Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

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

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	ARM® Cortex®-M0+
Core Size	32-Bit Single-Core
Speed	48MHz
Connectivity	I²C, LINbus, SCI, SPI, UART/USART, USB
Peripherals	DMA, POR, PWM, WDT
Number of I/O	37
Program Memory Size	128KB (128K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	16K x 8
Voltage - Supply (Vcc/Vdd)	1.62V ~ 3.63V
Data Converters	A/D 14x12b; D/A 2x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	48-TQFP
Supplier Device Package	48-TQFP (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atsaml21g17b-ant

12.2. Nested Vector Interrupt Controller

12.2.1. Overview

The Nested Vectored Interrupt Controller (NVIC) in the SAM L21 supports 32 interrupt lines with four different priority levels. For more details, refer to the Cortex-M0+ Technical Reference Manual (http://www.arm.com).

12.2.2. Interrupt Line Mapping

Each of the 28 interrupt lines is connected to one peripheral instance, as shown in the table below. Each peripheral can have one or more interrupt flags, located in the peripheral's Interrupt Flag Status and Clear (INTFLAG) register.

An interrupt flag is set when the interrupt condition occurs. Each interrupt in the peripheral can be individually enabled by writing a 1 to the corresponding bit in the peripheral's Interrupt Enable Set (INTENSET) register, and disabled by writing 1 to the corresponding bit in the peripheral's Interrupt Enable Clear (INTENCLR) register.

An interrupt request is generated from the peripheral when the interrupt flag is set and the corresponding interrupt is enabled.

The interrupt requests for one peripheral are ORed together on system level, generating one interrupt request for each peripheral. An interrupt request will set the corresponding interrupt pending bit in the NVIC interrupt pending registers (SETPEND/CLRPEND bits in ISPR/ICPR).

For the NVIC to activate the interrupt, it must be enabled in the NVIC interrupt enable register (SETENA/CLRENA bits in ISER/ICER). The NVIC interrupt priority registers IPR0-IPR7 provide a priority field for each interrupt.

Table 12-3. Interrupt Line Mapping

Peripheral source	NVIC line
EIC NMI – External Interrupt Controller	NMI
PM – Power Manager MCLK - Main Clock OSCCTRL - Oscillators Controller OSC32KCTRL - 32KHz Oscillators Controller SUPC - Supply Controller PAC - Protecion Access Controller	0
WDT – Watchdog Timer	1
RTC – Real Time Counter	2
EIC – External Interrupt Controller	3
NVMCTRL – Non-Volatile Memory Controller	4
DMAC - Direct Memory Access Controller	5
USB - Universal Serial Bus	6
EVSYS – Event System	7

Table 12-7. Low-Power Bus Matrix Slaves

Low-Power Bus Matrix Slaves	Slave ID
AHB-APB Bridge A	0
AHB-APB Bridge C	1
AHB-APB Bridge D	2
AHB-APB Bridge E	3
LP SRAM Port 2- H2LBRIDGEM access	5
LP SRAM Port 1- DMAC access	7
L2HBRIDGES - Low-Power to High-Speed bus matrix AHB to AHB bridge	8
HS SRAM Port 2- HMATRIXLP access	9

12.4.3. SRAM Quality of Service

To ensure that masters with latency requirements get sufficient priority when accessing RAM, priority levels can be assigned to the masters for different types of access.

The Quality of Service (QoS) level is independently selected for each master accessing the RAM. For any access to the RAM, the RAM also receives the QoS level. The QoS levels and their corresponding bit values for the QoS level configuration are shown in the following table.

Table 12-8. Quality of Service

Value	Name	Description
0x0	DISABLE	Background (no sensitive operation)
0x1	LOW	Sensitive Bandwidth
0x2	MEDIUM	Sensitive Latency
0x3	HIGH	Critical Latency

If a master is configured with QoS level DISABLE (0x0) or LOW (0x1) there will be a minimum latency of one cycle for the RAM access.

The priority order for concurrent accesses are decided by two factors. First, the QoS level for the master and second, a static priority given by the port ID. The lowest port ID has the highest static priority. See the tables below for details.

The MTB has a fixed QoS level HIGH (0x3).

The CPU QoS level can be written/read, using 32-bit access only, at address 0x41008114 bits [1:0]. Its reset value is 0x3.

Refer to different master QOSCTRL registers for configuring QoS for the other masters (USB, DMAC).

Figure 15-5. Conceptual 64-bit Peripheral ID

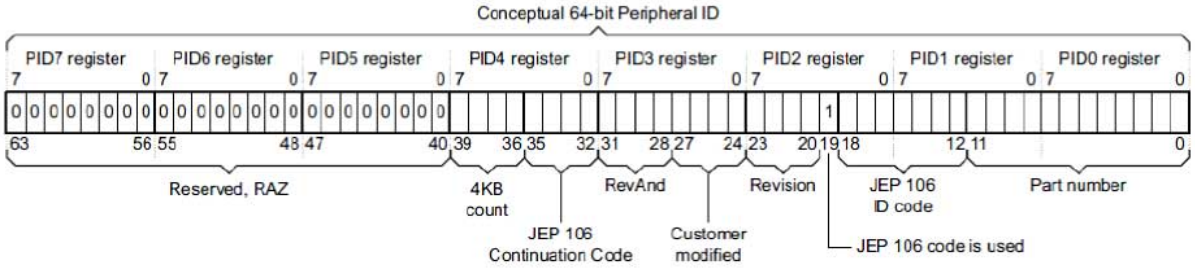


Table 15-2. Conceptual 64-Bit Peripheral ID Bit Descriptions

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

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

15.10.2. Chip Identification Method

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

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

15.11. Functional Description

15.11.1. 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.

Offset	Name	Bit Pos.								
0x40	GENCTRLn8	7:0				SRC[4:0]				
0x41		15:8			RUNSTDBY	DIVSEL	OE	OOV	IDC	GENEN
0x42		23:16	DIV[7:0]							
0x43		31:24	DIV[15:8]							
0x44 ... 0x7F	Reserved									
0x80	PCHCTRL0	7:0	WRTLOCK	CHEN			GEN[3:0]			
0x81		15:8								
0x82		23:16								
0x83		31:24								
0x84	PCHCTRL1	7:0	WRTLOCK	CHEN			GEN[3:0]			
0x85		15:8								
0x86		23:16								
0x87		31:24								
0x88	PCHCTRL2	7:0	WRTLOCK	CHEN			GEN[3:0]			
0x89		15:8								
0x8A		23:16								
0x8B		31:24								
0x8C	PCHCTRL3	7:0	WRTLOCK	CHEN			GEN[3:0]			
0x8D		15:8								
0x8E		23:16								
0x8F		31:24								
0x90	PCHCTRL4	7:0	WRTLOCK	CHEN			GEN[3:0]			
0x91		15:8								
0x92		23:16								
0x93		31:24								
0x94	PCHCTRL5	7:0	WRTLOCK	CHEN			GEN[3:0]			
0x95		15:8								
0x96		23:16								
0x97		31:24								
0x98	PCHCTRL6	7:0	WRTLOCK	CHEN			GEN[3:0]			
0x99		15:8								
0x9A		23:16								
0x9B		31:24								
0x9C	PCHCTRL7	7:0	WRTLOCK	CHEN			GEN[3:0]			
0x9D		15:8								
0x9E		23:16								
0x9F		31:24								
0xA0	PCHCTRL8	7:0	WRTLOCK	CHEN			GEN[3:0]			
0xA1		15:8								
0xA2		23:16								
0xA3		31:24								
0xA4	PCHCTRL9	7:0	WRTLOCK	CHEN			GEN[3:0]			
0xA5		15:8								
0xA6		23:16								
0xA7		31:24								

Offset	Name	Bit Pos.								
0xD4	PCHCTRL21	7:0	WRTLOCK	CHEN			GEN[3:0]			
0xD5		15:8								
0xD6		23:16								
0xD7		31:24								
0xD8	PCHCTRL22	7:0	WRTLOCK	CHEN			GEN[3:0]			
0xD9		15:8								
0xDA		23:16								
0xDB		31:24								
0xDC	PCHCTRL23	7:0	WRTLOCK	CHEN			GEN[3:0]			
0xDD		15:8								
0xDE		23:16								
0xDF		31:24								
0xE0	PCHCTRL24	7:0	WRTLOCK	CHEN			GEN[3:0]			
0xE1		15:8								
0xE2		23:16								
0xE3		31:24								
0xE4	PCHCTRL25	7:0	WRTLOCK	CHEN			GEN[3:0]			
0xE5		15:8								
0xE6		23:16								
0xE7		31:24								
0xE8	PCHCTRL26	7:0	WRTLOCK	CHEN			GEN[3:0]			
0xE9		15:8								
0xEA		23:16								
0xEB		31:24								
0xEC	PCHCTRL27	7:0	WRTLOCK	CHEN			GEN[3:0]			
0xED		15:8								
0xEE		23:16								
0xEF		31:24								
0xF0	PCHCTRL28	7:0	WRTLOCK	CHEN			GEN[3:0]			
0xF1		15:8								
0xF2		23:16								
0xF3		31:24								
0xF4	PCHCTRL29	7:0	WRTLOCK	CHEN			GEN[3:0]			
0xF5		15:8								
0xF6		23:16								
0xF7		31:24								
0xF8	PCHCTRL30	7:0	WRTLOCK	CHEN			GEN[3:0]			
0xF9		15:8								
0xFA		23:16								
0xFB		31:24								
0xFC	PCHCTRL31	7:0	WRTLOCK	CHEN			GEN[3:0]			
0xFD		15:8								
0xFE		23:16								
0xFF		31:24								

The sleep modes (idle, standby, backup, and off) and their effect on the clocks activity, the regulator and the NVM state are described in the table and the sections below. Refer to Power Domain Controller for the power domain gating effect.

Table 20-2. Sleep Mode Overview

Mode	Main clock	CPU	AHBx and APBx clock	GCLK clocks	Oscillators		Regulator	NVM
					ONDEMAND = 0	ONDEMAND = 1		
Active	Run	Run	Run	Run(3)	Run	Run if requested	MAINVREG	active
IDLE	Run	Stop	Stop(1)	Run(3)	Run	Run if requested	MAINVREG	active
STANDBY	Stop(1)	Stop	Stop(1)	Stop(1)	Run if requested or RUNSTDBY=1	Run if requested	MAINVREG in low power mode	Ultra Low power
BACKUP	Stop	Stop	Stop	Stop	Stop	Stop	Backup regulator (ULPVREG)	OFF
OFF	Stop	Stop	Stop	OFF	OFF	OFF	OFF	OFF

Note:

1. Running if requested by peripheral during SleepWalking.
2. Running during SleepWalking.
3. Following On-Demand Clock Request principle.

IDLE Mode

The IDLE mode allows power optimization with the fastest wake-up time.

The CPU is stopped, and peripherals are still working. As in active mode, the AHBx and APBx clocks for peripheral are still provided if requested. As the main clock source is still running, wake-up time is very fast.

- Entering IDLE mode: The IDLE mode is entered by executing the WFI instruction. Additionally, if the SLEEPONEXIT bit in the ARM Cortex System Control register (SCR) is set, the IDLE mode will be entered when the CPU exits the lowest priority ISR (Interrupt Service Routine, see ARM Cortex documentation for details). This mechanism can be useful for applications that only require the processor to run when an interrupt occurs. Before entering the IDLE mode, the user must select the idle Sleep Mode in the Sleep Configuration register (SLEEPCFG.SLEEPMODE=IDLE).
- Exiting IDLE mode: The processor wakes the system up when it detects any non-masked interrupt with sufficient priority to cause exception entry. The system goes back to the ACTIVE mode. The CPU and affected modules are restarted.

GCLK clocks, regulators and RAM are not affected by the idle sleep mode and operate in normal mode.

STANDBY Mode

The STANDBY mode is the lowest power configuration while keeping the state of the logic and the content of the RAM.

In this mode, all clocks are stopped except those configured to be running sleepwalking tasks. The clocks can also be active on request or at all times, depending on their on-demand and run-in-standby settings. Either synchronous (CLK_APBx or CLK_AHBx) or generic (GCLK_x) clocks or both can be involved in sleepwalking tasks. This is the case when for example the SERCOM RUNSTDBY bit is written to '1'.

- Entering STANDBY mode: This mode is entered by executing the WFI instruction after writing the Sleep Mode bit in the Sleep Configuration register (SLEEPCFG.SLEEPMODE=STANDBY). The SLEEPONEXIT feature is also available as in IDLE mode.

25.7. Register Summary - COUNT32

Offset	Name	Bit Pos.								
0x00	CTRLA	7:0	MATCHCLR				MODE[1:0]		ENABLE	SWRST
0x01		15:8	COUNTSYNC				PRESCALER[3:0]			
0x02 ... 0x03	Reserved									
0x04	EVCTRL	7:0	PEREO7	PEREO6	PEREO5	PEREO4	PEREO3	PEREO2	PEREO1	PEREO0
0x05		15:8	OVFEO							CMPEO0
0x06		23:16								
0x07		31:24								
0x08	INTENCLR	7:0	PER7	PER6	PER5	PER4	PER3	PER2	PER1	PER0
0x09		15:8	OVF							CMP0
0x0A	INTENSET	7:0	PER7	PER6	PER5	PER4	PER3	PER2	PER1	PER0
0x0B		15:8	OVF							CMP0
0x0C	INTFLAG	7:0	PER7	PER6	PER5	PER4	PER3	PER2	PER1	PER0
0x0D		15:8	OVF							CMP0
0x0E	DBGCTRL	7:0								DBGRUN
0x0F	Reserved									
0x10	SYNCBUSY	7:0			COMP0		COUNT	FREQCORR	ENABLE	SWRST
0x11		15:8	COUNTSYNC							
0x12		23:16				GP1	GP0			
0x13		31:24								
0x14	FREQCORR	7:0	SIGN		VALUE[5:0]					
0x15 ... 0x17	Reserved									
0x18	COUNT	7:0	COUNT[7:0]							
0x19		15:8	COUNT[15:8]							
0x1A		23:16	COUNT[23:16]							
0x1B		31:24	COUNT[31:24]							
0x1C ... 0x1F	Reserved									
0x20	COMP0	7:0	COMP[7:0]							
0x21		15:8	COMP[15:8]							
0x22		23:16	COMP[23:16]							
0x23		31:24	COMP[31:24]							
0x24 ... 0x3F	Reserved									
0x40	GP0	7:0	GP[7:0]							
0x41		15:8	GP[15:8]							
0x42		23:16	GP[23:16]							
0x43		31:24	GP[31:24]							

25.10. Register Description - COUNT16

This Register Description section is valid if the RTC is in COUNT16 mode (CTRLA.MODE=1).

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.

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:

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

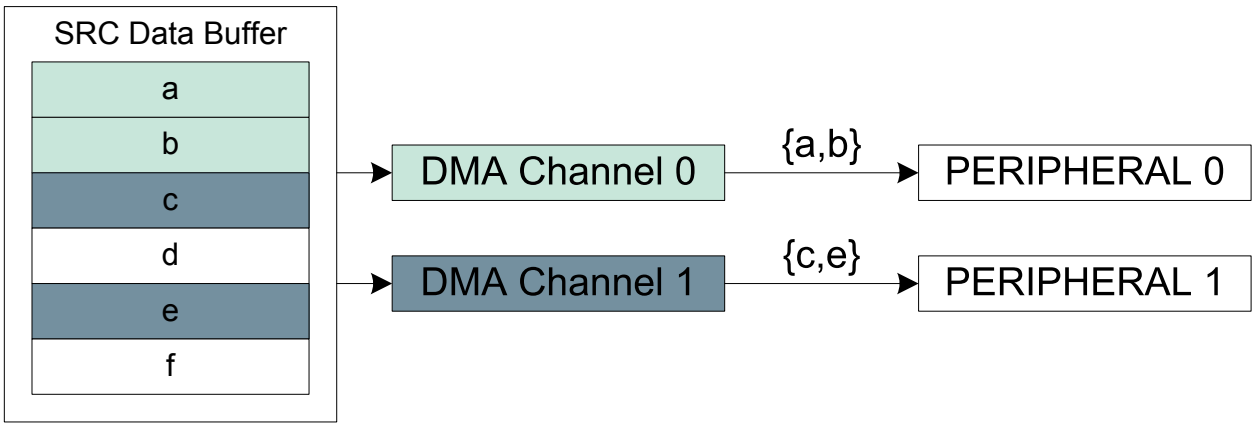
If BTCTRL.STEPSEL=0:

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

- $\text{SRCADDR}_{\text{START}}$ 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 26-8. Source Address Increment



Incrementation for the destination address of a block transfer is enabled by setting the Destination Address Incrementation Enable bit in the Block Transfer Control register (BTCTRL.DSTINC=1). The step size of the incrementation is configurable by clearing BTCTRL.STEPSEL=0 and writing BTCTRL.STEPSIZE to the desired step size. If BTCTRL.STEPSEL=1, the step size for the destination incrementation will be the size of one beat.

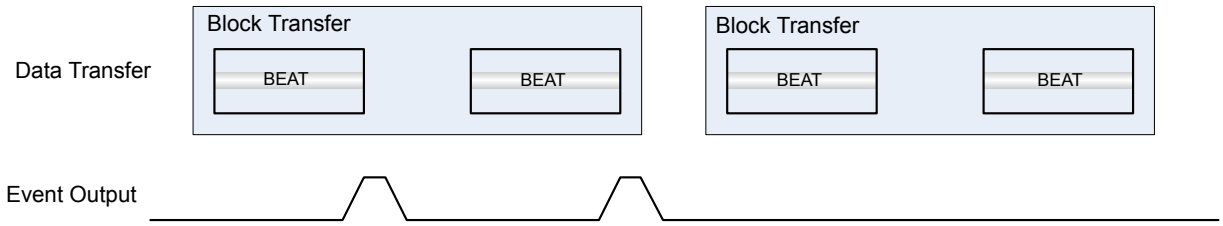
When the destination address incrementation is configured (BTCTRL.DSTINC=1), SRCADDR must be set and calculated as follows:

$DSTADDR = DSTADDR_{START} + BTCNT \bullet \left(BEATSIZE + 1\right) \bullet 2^{STEPSIZE}$	where BTCTRL.STEPSEL is zero
$DSTADDR = DSTADDR_{START} + BTCNT \bullet (BEATSIZE + 1)$	where BTCTRL.STEPSEL is one

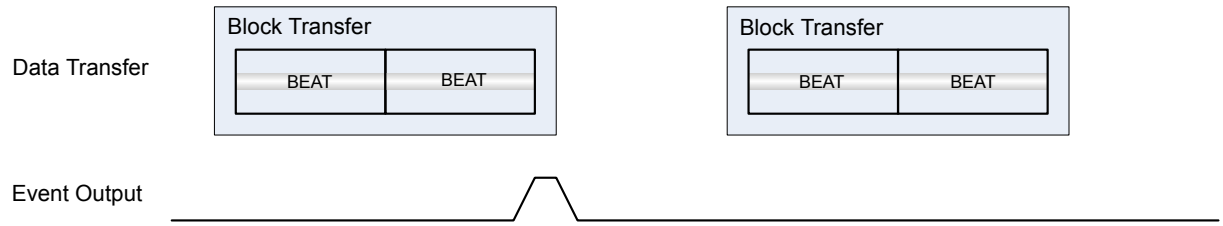
- $\text{DSTADDR}_{\text{START}}$ is the destination address of the first beat transfer in the block transfer
- BTCNT is the initial number of beats remaining in the block transfer

Figure 26-15. Event Output Generation

Beat Event Output



Block Event Output



26.6.3.6. Aborting Transfers

Transfers on any channel can be aborted gracefully by software by disabling the corresponding DMA channel. It is also possible to abort all ongoing or pending transfers by disabling the DMAC.

When a DMA channel disable request or DMAC disable request is detected:

- Ongoing transfers of the active channel will be disabled when the ongoing beat transfer is completed and the write-back memory section is updated. This prevents transfer corruption before the channel is disabled.
- All other enabled channels will be disabled in the next clock cycle.

The corresponding Channel Enable bit in the Channel Control A register is cleared (CHCTRLA.ENABLE=0) when the channel is disabled.

The corresponding DMAC Enable bit in the Control register is cleared (CTRL.DMAENABLE=0) when the entire DMAC module is disabled.

26.6.3.7. CRC Operation

A cyclic redundancy check (CRC) is an error detection technique used to find errors in data. It is commonly used to determine whether the data during a transmission, or data present in data and program memories has been corrupted or not. A CRC takes a data stream or a block of data as input and generates a 16- or 32-bit output that can be appended to the data and used as a checksum.

When the data is received, the device or application repeats the calculation: If the new CRC result does not match the one calculated earlier, the block contains a data error. The application will then detect this and may take a corrective action, such as requesting the data to be sent again or simply not using the incorrect data.

The CRC engine in DMAC supports two commonly used CRC polynomials: CRC-16 (CRC-CCITT) and CRC-32 (IEEE 802.3). Typically, applying CRC-n (CRC-16 or CRC-32) to a data block of arbitrary length will detect any single alteration that is ≤n bits in length, and will detect the fraction 1-2-n of all longer error bursts.

29.8.9. Data Input Value

Name: IN
Offset: 0x20
Reset: 0x00000000
Property: -

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

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

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

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

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

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

Value	Name	Description
0x2	COUNT32	Counter in 32-bit mode
0x3	-	Reserved

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

Value	Description
0	The peripheral is disabled.
1	The peripheral is enabled.

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

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

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

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

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

Bits 20, 21 – COPEN0, COPEN1: Capture On Pin x Enable [x = 1..0]
This bit selects the trigger source for capture operation, either events or I/O pin input.

Value	Description
0	Event from Event System is selected as trigger source for capture operation on channel x.
1	I/O pin is selected as trigger source for capture operation on channel x.

Bits 16, 17 – CAPTEN0, CAPTEN1: Capture Channel x Enable [x = 1..0]
These bits are used to select whether channel x is a capture or a compare channel.

These bits are not synchronized.

Value	Description
0	CAPTENx disables capture on channel x.
1	CAPTENx enables capture on channel x.

35.8.5. Interrupt Enable Clear

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

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

Bit	7	6	5	4	3	2	1	0
			MC1	MC0			ERR	OVF
Access			R/W	R/W			R/W	R/W
Reset			0	0			0	0

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

Writing a '1' to this bit will clear the Error Interrupt Enable bit, which disables the Error interrupt.

Value	Description
0	The Error interrupt is disabled.
1	The Error interrupt is enabled.

Bit 0 – OVF: Overflow Interrupt Enable
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Overflow Interrupt Enable bit, which disables the Overflow interrupt request.

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

Bits 5,4 – MCx: Match or Capture Channel x Interrupt Enable [x = 1..0]
Writing a '0' to these bits has no effect.

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

Value	Description
0	The Match or Capture Channel x interrupt is disabled.
1	The Match or Capture Channel x interrupt is enabled.

37.7. Register Summary

Offset	Name	Bit Pos.								
0x00	CTRLA	7:0		RUNSTDBY					ENABLE	
0x01 ... 0x03	Reserved									
0x04	EVCTRL	7:0								DATARDYEO
0x05 ... 0x07	Reserved									
0x08	INTENCLR	7:0								DATARDY
0x09	INTENSET	7:0								DATARDY
0x0A	INTFLAG	7:0								DATARDY
0x0B ... 0x1F	Reserved									
0x20	DATA	7:0	DATA[7:0]							
0x21		15:8	DATA[15:8]							
0x22		23:16	DATA[23:16]							
0x23		31:24	DATA[31:24]							

37.8. 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.

Refer to *PAC - Peripheral Access Controller* and Synchronizationfor details.

Related Links

PAC - Peripheral Access Controller on page 59

40.8.2. Sequential Control x

Name: SEQCTRLn
Offset: 0x04 + n*0x01 [n=0..1]
Reset: 0x00
Property: PAC Write-Protection, Enable-Protected

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

Bits 3:0 – SEQSEL[3:0]: Sequential Selection
These bits select the sequential configuration:

Sequential Selection

Value	Name	Description
0x0	DISABLE	Sequential logic is disabled
0x1	DFF	D flip flop
0x2	JK	JK flip flop
0x3	LATCH	D latch
0x4	RS	RS latch
0x5 - 0xF		Reserved

Value	Name	Description
0x0	Mode 0	Minimum current consumption, but the slowest mode
0x1	Mode 1	Low current consumption, slow speed
0x2	Mode 2	High current consumption, fast speed
0x3	Mode 3	Maximum current consumption but the fastest mode

Bit 2 – ANAOUT: Analog Output
This bit controls a switch connected to the OPAMP output.

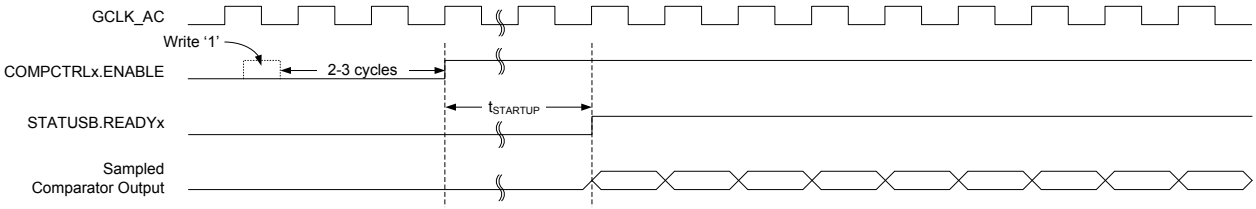
Value	Description
0	Swith open. No ADC or AC connection.
1	Switch closed. OPAMP output is connected to the ADC or AC input.

Bit 1 – ENABLE: Operational Amplifier Enable

Value	Description
0	The OPAMPx is disabled
1	The OPAMPx is enabled

otherwise GCLK_AC is disabled until the next edge detection. Filtering is not possible with this configuration.

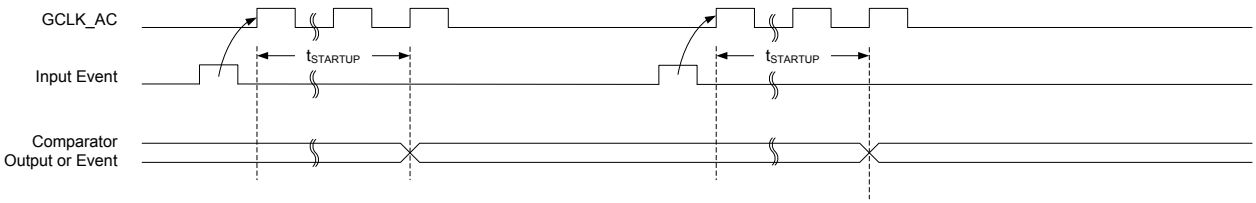
Figure 43-9. Continuous Mode SleepWalking



43.6.14.2. Single-Shot Measurement during Sleep

For low-power operation, event-triggered measurements can be performed during sleep modes. When the event occurs, the Power Manager will start GCLK_AC. The comparator is enabled, and after the start-up time has passed, a comparison is done, with filtering if desired, and the appropriate peripheral events and interrupts are also generated, as shown in Figure 43-10. The comparator and GCLK_AC are then disabled again automatically, unless configured to wake the system from sleep. Filtering is allowed with this configuration.

Figure 43-10. Single-Shot SleepWalking



43.6.15. Synchronization

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

The following bits are synchronized when written:

- Software Reset bit in control register (CTRLA.SWRST)
- Enable bit in control register (CTRLA.ENABLE)
- Enable bit in Comparator Control register (COMPCTRLn.ENABLE)

The following registers are synchronized when written:

- Window Control register (WINCTRL)

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

Related Links

Register Synchronization on page 129

44. DAC – Digital-to-Analog Converter

44.1. Overview

The Digital-to-Analog Converter (DAC) converts a digital value to a voltage. The DAC Controller controls two DACs, which can operate either as two independent DACs or as a single DAC in differential mode. Each DAC is 12-bit resolution and it is capable of converting up to 1,000,000 samples per second (MSPS).

44.2. Features

- Two independent DACs or single DAC in differential mode
- DAC with 12-bit resolution
- Up to 1MSPS conversion rate
- Hardware support for 16-bit using dithering
- Multiple trigger sources
- High-drive capabilities
- DAC0 used as internal input
- DMA support

44.3. Block Diagram

Figure 44-1. DAC Controller Block Diagram.

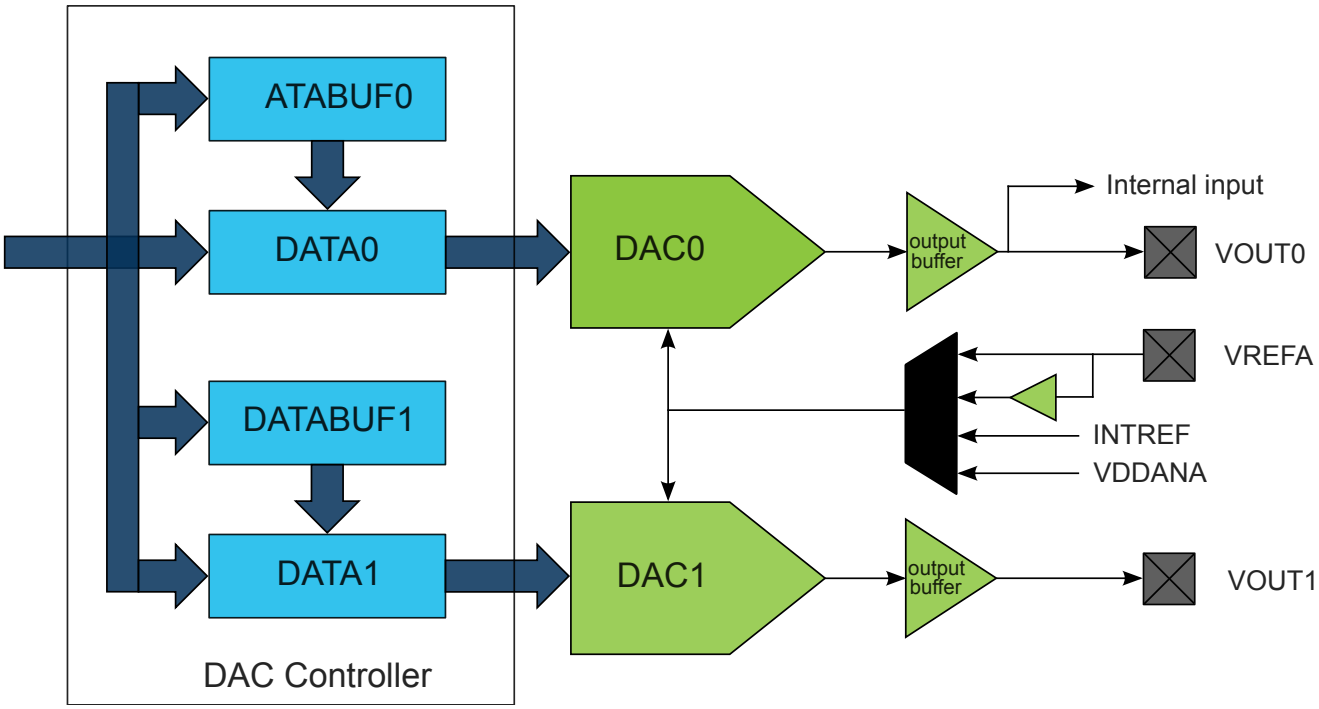


Table 46-36. Power Consumption(1)

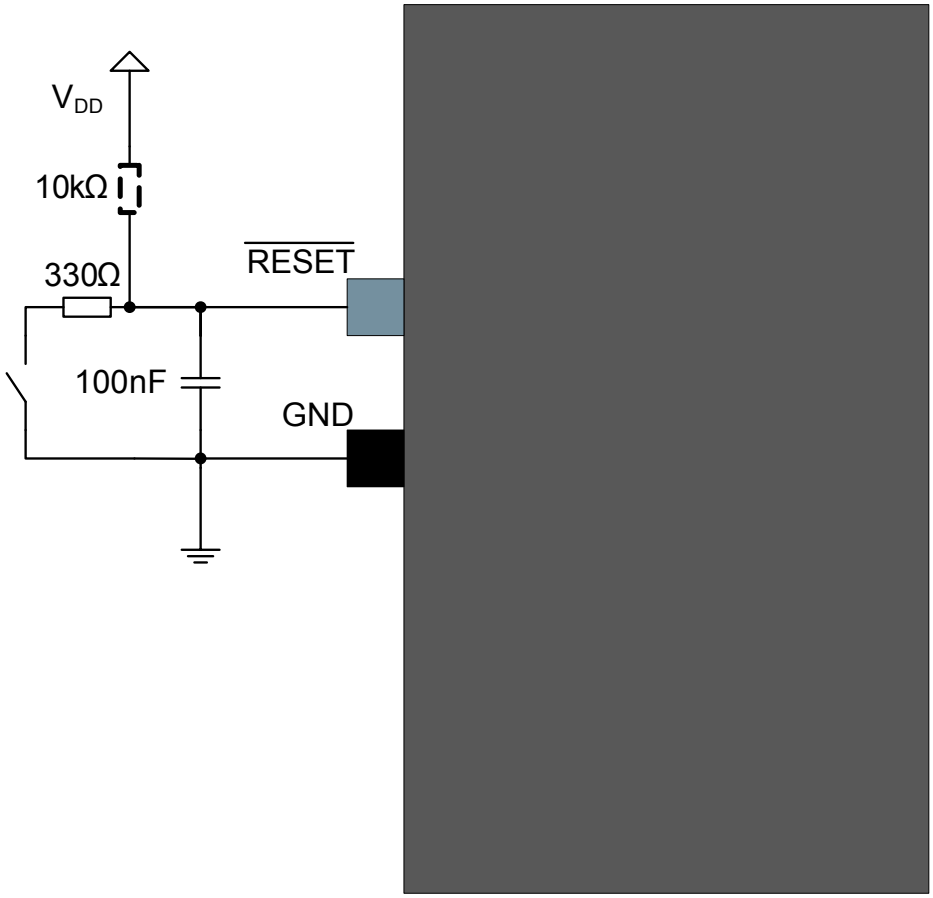
Symbol	Parameters	Conditions	Ta	Min.	Typ.	Max.	Unit
IDD	DC supply current (Voltage Doubler OFF)	Mode 3,VCC =3.3V	Max 85°C Typ 25°C	-	184	312	uA
		Mode 2,VCC =3.3V		-	72	127	
		Mode 1,VCC =3.3V		-	21	33	
		Mode 0 ,VCC =3.3V		-	6	9	
	Voltage Doubler consumption	VCC =3.3V		-	0,69	1,5	

Note: 1. These values are based on characterization.

Table 46-37. Static Characteristics in 1X Gain(1)

Symbol	Parameters	Conditions	Min.	Typ.	Max.	Unit
G0	Open loop gain	Mode 3	-	114.5	-	dB
		Mode 2	-	117.6	-	
		Mode 1	-	116.8	-	
		Mode 0	-	108.5	-	
GBW	Gain Bandwidth	Mode 3	-	7.1	-	MHz
		Mode 2	-	2.8	-	
		Mode 1	-	0.85	-	
		Mode 0	-	0.2	-	
Φ_m	Phase margin	Mode 3	-	71.5	-	deg
		Mode 2	-	64	-	
		Mode 1	-	56	-	
		Mode 0	-	52	-	
T_{r1}	Response Time at 240µV (X1 gain)	Mode 3	-	1.3	-	µs
		Mode 2	-	3.3	-	
		Mode 1	-	13	-	
		Mode 0	-	52	-	
ΔT_{r1}	Response Time Variation for 10mV	Mode 3	-	100	-	ns
		Mode 2	-	-	-	
		Mode 1	-	-	-	
		Mode 0	-	-	-	
T_{start}	Start-up time (Enable to Ready), (Voltage Doubler OFF)	Mode 3	-	2.7	-	µs
		Mode 2	-	6.35	-	
		Mode 1	-	21.5	-	
		Mode 0	-	88.5	-	
Oe	Input Offset Voltage		-	-	+-3.5	mV

Figure 49-6. External Reset Circuit Schematic



A pull-up resistor makes sure that the reset does not go low and unintentionally causing a device reset. An additional resistor has been added in series with the switch to safely discharge the filtering capacitor, i.e. preventing a current surge when shorting the filtering capacitor which again can cause a noise spike that can have a negative effect on the system.

Table 49-3. Reset Circuit Connections

Signal Name	Recommended Pin Connection	Description
$\overline{\text{RESET}}$	Reset low level threshold voltage V_{DDIO} = 1.6V - 2.0V: Below 0.33 * V_{DDIO} V_{DDIO} = 2.7V - 3.6V: Below 0.36 * V_{DDIO} Decoupling/filter capacitor 100nF(1) Pull-up resistor 10kΩ(1)(2) Resistor in series with the switch 330Ω(1)	Reset pin

1. These values are only given as a typical example.

2. The SAM L21 features an internal pull-up resistor on the $\overline{\text{RESET}}$ pin, hence an external pull-up is optional.