

Welcome to [E-XFL.COM](https://www.e-xfl.com)

### What is "[Embedded - Microcontrollers](#)"?

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

### Applications of "[Embedded - Microcontrollers](#)"

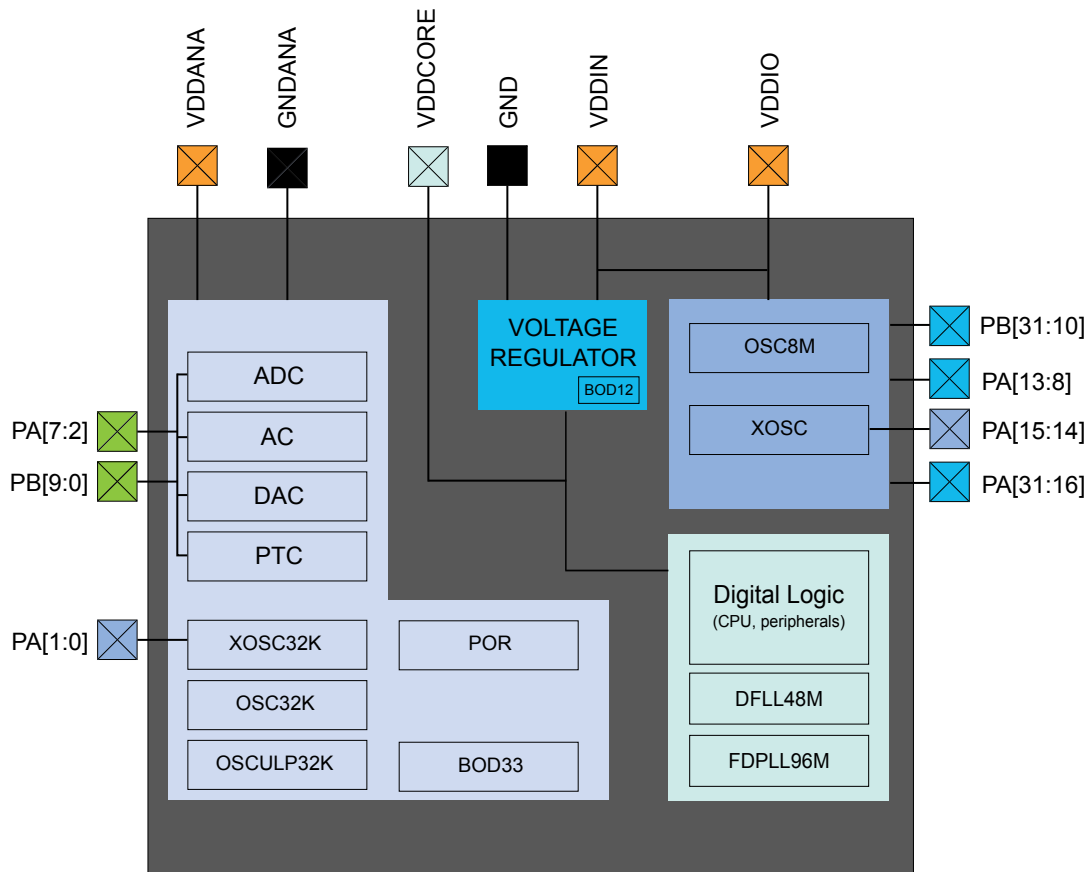
#### Details

Product Status	Active
Core Processor	ARM® Cortex®-M0+
Core Size	32-Bit Single-Core
Speed	48MHz
Connectivity	I <sup>2</sup> C, LINbus SBC, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, DMA, POR, PWM, WDT
Number of I/O	32
Program Memory Size	16KB (16K x 8)
Program Memory Type	FLASH
EEPROM Size	512 x 8
RAM Size	4K x 8
Voltage - Supply (Vcc/Vdd)	0V ~ 3.8V
Data Converters	A/D 13x12b; D/A 1x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TC)
Mounting Type	Surface Mount, Wettable Flank
Package / Case	48-VFQFN Exposed Pad
Supplier Device Package	48-VQFN (7x7)
Purchase URL	<a href="https://www.e-xfl.com/product-detail/microchip-technology/atsamha1g14a-mbt-bvao">https://www.e-xfl.com/product-detail/microchip-technology/atsamha1g14a-mbt-bvao</a>

Trademarks.....	889
Quality Management System Certified by DNV.....	889
Worldwide Sales and Service.....	890

## 8. Power Supply and Start-Up Considerations

### 8.1 Power Domain Overview



## 8.2 Power Supply Considerations

### 8.2.1 Power Supplies

The device has several different power supply pins:

- VDDIO: Powers I/O lines, OSC8M and XOSC. Voltage is 2.7V to 3.63V.
- VDDIN: Powers I/O lines and the internal regulator. Voltage is 2.7V to 3.63V.
- VDDANA: Powers I/O lines and the ADC, AC, DAC, PTC, OSCULP32K, OSC32K, XOSC32K. Voltage is 2.7V to 3.63V.
- VDDCORE: Internal regulated voltage output. Powers the core, memories, peripherals, FDPLL96M, and DFLL48M. Voltage is 1.2V.

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

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

**Figure 15-3. Hot-Plugging Detection Timing Diagram**



The presence of a debugger probe is detected when either Hot-Plugging or Cold-Plugging is detected. Once detected, the Debugger Present bit of the Status B register (STATUSB.DBGPRES) is set. For security reasons, Hot-Plugging is not available when the device is protected by the NVMCTRL security bit.

This detection requires that pads are correctly powered. Thus, at cold startup, this detection cannot be done until POR is released. If the device is protected, Cold-Plugging is the only way to detect a debugger probe, and so the external reset timing must be longer than the POR timing. If external reset is deasserted before POR release, the user must retry the procedure above until it gets connected to the device.

#### Related Links

[NVMCTRL – Nonvolatile Memory Controller Security Bit](#)

## 15.7 Chip Erase

Chip-Erase consists of removing all sensitive information stored in the chip and clearing the NVMCTRL security bit. Therefore, all volatile memories and the Flash memory (including the EEPROM emulation area) will be erased. The Flash auxiliary rows, including the user row, will not be erased.

When the device is protected, the debugger must first reset the device in order to be detected. This ensures that internal registers are reset after the protected state is removed. The Chip-Erase operation is triggered by writing a '1' to the Chip-Erase bit in the Control register (CTRL.CE). This command will be discarded if the DSU is protected by the Peripheral Access Controller (PAC). Once issued, the module clears volatile memories prior to erasing the Flash array. To ensure that the Chip-Erase operation is completed, check the Done bit of the Status A register (STATUSA.DONE).

The Chip-Erase operation depends on clocks and power management features that can be altered by the CPU. For that reason, it is recommended to issue a Chip-Erase after a Cold-Plugging procedure to ensure that the device is in a known and safe state.

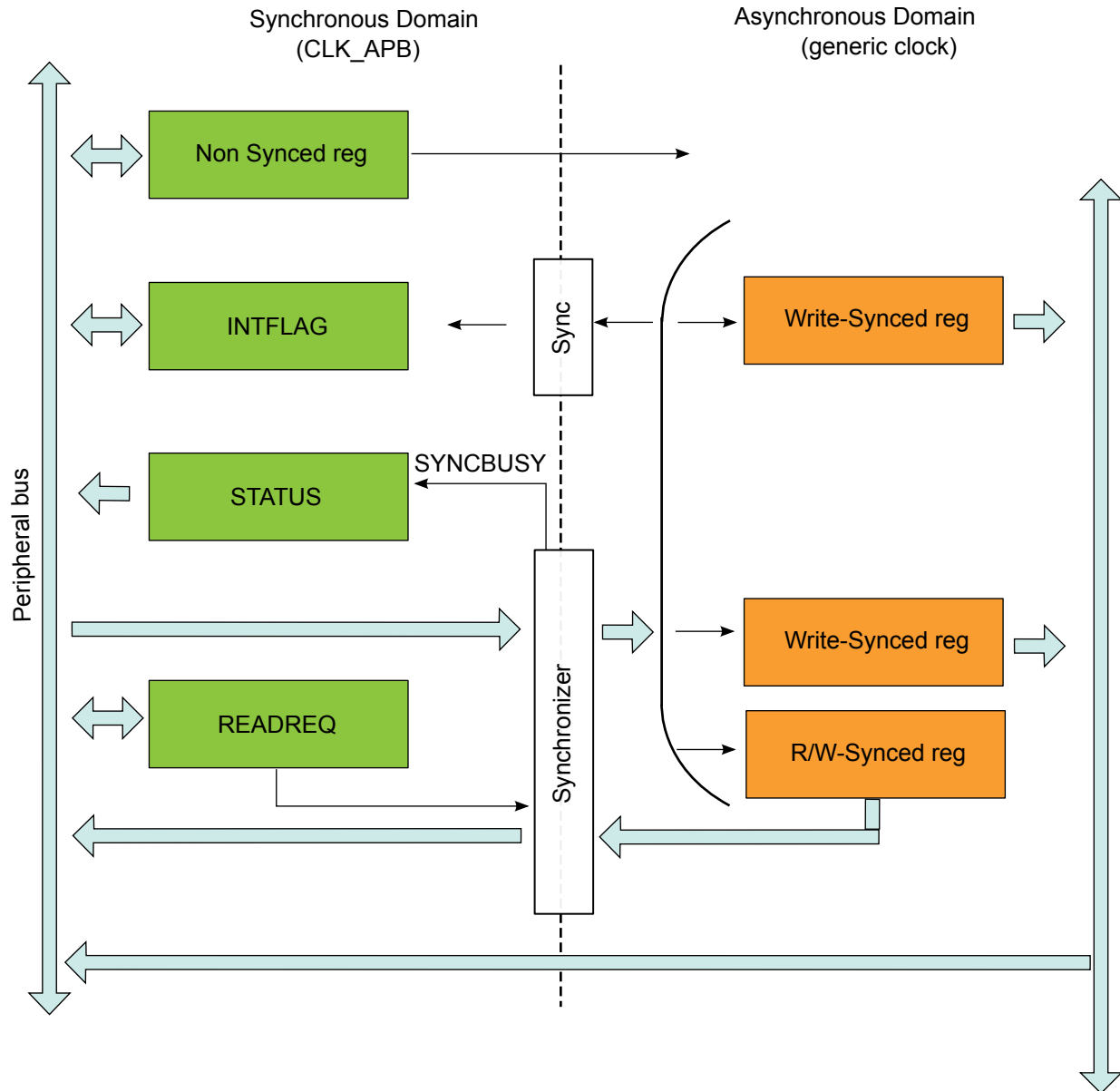
The recommended sequence is as follows:

1. Issue the Cold-Plugging procedure (refer to [Cold Plugging](#)). The device then:
  - 1.1. Detects the debugger probe.
  - 1.2. Holds the CPU in reset.
2. Issue the Chip-Erase command by writing a '1' to CTRL.CE. The device then:
  - 2.1. Clears the system volatile memories.

Each individual register description will have the properties "Read-Synchronized" and/or "Write-Synchronized" if a register is synchronized.

As shown in the figure below, the common synchronizer is used for all registers in one peripheral. Therefore, status register (STATUS) of each peripheral can be synchronized at a time.

**Figure 16-3. Synchronization**



### 16.3.1.2 Write-Synchronization

Write-Synchronization is triggered by writing to a register in the peripheral clock domain. The Synchronization Busy bit in the Status register (STATUS.SYNCBUSY) will be set when the write-synchronization starts and cleared when the write-synchronization is complete. Refer to [Synchronization Delay](#) for details on the synchronization delay.

When the write-synchronization is ongoing (STATUS.SYNCBUSY is one), any of the following actions will cause the peripheral bus to stall until the synchronization is complete:

The XOSC source can be divided inside the FDPLL96M. The user must make sure that the programmable clock divider and XOSC frequency provides a valid CLK\_FDPLL96M\_REF clock frequency that meets the FDPLL96M input frequency range.

The output clock of the FDPLL96M is CLK\_FDPLL96M. The state of the CLK\_FDPLL96M clock only depends on the FDPLL96M internal control of the final clock gater CG.

The FDPLL96M requires a 32kHz clock from the GCLK when the FDPLL96M internal lock timer is used. This clock must be configured and enabled in the Generic Clock Controller before using the FDPLL96M. Refer to *GCLK – Generic Clock Controller* for details.

**Table 19-3. Generic Clock Input for FDPLL96M**

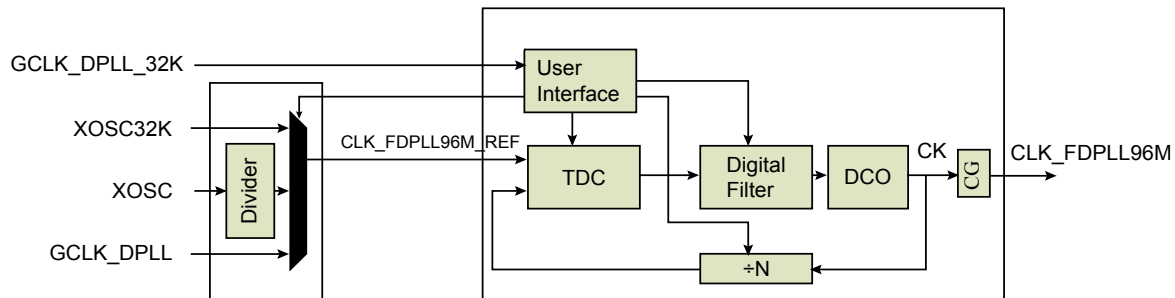
Generic Clock	FDPLL96M
FDPLL96M 32kHz clock	GCLK_DPLL_32K for internal lock timer
FDPLL96M	GCLK_DPLL for CLK_FDPLL96M_REF

**Related Links**

[GCLK - Generic Clock Controller](#)

**19.6.8.2 Block Diagram**

**Figure 19-2. FDPLL96M Block Diagram**



**19.6.8.3 Principle of Operation**

The task of the FDPLL96M is to maintain coherence between the input reference clock signal (CLK\_FDPLL96M\_REF) and the respective output frequency CK via phase comparison. The FDPLL96M supports three independent sources of clocks; XOSC32K, XOSC and GCLK\_DPLL. When the FDPLL96M is enabled, the relationship between the reference clock (CLK\_FDPLL96M\_REF) frequency and the output clock (CLK\_FDPLL96M) frequency is defined below.

$$f_{clk\_fdpll96m} = f_{clk\_fdpll96m\_ref} \times \left( LDR + 1 + \frac{LDRFRAC}{16} \right)$$

Where LDR is the loop divider ratio integer part, LDRFRAC is the loop divider ratio fractional part,  $f_{ckrx}$  is the frequency of the selected reference clock and  $f_{ck}$  is the frequency of the FDPLL96M output clock. As previously stated a clock divider exist between XOSC and CLK\_FDPLL96M\_REF. The frequency between the two clocks is defined below.

$$f_{clk\_fdpll96m\_ref} = f_{xosc} \times \left( \frac{1}{2 \times (DIV + 1)} \right)$$

When the FDPLL96M is disabled, the output clock is reset. If the loop divider ratio fractional part (DPLLATIO.LDRFRAC) field is reset, the FDPLL96M works in integer mode, otherwise the fractional

Value	Description
0	The oscillator is disabled in standby sleep mode.
1	The oscillator is not stopped in standby sleep mode. If XOSC.ONDEMAND is one, the clock source will be running when a peripheral is requesting the clock. If XOSC.ONDEMAND is zero, the clock source will always be running in standby sleep mode.

**Bit 2 – XTALEN** Crystal Oscillator Enable

This bit controls the connections between the I/O pads and the external clock or crystal oscillator:

Value	Description
0	External clock connected on XIN. XOUT can be used as general-purpose I/O.
1	Crystal connected to XIN/XOUT.

**Bit 1 – ENABLE** Oscillator Enable

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

If On Demand is disabled the oscillator will always be running when enabled.

In standby sleep mode, the On Demand operation is still active if the DFLLCTRL.RUNSTDBY bit is one. If DFLLCTRL.RUNSTDBY is zero, the oscillator is disabled.

Value	Description
0	The oscillator is always on, if enabled.
1	The oscillator is enabled when a peripheral is requesting the oscillator to be used as a clock source. The oscillator is disabled if no peripheral is requesting the clock source.

#### Bit 4 – LLAW Lose Lock After Wake

Value	Description
0	Locks will not be lost after waking up from sleep modes if the DFLL clock has been stopped.
1	Locks will be lost after waking up from sleep modes if the DFLL clock has been stopped.

#### Bit 3 – STABLE Stable DFLL Frequency

Value	Description
0	FINE calibration tracks changes in output frequency.
1	FINE calibration register value will be fixed after a fine lock.

#### Bit 2 – MODE Operating Mode Selection

Value	Description
0	The DFLL operates in open-loop operation.
1	The DFLL operates in closed-loop operation.

#### Bit 1 – ENABLE DFLL Enable

Due to synchronization, there is delay from updating the register until the peripheral is enabled/disabled. The value written to DFLLCTRL.ENABLE will read back immediately after written.

Value	Description
0	The DFLL oscillator is disabled.
1	The DFLL oscillator is enabled.



[NVM User Row Mapping](#)

### 22.6.2.6 Transfer Triggers and Actions

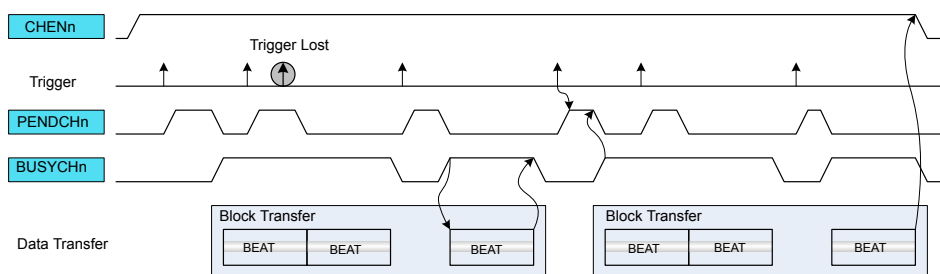
A DMA transfer through a DMA channel can be started only when a DMA transfer request is detected, and the DMA channel has been granted access to the DMA. A transfer request can be triggered from software, from a peripheral, or from an event. There are dedicated Trigger Source selections for each DMA Channel Control B (CHCTRLB.TRIGSRC).

The trigger actions are available in the Trigger Action bit group in the Channel Control B register (CHCTRLB.TRIGACT). By default, a trigger generates a request for a block transfer operation. If a single descriptor is defined for a channel, the channel is automatically disabled when a block transfer has been completed. If a list of linked descriptors is defined for a channel, the channel is automatically disabled when the last descriptor in the list is executed. If the list still has descriptors to execute, the channel will be waiting for the next block transfer trigger. When enabled again, the channel will wait for the next block transfer trigger. The trigger actions can also be configured to generate a request for a beat transfer (CHCTRLB.TRIGACT=0x2) or transaction transfer (CHCTRLB.TRIGACT=0x3) instead of a block transfer (CHCTRLB.TRIGACT=0x0).

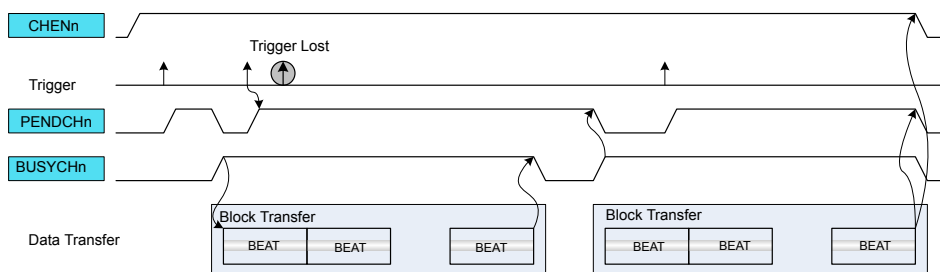
Figure 22-7 shows an example where triggers are used with two linked block descriptors.

**Figure 22-7. Trigger Action and Transfers**

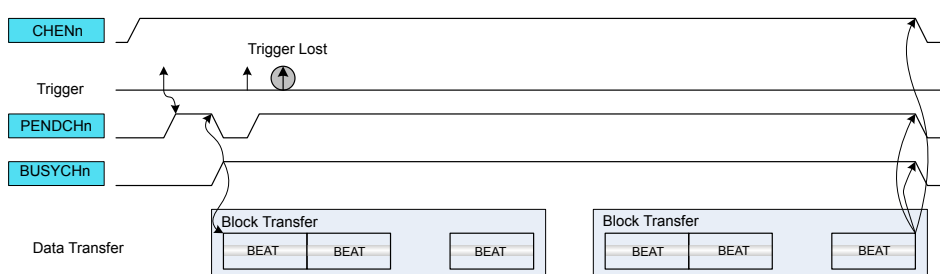
#### Beat Trigger Action



#### Block Trigger Action



#### Transaction Trigger Action



If the trigger source generates a transfer request for a channel during an ongoing transfer, the new transfer request will be kept pending (CHSTATUS.PEND=1), and the new transfer can start after the

## 22.10.1 Block Transfer Control

**Name:** BTCTRL  
**Offset:** 0x00  
**Property:** -

The BTCTRL register offset is relative to (BASEADDR or WRBADDR) + Channel Number \* 0x10

Bit	15	14	13	12	11	10	9	8
	STEPSIZE[2:0]			STEPSEL	DSTINC	SRCINC	BEATSIZE[1:0]	
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
				BLOCKACT[1:0]		EVOSEL[1:0]		VALID
Access								
Reset								

### Bits 15:13 – STEPSIZE[2:0] Address Increment Step Size

These bits select the address increment step size. The setting apply to source or destination address, depending on STEPSEL setting.

Value	Name	Description
0x0	X1	Next ADDR = ADDR + (Beat size in byte) * 1
0x1	X2	Next ADDR = ADDR + (Beat size in byte) * 2
0x2	X4	Next ADDR = ADDR + (Beat size in byte) * 4
0x3	X8	Next ADDR = ADDR + (Beat size in byte) * 8
0x4	X16	Next ADDR = ADDR + (Beat size in byte) * 16
0x5	X32	Next ADDR = ADDR + (Beat size in byte) * 32
0x6	X64	Next ADDR = ADDR + (Beat size in byte) * 64
0x7	X128	Next ADDR = ADDR + (Beat size in byte) * 128

### Bit 12 – STEPSEL Step Selection

This bit selects if source or destination addresses are using the step size settings.

Value	Name	Description
0x0	DST	Step size settings apply to the destination address
0x1	SRC	Step size settings apply to the source address

### Bit 11 – DSTINC Destination Address Increment Enable

Writing a '0' to this bit will disable the destination address incrementation. The address will be kept fixed during the data transfer.

Writing a '1' to this bit will enable the destination address incrementation. By default, the destination address is incremented by 1. If the STEPSEL bit is cleared, flexible step-size settings are available in the STEPSIZE register.

Value	Description
0	The Destination Address Increment is disabled.
1	The Destination Address Increment is enabled.

### 22.10.5 Next Descriptor Address

**Name:** DESCADDR  
**Offset:** 0x0C  
**Property:** -

The DESCADDR register offset is relative to (BASEADDR or WRBADDR) + Channel Number \* 0x10

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

#### Bits 31:0 – DESCADDR[31:0] Next Descriptor Address

This bit group holds the SRAM address of the next descriptor. The value must be 128-bit aligned. If the value of this SRAM register is 0x00000000, the transaction will be terminated when the DMAC tries to load the next transfer descriptor.

#### 24.6.6 Security Bit

The security bit allows the entire chip to be locked from external access for code security. The security bit can be written by a dedicated command, Set Security Bit (SSB). Once set, the only way to clear the security bit is through a debugger Chip Erase command. After issuing the SSB command, the PROGE error bit can be checked.

In order to increase the security level it is recommended to enable the internal BOD33 when the security bit is set.

##### Related Links

[DSU - Device Service Unit](#)

#### 24.6.7 Cache

The NVM Controller cache reduces the device power consumption and improves system performance when wait states are required. Only the NVM main array address space is cached. It is a direct-mapped cache that implements 8 lines of 64 bits (i.e., 64 Bytes). NVM Controller cache can be enabled by writing a '0' to the Cache Disable bit in the Control B register ([CTRLB.CACHEDIS](#)).

The cache can be configured to three different modes using the Read Mode bit group in the Control B register ([CTRLB.READMODE](#)).

The INVALL command can be issued using the Command bits in the Control A register to invalidate all cache lines ([CTRLA.CMD=INVALL](#)). Commands affecting NVM content automatically invalidate cache lines.

---

**Bit 2 – RXC** Receive Complete Interrupt Enable

Writing '0' to this bit has no effect.

Writing '1' to this bit will set the Receive Complete Interrupt Enable bit, which enables the Receive Complete interrupt.

Value	Description
0	Receive Complete interrupt is disabled.
1	Receive Complete interrupt is enabled.

**Bit 1 – TXC** Transmit Complete Interrupt Enable

Writing '0' to this bit has no effect.

Writing '1' to this bit will set the Transmit Complete Interrupt Enable bit, which enables the Transmit Complete interrupt.

Value	Description
0	Transmit Complete interrupt is disabled.
1	Transmit Complete interrupt is enabled.

**Bit 0 – DRE** Data Register Empty Interrupt Enable

Writing '0' to this bit has no effect.

Writing '1' to this bit will set the Data Register Empty Interrupt Enable bit, which enables the Data Register Empty interrupt.

Value	Description
0	Data Register Empty interrupt is disabled.
1	Data Register Empty interrupt is enabled.

### 29.6.5 Sleep Mode Operation

The behavior in sleep mode is depending on the master/slave configuration and the Run In Standby bit in the Control A register (CTRLA.RUNSTDBY):

- Master operation, CTRLA.RUNSTDBY=1: The peripheral clock GCLK\_SERCOM\_CORE will continue to run in idle sleep mode and in standby sleep mode. Any interrupt can wake up the device.
- Master operation, CTRLA.RUNSTDBY=0: GLK\_SERCOMx\_CORE will be disabled after the ongoing transaction is finished. Any interrupt can wake up the device.
- Slave operation, CTRLA.RUNSTDBY=1: The Receive Complete interrupt can wake up the device.
- Slave operation, CTRLA.RUNSTDBY=0: All reception will be dropped, including the ongoing transaction.

### 29.6.6 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 the CTRLA register (CTRLA.SWRST)
- Enable bit in the CTRLA register (CTRLA.ENABLE)
- Receiver Enable bit in the CTRLB register (CTRLB.RXEN)

**Note:** CTRLB.RXEN is write-synchronized somewhat differently. See also *CTRLB* register for details.

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

#### Related Links

[Register Synchronization](#)

### Bit 0 – SWRST Software Reset

Writing '0' to this bit has no effect.

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

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

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

This bit is not enable-protected.

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



**Note:** The I<sup>2</sup>C standard *Fm+* (Fast-mode plus) requires a nominal high to low SCL ratio of 1:2, and BAUD should be set accordingly. At a minimum, BAUD.BAUD and/or BAUD.BAUDLOW must be non-zero.

**Startup Timing** The minimum time between SDA transition and SCL rising edge is 6 APB cycles when the DATA register is written in smart mode. If a greater startup time is required due to long rise times, the time between DATA write and IF clear must be controlled by software.

**Note:** When timing is controlled by user, the Smart Mode cannot be enabled.

#### Related Links

[Electrical Characteristics](#)

#### Master Clock Generation (High-Speed Mode)

For I<sup>2</sup>C *Hs* transfers, there is no SCL synchronization. Instead, the SCL frequency is determined by the GCLK\_SERCOMx\_CORE frequency ( $f_{GCLK}$ ) and the High-Speed Baud setting in the Baud register (BAUD.HSBAUD). When BAUD.HSBAUDLOW=0, the HSBAUD value will determine both SCL high and SCL low. In this case the following formula determines the SCL frequency.

$$f_{SCL} = \frac{f_{GCLK}}{2 + 2 \cdot HS\ BAUD}$$

When HSBAUDLOW is non-zero, the following formula determines the SCL frequency.

$$f_{SCL} = \frac{f_{GCLK}}{2 + HS\ BAUD + HSBAUDLOW}$$

**Note:** The I<sup>2</sup>C standard *Hs* (High-speed) requires a nominal high to low SCL ratio of 1:2, and HSBAUD should be set accordingly. At a minimum, BAUD.HSBAUD and/or BAUD.HSBAUDLOW must be non-zero.

#### Transmitting Address Packets

The I<sup>2</sup>C master starts a bus transaction by writing the I<sup>2</sup>C slave address to ADDR.ADDR and the direction bit, as described in [Principle of Operation](#). If the bus is busy, the I<sup>2</sup>C master will wait until the bus becomes idle before continuing the operation. When the bus is idle, the I<sup>2</sup>C master will issue a start condition on the bus. The I<sup>2</sup>C master will then transmit an address packet using the address written to ADDR.ADDR. After the address packet has been transmitted by the I<sup>2</sup>C master, one of four cases will arise according to arbitration and transfer direction.

#### Case 1: Arbitration lost or bus error during address packet transmission

If arbitration was lost during transmission of the address packet, the Master on Bus bit in the Interrupt Flag Status and Clear register (INTFLAG.MB) and the Arbitration Lost bit in the Status register (STATUS.ARBLOST) are both set. Serial data output to SDA is disabled, and the SCL is released, which disables clock stretching. In effect the I<sup>2</sup>C master is no longer allowed to execute any operation on the bus until the bus is idle again. A bus error will behave similarly to the arbitration lost condition. In this case, the MB interrupt flag and Master Bus Error bit in the Status register (STATUS.BUSERR) are both set in addition to STATUS.ARBLOST.

The Master Received Not Acknowledge bit in the Status register (STATUS.RXNACK) will always contain the last successfully received acknowledge or not acknowledge indication.

In this case, software will typically inform the application code of the condition and then clear the interrupt flag before exiting the interrupt routine. No other flags have to be cleared at this moment, because all flags will be cleared automatically the next time the ADDR.ADDR register is written.

#### Case 2: Address packet transmit complete – No ACK received

Offset	Name	Bit Pos.									
0x63											
0x64	PATTB	7:0	PGEB0[7:0]								
		15:8	PGVB0[7:0]								
0x66 ... 0x67	Reserved										
0x68	WAVEB	7:0	CIPERENB		RAMPB[1:0]			WAVEGENB[2:0]			
		15:8					CICCENB3	CICCENB2	CICCENB1	CICCENB0	
		23:16					POLB3	POLB2	POLB1	POLB0	
		31:24					SWAPB 3	SWAPB 2	SWAPB 1	SWAPB 0	
0x6C	PERB	7:0	PERB[1:0]		DITHERB[5:0]						
		15:8	PERB[9:2]								
		23:16	PERB[17:10]								
		31:24									
0x70	CCB0	7:0	CCB[1:0]		DITHERB[5:0]						
		15:8	CCB[9:2]								
		23:16	CCB[17:10]								
		31:24									
0x74	CCB1	7:0	CCB[1:0]		DITHERB[5:0]						
		15:8	CCB[9:2]								
		23:16	CCB[17:10]								
		31:24									
0x78	CCB2	7:0	CCB[1:0]		DITHERB[5:0]						
		15:8	CCB[9:2]								
		23:16	CCB[17:10]								
		31:24									
0x7C	CCB3	7:0	CCB[1:0]		DITHERB[5:0]						
		15:8	CCB[9:2]								
		23:16	CCB[17:10]								
		31:24									

### 32.8 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.

### 35.6.2.3 Enabling the Output Buffer

To enable the DAC output on the  $V_{OUT}$  pin, the output driver must be enabled by writing a one to the External Output Enable bit in the Control B register (CTRLB.EOEN).

The DAC output buffer provides a high-drive-strength output, and is capable of driving both resistive and capacitive loads. To minimize power consumption, the output buffer should be enabled only when external output is needed.

### 35.6.2.4 Digital to Analog Conversion

The DAC converts a digital value (stored in the DATA register) into an analog voltage. The conversion range is between GND and the selected DAC voltage reference. The default voltage reference is the internal reference voltage. Other voltage reference options are the analog supply voltage (VDDANA) and the external voltage reference (VREFA). The voltage reference is selected by writing to the Reference Selection bits in the Control B register (CTRLB.REFSEL).

The output voltage from the DAC can be calculated using the following formula:

$$V_{OUT} = \frac{DATA}{0x3FF} \cdot VREF$$

A new conversion starts as soon as a new value is loaded into DATA. DATA can either be loaded via the APB bus during a CPU write operation, using DMA, or from the DATABUF register when a START event occurs. Refer to [Events](#) for details. As there is no automatic indication that a conversion is done, the sampling period must be greater than or equal to the specified conversion time.

### 35.6.3 DMA Operation

The DAC generates the following DMA request:

- Data Buffer Empty (EMPTY): The request is set when data is transferred from DATABUF to the internal data buffer of DAC. The request is cleared when DATABUF register is written, or by writing a one to the EMPTY bit in the Interrupt Flag register (INTFLAG.EMPTY).

For each Start Conversion event, DATABUF is transferred into DATA and the conversion starts. When DATABUF is empty, the DAC generates the DMA request for new data. As DATABUF is initially empty, a DMA request is generated whenever the DAC is enabled.

If the CPU accesses the registers that are the source of a DMA request set/clear condition, the DMA request can be lost or the DMA transfer can be corrupted, if enabled.

When DAC registers are write-protected by Peripheral Access Controller, DATABUF cannot be written. To bypass DATABUF write protection, Bypass DATABUF Write Protection bit (CTRLB.BDWP) must be written to '1'

### 35.6.4 Interrupts

The DAC Controller has the following interrupt sources:

- Data Buffer Empty (EMPTY): Indicates that the internal data buffer of the DAC is empty.
- Underrun (UNDERRUN): Indicates that the internal data buffer of the DAC is empty and a DAC start of conversion event occurred. Refer to [Events](#) for details.
- Synchronization Ready (SYNCRDY): this asynchronous interrupt can be used to wake-up the device from any sleep mode.

Each interrupt source has an interrupt flag associated with it. The interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG) 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 register

Value	Description
0	The DAC output is turned off.
1	The high-drive output buffer drives the DAC output to the $V_{OUT}$ pin.

**37.11.6 Analog Comparator Characteristics**
**Table 37-34. Electrical and Timing**

Parameter	Conditions	Symbol	Min.	Typ.	Max.	Unit
Positive input voltage range			0	-	$V_{DDANA}$	V
Negative input voltage range			0	-	$V_{DDANA}$	
Offset	Hysteresis = 0, Fast mode		-26	0	26	mV
	Hysteresis = 0, Low power mode		-28	0	28	mV
Hysteresis	Hysteresis = 1, Fast mode		8	50	102	mV
	Hysteresis = 1, Low power mode		14	50	75	mV
Propagation delay	Changes for $V_{ACM} = V_{DDANA}/2$ 100mV overdrive, Fast mode			90	180	ns
	Changes for $V_{ACM} = V_{DDANA}/2$ 100mV overdrive, Low power mode			302	534	ns
Startup time	Enable to ready delay Fast mode	$t_{STARTUP}$		1	2	$\mu s$
	Enable to ready delay Low power mode		-	14	23	$\mu s$
INL <sup>(3)</sup>		$V_{SCALE}$	-1.4	0.201	1.4	LSB
DNL <sup>(3)</sup>			-0.9	0.022	0.9	LSB
Offset Error <sup>(1)(2)</sup>			-0.2	0.056	0.92	LSB
Gain Error <sup>(1)(2)</sup>			-0.89	0.079	0.89	LSB

1. According to the standard equation  $V(X) = V_{LSB} \times (X + 1)$ ;  $V_{LSB} = V_{DDANA}/64$
2. Data computed with the Best Fit method
3. Data computed using histogram

**37.11.7 Internal 1.1V Bandgap Reference Characteristics**
**Table 37-35. Bandgap and Internal 1.1V Reference Characteristics**

Parameter	Conditions	Symbol	Min.	Typ.	Max.	Unit
Internal 1.1V Bandgap reference	After calibration at $T = 25^{\circ}C$ , over $[-40, +105]^{\circ}C$ , $V_{DD} = 3.3V$	INT1V	1.07	1.1	1.12	V
	Over voltage at $25^{\circ}C$		1.08	1.1	1.11	V