	DMAC QOSCTRL.DQOS	0x2
DSU - Device Service Unit	1	Bus Matrix	DSU CFG.LQOS	0x2
CM23 - Cortex M23 Processor	0	Bus Matrix	0x41008114, bits[1:0](1)	0x3

Note:

1. The CPU QoS level can be written/read, using 32-bit access only.

14.2 Block Diagram

Figure 14-1. Boot ROM Block Diagram



Related Links

13.1 Features

14.3 Product Dependencies

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

Related Links

13.1 Features

14.3.1 Clocks

The device selects the OSC16M oscillator which is enabled by default after reset and configured at 4 MHz.

14.3.2 NVM User (UROW) and Boot Configuration (BOCOR) rows

The Boot ROM reads the different NVM rows during its execution.

The relevant fuses must be set appropriately by any configuration tools supported the device in order to operate correctly.

Refer to the 10.2 NVM Rows section for additional information.

14.3.3 Debug Operations

For security reasons, no debug is possible during the Boot ROM execution except when entering the Boot ROM CPU Park mode.

14.4 Functional Description

Related Links

13.1 Features

14.4.1 SAM L10 Boot ROM Flow

The SAM L10 Boot ROM checks firstly if a debugger is present to enter the Boot Interactive mode which allows the user to perform specific tasks via a debugger connection.

22. PM – Power Manager

22.1 Overview

The Power Manager (PM) controls the sleep modes and the power domain gating 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.

Performance level technique consists of adjusting the regulator output voltage to reduce power consumption. The user can select on the fly the performance level configuration which best suits the application.

The power domain gating technique enables the PM to turn off unused power domain supplies individually, while keeping others powered up. Based on activity monitoring, power domain gating is managed automatically by hardware without software intervention. This technique is transparent for the application while minimizing the static consumption. The user can also manually control which power domains will be turned on and off in standby sleep mode.

The internal state of the logic is retained (retention state) allowing the application context to be kept in non-active states.

22.2 Features

- Power management control
  - Sleep modes: Idle, Standby, and Off
  - Performance levels: PL0 and PL2
  - SleepWalking available in Standby mode.
  - Full retention state in Standby mode
- Power Domain Control
  - Standby Sleep Mode with static power gating
  - SleepWalking extension to power gating
  - SRAM sub-blocks retention in Standby mode

voltage regulator output supply level is automatically defined by the performance level or the sleep mode selected in the Power Manager module.

Related Links

22. PM – Power Manager

25.6.1.2 Initialization

After a Reset, the LDO voltage regulator supplying VDDCORE is enabled.

25.6.1.3 Selecting a Voltage Regulator

In Active mode, the type of the main voltage regulator supplying VDDCORE can be switched on the fly. The two alternatives are a LDO regulator and a Buck converter.

The main voltage regulator switching sequences are as follows:

- The user changes the value of the Voltage Regulator Selection bit in the Voltage Regulator System Control register (VREG.SEL)
- The start of the switching sequence is indicated by clearing the Voltage Regulator Ready bit in the STATUS register (STATUS.VREGRDY=0)
- Once the switching sequence is completed, STATUS.VREGRDY will read '1'

The Voltage Regulator Ready (VREGRDY) interrupt can also be used to detect a zero-to-one transition of the STATUS.VREGRDY bit.

25.6.1.4 Voltage Scaling Control

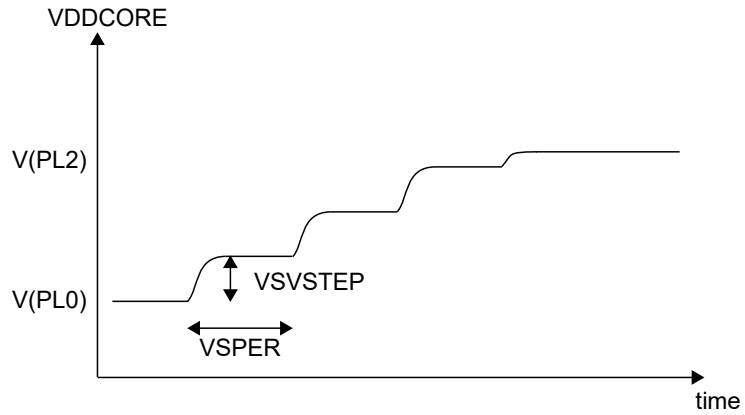
The VDDCORE supply will change under certain circumstances:

- When a new performance level (PL) is set
- When the Standby Sleep mode is entered or left
- When a sleepwalking task is requested in Standby Sleep mode

To prevent high peak current on the main power supply and to have a smooth transition of VDDCORE, both the voltage scaling step size and the voltage scaling frequency can be controlled: VDDCORE is changed by the selected step size of the selected period until the target voltage is reached.

The Voltage Scaling Voltage Step field is in the VREG register, VREG.VSVSTEP. The Voltage Scaling Period field is VREG.VSPER.

The following waveform shows an example of changing performance level from PL0 to PL2.



Setting VREG.VSVSTEP to the maximum value allows to transition in one voltage step.

Value	Name	Description
0x2	INT	The BOD33 generates an interrupt
0x3	-	Reserved

Bit 2 – HYST Hysteresis
This bit indicates whether hysteresis is enabled for the BOD33 threshold voltage.

This bit is loaded from NVM User Row at start-up.

This bit is not synchronized.

Value	Description
0	No hysteresis.
1	Hysteresis enabled.

Bit 1 – ENABLE Enable
This bit is loaded from NVM User Row at start-up.

This bit is not enable-protected.

Value	Description
0	BOD33 is disabled.
1	BOD33 is enabled.

- The transfer descriptor must be made valid by writing a one to the Valid bit in the Block Transfer Control register (BTCTRL.VALID)
- Number of beats in the block transfer must be selected by writing the Block Transfer Count (BTCNT) register
- Source address for the block transfer must be selected by writing the Block Transfer Source Address (SRCADDR) register
- Destination address for the block transfer must be selected by writing the Block Transfer Destination Address (DSTADDR) register

If CRC calculation is needed, the CRC engine must be configured before it is enabled, as outlined by the following steps:

- The CRC input source must selected by writing the CRC Input Source bit group in the CRC Control register (CRCCTRL.CRCSRC)
- The type of CRC calculation must be selected by writing the CRC Polynomial Type bit group in the CRC Control register (CRCCTRL.CRCPOLY)
- If I/O is selected as input source, the beat size must be selected by writing the CRC Beat Size bit group in the CRC Control register (CRCCTRL.CRCBEATSIZE)

28.6.2.2 Enabling, Disabling, and Resetting

The DMAC is enabled by writing the DMA Enable bit in the Control register (CTRL.DMAENABLE) to '1'. The DMAC is disabled by writing a '0' to CTRL.DMAENABLE.

A DMA channel is enabled by writing the Enable bit in the Channel Control A register (CHCTRLA.ENABLE) to '1', after writing the corresponding channel id to the Channel ID bit group in the Channel ID register (CHID.ID). A DMA channel is disabled by writing a '0' to CHCTRLA.ENABLE.

The CRC is enabled by writing a '1' to the CRC Enable bit in the Control register (CTRL.CRCENABLE). The CRC is disabled by writing a '0' to CTRL.CRCENABLE.

The DMAC is reset by writing a '1' to the Software Reset bit in the Control register (CTRL.SWRST) while the DMAC and CRC are disabled. All registers in the DMAC except DBGCTRL will be reset to their initial state.

A DMA channel is reset by writing a '1' to the Software Reset bit in the Channel Control A register (CHCTRLA.SWRST), after writing the corresponding channel id to the Channel ID bit group in the Channel ID register (CHID.ID). The channel registers will be reset to their initial state. The corresponding DMA channel must be disabled in order for the reset to take effect.

28.6.2.3 Transfer Descriptors

Together with the channel configurations the transfer descriptors decides how a block transfer should be executed. Before a DMA channel is enabled (CHCTRLA.ENABLE is written to one), and receives a transfer trigger, its first transfer descriptor has to be initialized and valid (BTCTRL.VALID). The first transfer descriptor describes the first block transfer of a transaction.

All transfer descriptors must reside in SRAM. The addresses stored in the Descriptor Memory Section Base Address (BASEADDR) and Write-Back Memory Section Base Address (WRBADDR) registers tell the DMAC where to find the descriptor memory section and the write-back memory section.

The descriptor memory section is where the DMAC expects to find the first transfer descriptors for all DMA channels. As BASEADDR points only to the first transfer descriptor of channel 0 (see figure below), all first transfer descriptors must be stored in a contiguous memory section, where the transfer descriptors must be ordered according to their channel number. For further details on linked descriptors, refer to 28.6.3.1 Linked Descriptors.

CRC on DMA data CRC-16 or CRC-32 calculations can be performed on data passing through any DMA channel. Once a DMA channel is selected as the source, the CRC engine will continuously generate the CRC on the data passing through the DMA channel. The checksum is available for readout once the DMA transaction is completed or aborted. A CRC can also be generated on SRAM, Flash, or I/O memory by passing these data through a DMA channel. If the latter is done, the destination register for the DMA data can be the data input (CRCDATAIN) register in the CRC engine.

CRC using the I/O interface Before using the CRC engine with the I/O interface, the application must set the CRC Beat Size bits in the CRC Control register (CRCCTRL.CRCBEATSIZE). 8/16/32-bit bus transfer type can be selected.

CRC can be performed on any data by loading them into the CRC engine using the CPU and writing the data to the CRCDATAIN register. Using this method, an arbitrary number of bytes can be written to the register by the CPU, and CRC is done continuously for each byte. This means if a 32-bit data is written to the CRCDATAIN register the CRC engine takes four cycles to calculate the CRC. The CRC complete is signaled by a set CRCBUSY bit in the CRCSTATUS register. New data can be written only when CRCBUSY flag is not set.

28.6.4 DMA Operation

Not applicable.

28.6.5 Interrupts

The DMAC channels have the following interrupt sources:

- Transfer Complete (TCMPL): Indicates that a block transfer is completed on the corresponding channel. Refer to 28.6.2.5 Data Transmission for details.
- Transfer Error (TERR): Indicates that a bus error has occurred during a burst transfer, or that an invalid descriptor has been fetched. Refer to 28.6.2.8 Error Handling for details.
- Channel Suspend (SUSP): Indicates that the corresponding channel has been suspended. Refer to 28.6.3.2 Channel Suspend and 28.6.2.5 Data Transmission for details.

Each interrupt source has an interrupt flag associated with it. The interrupt flag in the Channel Interrupt Flag Status and Clear (CHINTFLAG) register is set when the interrupt condition occurs. Each interrupt can be individually enabled by setting the corresponding bit in the Channel Interrupt Enable Set register (CHINTENSET=1), and disabled by setting the corresponding bit in the Channel Interrupt Enable Clear register (CHINTENCLR=1).

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, the DMAC is reset or the corresponding DMA channel is reset. See CHINTFLAG for details on how to clear interrupt flags. All interrupt requests are ORed together on system level to generate one combined interrupt request to the NVIC.

The user must read the Channel Interrupt Status (INTSTATUS) register to identify the channels with pending interrupts and must read the Channel Interrupt Flag Status and Clear (CHINTFLAG) register to determine which interrupt condition is present for the corresponding channel. It is also possible to read the Interrupt Pending register (INTPEND), which provides the lowest channel number with pending interrupt and the respective interrupt flags.

Note: Interrupts must be globally enabled for interrupt requests to be generated.

Offset	Name	Bit Pos.								
		31:24								
0x30	ACTIVE	7:0					LVLEXx	LVLEXx	LVLEXx	LVLEXx
		15:8	ABUSY			ID[4:0]				
		23:16	BTCNT[7:0]							
		31:24	BTCNT[15:8]							
0x34	BASEADDR	7:0	BASEADDR[7:0]							
		15:8			BASEADDR[13:8]					
		23:16								
		31:24								
0x38	WRBADDR	7:0	WRBADDR[7:0]							
		15:8			WRBADDR[13:8]					
		23:16								
		31:24								
0x3C ... 0x3E	Reserved									
0x3F	CHID	7:0					ID[3:0]			
0x40	CHCTRLA	7:0		RUNSTDBY					ENABLE	SWRST
0x41 ... 0x43	Reserved									
0x44	CHCTRLB	7:0		LVL[1:0]		EVOE	EVIE	EVACT[2:0]		
		15:8				TRIGSRC[4:0]				
		23:16	TRIGACT[1:0]							
		31:24							CMD[1:0]	
0x48 ... 0x4B	Reserved									
0x4C	CHINTENCLR	7:0						SUSP	TCMPL	TERR
0x4D	CHINTENSET	7:0						SUSP	TCMPL	TERR
0x4E	CHINTFLAG	7:0						SUSP	TCMPL	TERR
0x4F	CHSTATUS	7:0						FERR	BUSY	PEND

28.8 Register Description

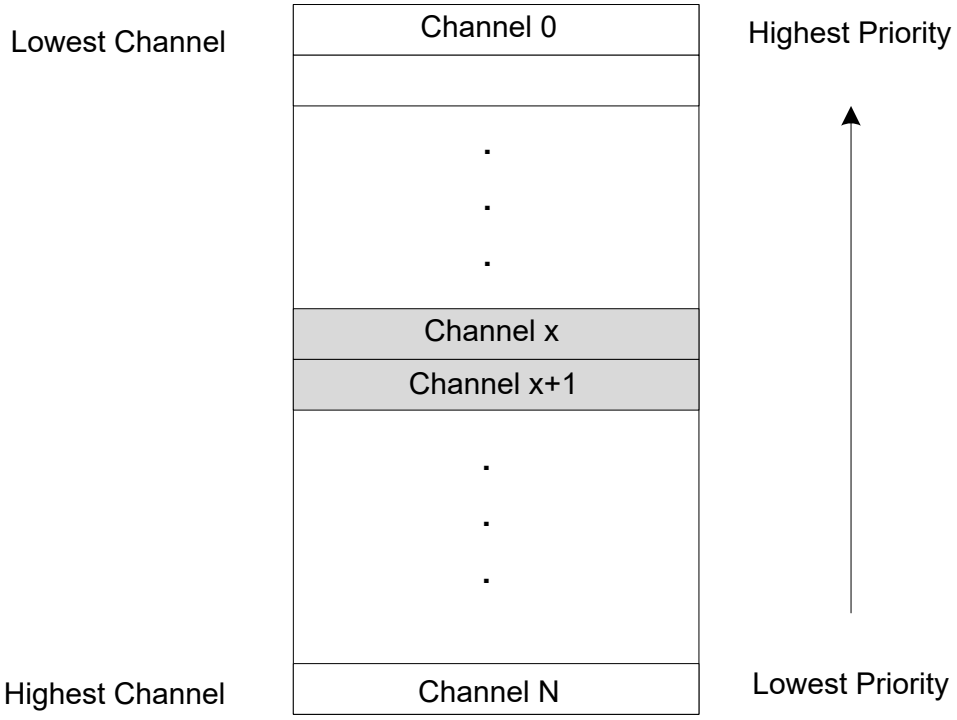
Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16- and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). Optional PAC write-protection is denoted by the "PAC Write-Protection" property in each individual register description. For details, refer to 28.5.8 Register Access Protection.

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

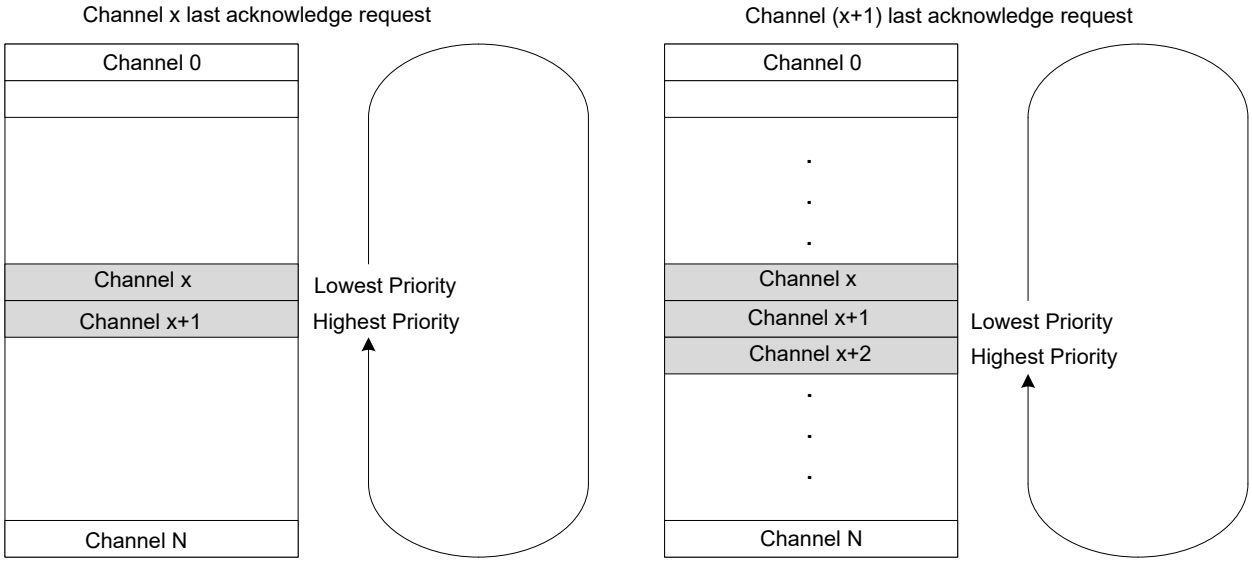
On **SAM L11** devices, this peripheral has different access permissions depending on PAC Security Attribution (Secure or Non-Secure):

Figure 33-3. Static Priority



The dynamic arbitration scheme available in the Event System is round-robin. Round-robin arbitration is enabled by writing PRICTRL.RREN to one. With the round-robin scheme, the channel number of the last channel being granted access will have the lowest priority the next time the arbiter has to grant access to a channel, as shown below. The channel number of the last channel being granted access, will be stored in the Channel Priority Number bit group in the Priority Control register (PRICTRL.PRI).

Figure 33-4. Round-Robin Scheduling



The Channel Pending Interrupt register (INTPEND) also offers the possibility to indirectly clear the interrupt flags of a specific channel. Writing a flag to one in this register, will clear the corresponding interrupt flag of the channel specified by the INTPEND.ID bits.

Case 1: Address packet accepted – Read flag set

The STATUS.DIR bit is ‘1’, indicating an I2C master read operation. The SCL line is forced low, stretching the bus clock. If an ACK is sent, I2C slave hardware will set the Data Ready bit in the Interrupt Flag register (INTFLAG.DRDY), indicating data are needed for transmit. If a NACK is sent, the I2C slave will wait for a new start condition and address match.

Typically, software will immediately acknowledge the address packet by sending an ACK/NACK bit. The I2C slave Command bit field in the Control B register (CTRLB.CMD) can be written to '0x3' for both read and write operations as the command execution is dependent on the STATUS.DIR bit. Writing ‘1’ to INTFLAG.AMATCH will also cause an ACK/NACK to be sent corresponding to the CTRLB.ACKACT bit.

Case 2: Address packet accepted – Write flag set

The STATUS.DIR bit is cleared, indicating an I2C master write operation. The SCL line is forced low, stretching the bus clock. If an ACK is sent, the I2C slave will wait for data to be received. Data, repeated start or stop can be received.

If a NACK is sent, the I2C slave will wait for a new start condition and address match. Typically, software will immediately acknowledge the address packet by sending an ACK/NACK. The I2C slave command CTRLB.CMD = 3 can be used for both read and write operation as the command execution is dependent on STATUS.DIR.

Writing ‘1’ to INTFLAG.AMATCH will also cause an ACK/NACK to be sent corresponding to the CTRLB.ACKACT bit.

37.6.2.5.2 Receiving Address Packets (SCLSM=1)

When SCLSM=1, the I2C slave will stretch the SCL line only after an ACK, see Slave Behavioral Diagram (SCLSM=1). When the I2C slave is properly configured, it will wait for a start condition to be detected.

When a start condition is detected, the successive address packet will be received and checked by the address match logic.

If the received address is not a match, the packet will be rejected and the I2C slave will wait for a new start condition.

If the address matches, the acknowledge action as configured by the Acknowledge Action bit Control B register (CTRLB.ACKACT) will be sent and the Address Match bit in the Interrupt Flag register (INTFLAG.AMATCH) is set. SCL will be stretched until the I2C slave clears INTFLAG.AMATCH. As the I2C slave holds the clock by forcing SCL low, the software is given unlimited time to respond to the address.

The direction of a transaction is determined by reading the Read/Write Direction bit in the Status register (STATUS.DIR). This bit will be updated only when a valid address packet is received.

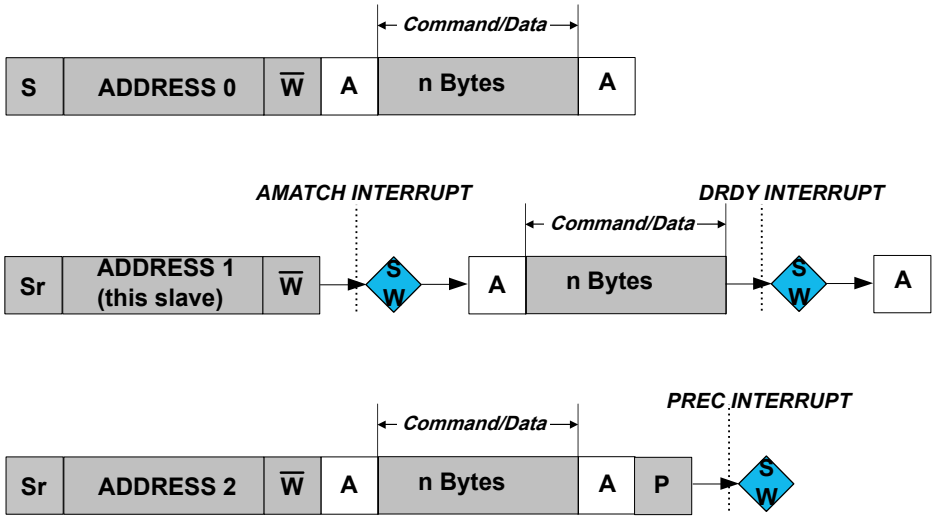
If the Transmit Collision bit in the Status register (STATUS.COLL) is set, the last packet addressed to the I2C slave had a packet collision. A collision causes the SDA and SCL lines to be released without any notification to software. The next AMATCH interrupt is, therefore, the first indication of the previous packet’s collision. Collisions are intended to follow the SMBus Address Resolution Protocol (*ARP*).

After the address packet has been received from the I2C master, INTFLAG.AMATCH be set to ‘1’ to clear it.

37.6.2.5.3 Receiving and Transmitting Data Packets

After the I2C slave has received an address packet, it will respond according to the direction either by waiting for the data packet to be received or by starting to send a data packet by writing to DATA.DATA.

Figure 37-13. PMBus Group Command Example



37.6.3 Additional Features

37.6.3.1 SMBus

The I^2C includes three hardware SCL low time-outs which allow a time-out to occur for SMBus SCL low time-out, master extend time-out, and slave extend time-out. This allows for SMBus functionality These time-outs are driven by the GCLK_SERCOM_SLOW clock. The GCLK_SERCOM_SLOW clock is used to accurately time the time-out and must be configured to use a 32KHz oscillator. The I^2C interface also allows for a SMBus compatible SDA hold time.

- $T_{TIMEOUT}$: SCL low time of 25..35ms – Measured for a single SCL low period. It is enabled by CTRLA.LOWTOUTEN.
- $T_{LOW:SEXT}$: Cumulative clock low extend time of 25 ms – Measured as the cumulative SCL low extend time by a slave device in a single message from the initial START to the STOP. It is enabled by CTRLA.SEXTTOEN.
- $T_{LOW:MEXT}$: Cumulative clock low extend time of 10 ms – Measured as the cumulative SCL low extend time by the master device within a single byte from START-to-ACK, ACK-to-ACK, or ACK-to-STOP. It is enabled by CTRLA.MEXTTOEN.

37.6.3.2 Smart Mode

The I^2C interface has a smart mode that simplifies application code and minimizes the user interaction needed to adhere to the I^2C protocol. The smart mode accomplishes this by automatically issuing an ACK or NACK (based on the content of CTRLB.ACKACT) as soon as DATA.DATA is read.

37.6.3.3 4-Wire Mode

Writing a '1' to the Pin Usage bit in the Control A register (CTRLA.PINOUT) will enable 4-wire mode operation. In this mode, the internal I^2C tri-state drivers are bypassed, and an external I^2C compliant tri-state driver is needed when connecting to an I^2C bus.

37.10.8 Synchronization Busy

Name: SYNCBUSY
Offset: 0x1C
Reset: 0x00000000

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
						SYSOP	ENABLE	SWRST
Access						R	R	R
Reset						0	0	0

Bit 2 – SYSOP System Operation Synchronization Busy
Writing CTRLB.CMD, STATUS.BUSSTATE, ADDR, or DATA when the SERCOM is enabled requires synchronization. When written, the SYNCBUSY.SYSOP bit will be set until synchronization is complete.

Value	Description
0	System operation synchronization is not busy.
1	System operation synchronization is busy.

Bit 1 – ENABLE SERCOM Enable Synchronization Busy
Enabling and disabling the SERCOM (CTRLA.ENABLE) requires synchronization. When written, the SYNCBUSY.ENABLE bit will be set until synchronization is complete.

Value	Description
0	Enable synchronization is not busy.
1	Enable synchronization is busy.

Bit 0 – SWRST Software Reset Synchronization Busy
Resetting the SERCOM (CTRLA.SWRST) requires synchronization. When written, the SYNCBUSY.SWRST bit will be set until synchronization is complete.

Value	Description
0	SWRST synchronization is not busy.
1	SWRST synchronization is busy.

37.10.9 Address

Name: ADDR
Offset: 0x24
Reset: 0x0000
Property: Write-Synchronized

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	LEN[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	TENBITEN	HS	LENEN			ADDR[10:8]		
Access	R/W	R/W	R/W			R/W	R/W	R/W
Reset	0	0	0			0	0	0

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

Bits 23:16 – LEN[7:0] Transaction Length
These bits define the transaction length of a DMA transaction from 0 to 255 bytes. The Transfer Length Enable (LENEN) bit must be written to '1' in order to use DMA.

Bit 15 – TENBITEN Ten Bit Addressing Enable
This bit enables 10-bit addressing. This bit can be written simultaneously with ADDR to indicate a 10-bit or 7-bit address transmission.

Value	Description
0	10-bit addressing disabled.
1	10-bit addressing enabled.

Bit 14 – HS High Speed
This bit enables High-speed mode for the current transfer from repeated START to STOP. This bit can be written simultaneously with ADDR for a high speed transfer.

Value	Description
0	High-speed transfer disabled.
1	High-speed transfer enabled.

Bit 13 – LENEN Transfer Length Enable

38.7.1.9 Waveform Generation Control

Name: WAVE
Offset: 0x0C
Reset: 0x00
Property: PAC Write-Protection, Enable-Protected

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

Bits 1:0 – WAVEGEN[1:0] Waveform Generation Mode
These bits select the waveform generation operation. They affect the top value, as shown in 38.6.2.6.1 Waveform Output Operations. They also control whether frequency or PWM waveform generation should be used. The waveform generation operations are explained in 38.6.2.6.1 Waveform Output Operations.

These bits are not synchronized.

Value	Name	Operation	Top Value	Output Waveform on Match	Output Waveform on Wraparound
0x0	NFRQ	Normal frequency	PER[1] / Max	Toggle	No action
0x1	MFRQ	Match frequency	CC0	Toggle	No action
0x2	NPWM	Normal PWM	PER[1] / Max	Set	Clear
0x3	MPWM	Match PWM	CC0	Set	Clear

1) This depends on the TC mode: In 8-bit mode, the top value is the Period Value register (PER). In 16- and 32-bit mode it is the respective MAX value.

38.7.3.8 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.

This interrupt is a synchronous wake-up source. See *Sleep Mode Controller* for details.

The interrupt source has an interrupt flag associated with it. The interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.DATARDY) is set to ‘1’ when the interrupt condition occurs. The interrupt can be enabled by writing a '1' to the corresponding bit in the Interrupt Enable Set register (INTENSET.DATARDY), 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, or the interrupt is disabled. See 39.8.5 INTFLAG for details on how to clear interrupt flags.

Note that interrupts must be globally enabled for interrupt requests to be generated.

Related Links

22.6.3.3 Sleep Mode Controller

39.6.4 Events

The TRNG can generate the following output event:

- Data Ready (DATARDY): Generated when a new random number is available in the DATA register.

Writing '1' to the Data Ready Event Output bit in the Event Control Register (EVCTRL.DATARDYEO) enables the DTARDY event. Writing a '0' to this bit disables the corresponding output event. Refer to *EVSYS – Event System* for details on configuring the Event System.

Related Links

33. EVSYS – Event System

39.6.5 Sleep Mode Operation

The Run in Standby bit in Control A register (CTRLA.RUNSTDBY) controls the behavior of the TRNG during standby sleep mode:

When this bit is '0', the TRNG is disabled during sleep, but maintains its current configuration.

When this bit is '1', the TRNG continues to operate during sleep and any enabled TRNG interrupt source can wake up the CPU.

39.6.6 Synchronization

Not applicable.

Signal	Description	Type
OA1NEG	OPAMP1 negative input	Analog input
OA2POS	OPAMP2 positive input	Analog input
OA2NEG	OPAMP2 negative input	Analog input
OA0OUT	OPAMP0 output	Analog output
OA1OUT	OPAMP1 output	Analog output
OA2OUT	OPAMP2 output	Analog output

One signal can be mapped on several pins.

Important:
When an analog peripheral is enabled, the analog output of the peripheral will interfere with the alternative functions of the output pads. This is also true even when the peripheral is used for internal purposes.

Analog inputs do not interfere with alternative pad functions.

44.5 Product Dependencies

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

44.5.1 I/O Lines

Using the OPAMP I/O lines requires the I/O pins to be configured. Refer to the *PORT - I/O Pin Controller* chapter for details.

44.5.2 Power Management

The OPAMP can operate in idle and standby sleep mode, according to the settings of the Run in Standby and On Demand bits in the OPAMP Control x registers (OPAMPCTRLx.RUNSTDBY and OPAMPCTRLx.ONDEMAND), as well as the Enable bit in the Control A register (CTRLA.ENABLE). Refer to *PM – Power Manager* for details on the different sleep modes.

Related Links

22. PM – Power Manager

44.5.3 Clocks

The OPAMP bus clock (CLK_OPAMP_APB) can be enabled and disabled in the Power Manager, and the default state of CLK_OPAMP_APB can be found in the *Peripheral Clock Masking*.

A clock (CLK_ULP32K) is required by the voltage doubler for low voltage operation (VCC < 2.5V). The CLK_ULP32K is a 32KHz clock which is provided by the OSCULP32K oscillator in the OSC32KCTRL module.

Related Links

19.6.2.6 Peripheral Clock Masking

44.5.4 DMA

Not applicable.

Mode	Conditions	Regulator Mode	Vcc	Ta	Typ.	Max.	Units
				125°C	15.5	33.2	
	4kB RAM retained,PDSW domain in retention	LPEFF Disable	1,8V	25°C	0.5	0.9	
				125°C	22.0	58.9	
		LPEFF Enable	3,3V	25°C	0.5	0.9	
				125°C	18.7	41.5	
		BUCK in standby PL0 (VREG.RUNSTDBY=1 and VREG.STDBYPL0=1)	1,8V	25°C	0.7	1.0	
				125°C	18.4	42.7	µA
			3,3V	25°C	0.8	1.5	
				125°C	14.6	31.0	
	4kB RAM retained,PDSW domain in retention and RTC running on XOSC32k	LPEFF Disable	1,8V	25°C	0.9	1.3	
				125°C	22.6	59.8	
		LPEFF Enable	3,3V	25°C	0.8	1.2	
				125°C	19.3	42.1	
		BUCK in standby PL0 (VREG.RUNSTDBY=1 and VREG.STDBYPL0=1)	1,8V	25°C	1.0	1.3	
				125°C	19.0	43.3	
			3,3V	25°C	1.1	1.7	
				125°C	15.2	31.6	
OFF			1,8V	25°C	34.6	54.4	nA
				125°C	4385.0	8291.5	
			3,3V	25°C	61.2	89.1	
				125°C	5489.5	10564.7	

47.4 Analog Characteristics

47.4.1 Brown-Out Detectors (BOD) Characteristics

Table 47-4. BOD33 Characteristics with BOD33.VREFSEL = 0

Symbol	Parameters	Conditions	Min.	Typ.	Max.	Unit
VBOD+ (2)	BOD33 high threshold level	BOD33.LEVEL = 6	1.66	1.68	1.70	V
		BOD33.LEVEL = 7	1.70	1.72	1.74	
		BOD33.LEVEL = 39	2.79	2.84	2.89	

Symbol	Parameter	Conditions		Min.	Typ.	Max.	Units
		Slave, VDD>1,62V		13.4	-	-	
tSOSS	MISO setup after SS low	Slave, VDD>2,70V			-	1* tSCK	
		Slave, VDD>1,62V			-	1* tSCK	
tSOSH	MISO hold after SS high	Slave, VDD>2,70V		8.7	-	-	
		Slave, VDD>1,62V		8.7	-	-	

Note:

1. These values are based on simulation. These values are not covered by test limits in production.
2. See I/O Pin Characteristics.
3. Where tSLAVE_OUT is the slave external device output response time, generally tEXT_SOV +tLINE_DELAY (See Note 7).
4. Where tSLAVE_IN is the slave external device input constraint, generally tEXT_SIS+tLINE_DELAY (See Note 7).
5. Where tMASTER_OUT is the master external device output response time, generally tEXT_MOV +tLINE_DELAY (See Note 7).
6. Where tMASTER_IN is the master external device input constraint, generally tEXT_MIS +tLINE_DELAY (See Note 7).
7. tLINE_DELAY is the transmission line time delay.
8. tEXT_MIS is the input constraint for the master external device.
9. tAPBC is the APB period for SERCOM.

Figure 47-4. SPI Timing Requirements in Master Mode