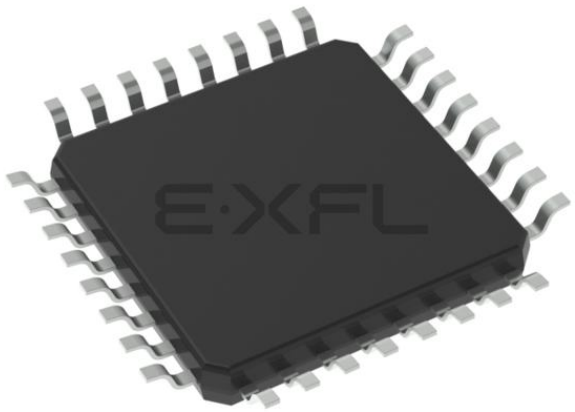


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

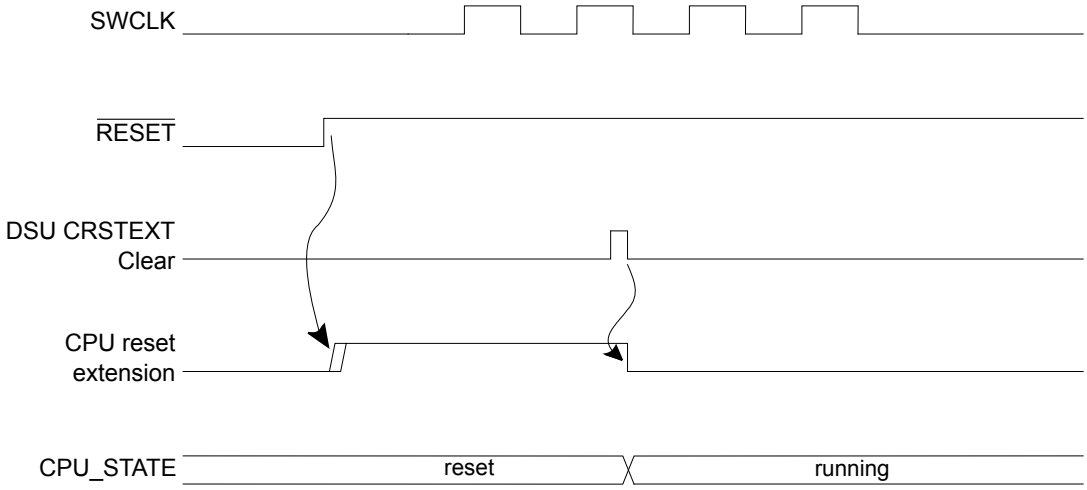
"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" 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, SPI, UART/USART, USB
Peripherals	Brown-out Detect/Reset, DMA, I²S, POR, PWM, WDT
Number of I/O	26
Program Memory Size	256KB (256K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	32K x 8
Voltage - Supply (Vcc/Vdd)	1.62V ~ 3.6V
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-TQFP
Supplier Device Package	32-TQFP (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atsamd21e18a-af

the reset extension phase, the CPU Reset Extension bit of the Status A register (STATUSA.CRSTEXT) is set. To release the CPU, write a '1' to STATUSA.CRSTEXT. STATUSA.CRSTEXT will then be set to zero. Writing a '0' to STATUSA.CRSTEXT has no effect. For security reasons, it is not possible to release the CPU reset extension when the device is protected by the NVMCTRL security bit. Trying to do so sets the Protection Error bit (PERR) of the Status A register (STATUSA.PERR).

Figure 13-2. Typical CPU Reset Extension Set and Clear Timing Diagram



Related Links

NVMCTRL – Non-Volatile Memory Controller

Security Bit

13.6.3 Debugger Probe Detection

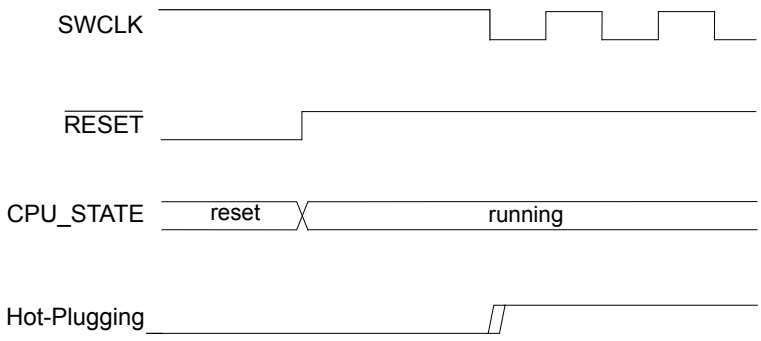
13.6.3.1 Cold Plugging

Cold-Plugging is the detection of a debugger when the system is in reset. Cold-Plugging is detected when the CPU reset extension is requested, as described above.

13.6.3.2 Hot Plugging

Hot-Plugging is the detection of a debugger probe when the system is not in reset. Hot-Plugging is not possible under reset because the detector is reset when POR or $\overline{\text{RESET}}$ are asserted. Hot-Plugging is active when a SWCLK falling edge is detected. The SWCLK pad is multiplexed with other functions and the user must ensure that its default function is assigned to the debug system. If the SWCLK function is changed, the Hot-Plugging feature is disabled until a power-reset or external reset occurs. Availability of the Hot-Plugging feature can be read from the Hot-Plugging Enable bit of the Status B register (STATUSB.HPE).

Figure 13-3. Hot-Plugging Detection Timing Diagram



Bits 7,6,5,4,3,2,1,0 – PEREOx : Periodic Interval x Event Output Enable [x=7:0]

Value	Description
0	Periodic Interval x event is disabled and will not be generated.
1	Periodic Interval x event is enabled and will be generated.

19.8.7 Event Control - MODE2

Name: EVCTRL
Offset: 0x04
Reset: 0x0000
Property: Enable-Protected, Write-Protected

Bit	15	14	13	12	11	10	9	8
	OVFEO							ALARMEO0
Access	R/W							R/W
Reset	0							0

Bit	7	6	5	4	3	2	1	0
	PEREO7	PEREO6	PEREO5	PEREO4	PEREO3	PEREO2	PEREO1	PEREO0
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 – OVFEO: Overflow Event Output Enable

Value	Description
0	Overflow event is disabled and will not be generated.
1	Overflow event is enabled and will be generated for every overflow.

Bit 8 – ALARMEO0: Alarm 0 Event Output Enable

Value	Description
0	Alarm 0 event is disabled and will not be generated.
1	Alarm 0 event is enabled and will be generated for every alarm.

Bits 7,6,5,4,3,2,1,0 – PEREOx : Periodic Interval x Event Output Enable [x=7:0]

Value	Description
0	Periodic Interval x event is disabled and will not be generated.
1	Periodic Interval x event is enabled and will be generated.

19.8.8 Interrupt Enable Clear - MODE0

Name: INTENCLR
Offset: 0x06
Reset: 0x00
Property: Write-Protected

Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
					CHINT11	CHINT10	CHINT9	CHINT8
Access					R	R	R	R
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CHINT7	CHINT6	CHINT5	CHINT4	CHINT3	CHINT2	CHINT1	CHINT0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 11:0 – CHINTn: Channel n Pending Interrupt [n=11..0]
This bit is set when Channel n has a pending interrupt/the interrupt request is received.

This bit is cleared when the corresponding Channel n interrupts are disabled or the interrupts sources are cleared.

20.8.12 Busy Channels

Name: BUSYCH
Offset: 0x28
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

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

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

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

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

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

Bit 0 – READY: NVM Ready Interrupt Enable
Writing a '0' to this bit has no effect.

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

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

22.8.6 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x14
Reset: 0x00
Property: –

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

Bit 1 – ERROR: Error
This flag is set on the occurrence of an NVME, LOCKE or PROGE error.

This bit can be cleared by writing a '1' to its bit location.

Value	Description
0	No errors have been received since the last clear.
1	At least one error has occurred since the last clear.

Bit 0 – READY: NVM Ready

Value	Description
0	The NVM controller is busy programming or erasing.
1	The NVM controller is ready to accept a new command.

22.8.7 Status

PMUXO[3:0]	Name	Description
0x8	I	Peripheral function I selected
0x9-0xF	-	Reserved

Bits 3:0 – PMUXE[3:0]: Peripheral Multiplexing for Even-Numbered Pin
These bits select the peripheral function for even-numbered pins (2*n) of a PORT group, if the corresponding PINCFGy.PMUXEN bit is '1'.

Not all possible values for this selection may be valid. For more details, refer to the *I/O Multiplexing and Considerations.*

PMUXE[3:0]	Name	Description
0x0	A	Peripheral function A selected
0x1	B	Peripheral function B selected
0x2	C	Peripheral function C selected
0x3	D	Peripheral function D selected
0x4	E	Peripheral function E selected
0x5	F	Peripheral function F selected
0x6	G	Peripheral function G selected
0x7	H	Peripheral function H selected
0x8	I	Peripheral function I selected
0x9-0xF	-	Reserved

23.8.13 Pin Configuration

There are up to 32 Pin Configuration registers in each PORT group, one for each I/O line.

Name: PINCFGn
Offset: 0x40 + n*0x01 [n=0..31]
Reset: 0x00
Property: PAC Write-Protection

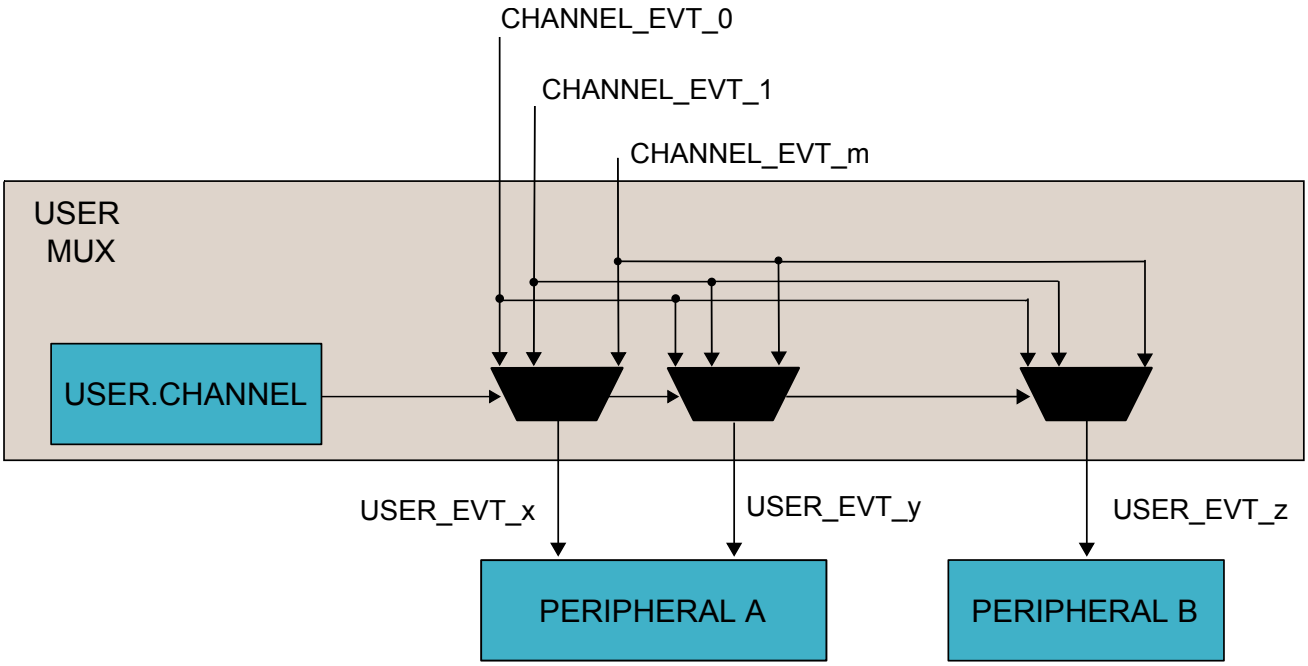
Bit	7	6	5	4	3	2	1	0
		DRVSTR				PULLEN	INEN	PMUXEN
Access		R/W				R/W	R/W	R/W
Reset		0				0	0	0

Bit 6 – DRVSTR: Output Driver Strength Selection
This bit controls the output driver strength of an I/O pin configured as an output.

Value	Description
0	Pin drive strength is set to normal drive strength.
1	Pin drive strength is set to stronger drive strength.

To configure a user multiplexer, the USER register must be written in a single 16-bit write. It is possible to read out the configuration of a user by first selecting the user by writing to USER.USER using an 8-bit write and then performing a read of the 16-bit USER register.

Figure 24-2. User MUX



24.6.2.4 Channel Setup

An event channel can select one event from a list of event generators. Depending on configuration, the selected event could be synchronized, resynchronized or asynchronously sent to the users. When synchronization or resynchronization is required, the channel includes an internal edge detector, allowing the Event System to generate internal events when rising, falling or both edges are detected on the selected event generator. An event channel is able to generate internal events for the specific software commands. All these configurations are available in the Channel register (CHANNEL).

To configure a channel, the Channel register must be written in a single 32-bit write. It is possible to read out the configuration of a channel by first selecting the channel by writing to CHANNEL.CHANNEL using a, 8-bit write, and then performing a read of the CHANNEL register.

24.6.2.5 Channel Path

There are three different ways to propagate the event provided by an event generator:

- Asynchronous path
- Synchronous path
- Resynchronized path

Value	Event Generator	Description
0x40	TC4 MC0	Match/Capture 0
0x41	TC4 MC1	Match/Capture 1
0x42	ADC RESRDY	Result Ready
0x43	ADC WINMON	Window Monitor
0x44	AC COMP0	Comparator 0
0x45	AC COMP1	Comparator 1
0x46	AC WIN0	Window 0
0x47	DAC EMPTY	Data Buffer Empty
0x48	PTC EOC	End of Conversion
0x49	PTC WCOMP	Window Comparator
0x4A-0x7F	Reserved	

Bit 8 – SWEVT: Software Event
This bit is used to insert a software event on the channel selected by the CHANNEL.CHANNEL bit group.

This bit has the same behavior similar to an event.

This bit must be written together with CHANNEL.CHANNELusing a 16-bit write.

Writing a zero to this bit has no effect.

Writing a one to this bit will trigger a software event for the corresponding channel.

This bit will always return zero when read.

Bits 3:0 – CHANNEL[3:0]: Channel Selection
These bits are used to select the channel to be set up or read from.

24.8.3 User Multiplexer

Name: USER
Offset: 0x08
Reset: 0x0000
Property: Write-Protected

Bit	15	14	13	12	11	10	9	8
				CHANNEL[4:0]				
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

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

Case 2: Data sent

When a byte transmission is successfully completed, the INTFLAG.DRDY interrupt flag is set. If NACK is received, indicated by STATUS.RXNACK=1, the I^2C slave must expect a stop or a repeated start to be received. The I^2C slave must release the data line to allow the I^2C master to generate a stop or repeated start. Upon detecting a stop condition, the Stop Received bit in the Interrupt Flag register (INTFLAG.PREC) will be set and the I^2C slave will return to IDLE state.

High-Speed Mode

When the I^2C slave is configured in High-speed mode (*Hs*, CTRLA.SPEED=0x2) and CTRLA.SCLSM=1, switching between Full-speed and High-speed modes is automatic. When the slave recognizes a START followed by a master code transmission and a NACK, it automatically switches to High-speed mode and sets the High-speed status bit (STATUS.HS). The slave will then remain in High-speed mode until a STOP is received.

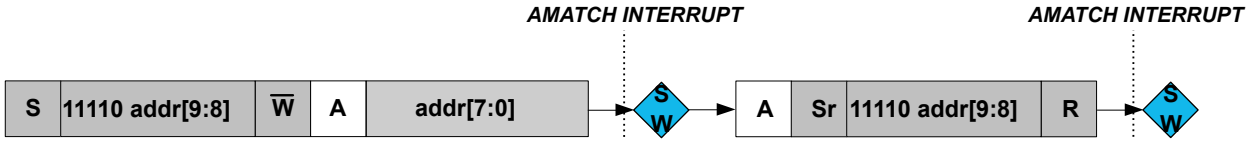
10-Bit Addressing

When 10-bit addressing is enabled (ADDR.TENBITEN=1), the two address bytes following a START will be checked against the 10-bit slave address recognition. The first byte of the address will always be acknowledged, and the second byte will raise the address interrupt flag, see 10-bit Addressing.

If the transaction is a write, then the 10-bit address will be followed by *N* data bytes.

If the operation is a read, the 10-bit address will be followed by a repeated START and reception of '11110 `ADDR[9:8]` 1', and the second address interrupt will be received with the DIR bit set. The slave matches on the second address as it it was addressed by the previous 10-bit address.

Figure 28-11. 10-bit Addressing



PMBus Group Command

When the PMBus Group Command bit in the CTRLB register is set (CTRLB.GCMD=1) and 7-bit addressing is used, INTFLAG.PREC will be set when a STOP condition is detected on the bus. When CTRLB.GCMD=0, a STOP condition without address match will not be set INTFLAG.PREC.

The group command protocol is used to send commands to more than one device. The commands are sent in one continuous transmission with a single STOP condition at the end. When the STOP condition is detected by the slaves addressed during the group command, they all begin executing the command they received.

PMBus Group Command Example shows an example where this slave, bearing ADDRESS 1, is addressed after a repeated START condition. There can be multiple slaves addressed before and after this slave. Eventually, at the end of the group command, a single STOP is generated by the master. At this point a STOP interrupt is asserted.

Bit 7 – ERROR: Error Interrupt Enable
Writing '0' to this bit has no effect.

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

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

Bit 2 – DRDY: Data Ready Interrupt Enable
Writing '0' to this bit has no effect.

Writing '1' to this bit will set the Data Ready bit, which enables the Data Ready interrupt.

Value	Description
0	The Data Ready interrupt is disabled.
1	The Data Ready interrupt is enabled.

Bit 1 – AMATCH: Address Match Interrupt Enable
Writing '0' to this bit has no effect.

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

Value	Description
0	The Address Match interrupt is disabled.
1	The Address Match interrupt is enabled.

Bit 0 – PREC: Stop Received Interrupt Enable
Writing '0' to this bit has no effect.

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

Value	Description
0	The Stop Received interrupt is disabled.
1	The Stop Received interrupt is enabled.

28.8.5 Interrupt Flag Status and Clear

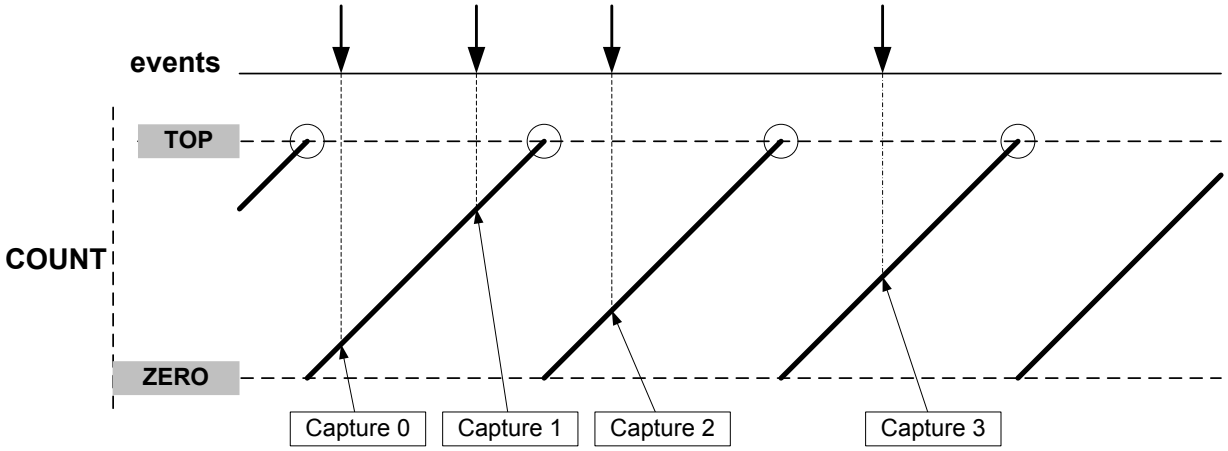
Name: INTFLAG
Offset: 0x18
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
	ERROR					DRDY	AMATCH	PREC
Access	R/W					R/W	R/W	R/W
Reset	0					0	0	0

29.8 Register Summary

Offset	Name	Bit Pos.								
0x00	CTRLA	7:0			SEREN1	SEREN0	CKEN1	CKEN0	ENABLE	SWRST
0x01 ... 0x03	Reserved									
0x04	CLKCTRLn0	7:0	BITDELAY	FSWIDTH[1:0]		NBSLOTS[2:0]			SLOTSIZE[1:0]	
0x05		15:8				SCKSEL	FSINV			FSSEL
0x06		23:16	MCKDIV[4:0]					MCKEN		MCKSEL
0x07		31:24	MCKOUTINV	SCKOUTINV	FSOUTINV	MCKOUTDIV[4:0]				
0x08	CLKCTRLn1	7:0	BITDELAY	FSWIDTH[1:0]		NBSLOTS[2:0]			SLOTSIZE[1:0]	
0x09		15:8				SCKSEL	FSINV			FSSEL
0x0A		23:16	MCKDIV[4:0]					MCKEN		MCKSEL
0x0B		31:24	MCKOUTINV	SCKOUTINV	FSOUTINV	MCKOUTDIV[4:0]				
0x0C	INTENCLR	7:0			RXOR1	RXOR0			RXRDY1	RXRDY0
0x0D		15:8			TXUR1	TXUR0			TXRDY1	TXRDY0
0x0E ... 0x0F	Reserved									
0x10	INTENSET	7:0			RXOR1	RXOR0			RXRDY1	RXRDY0
0x11		15:8			TXUR1	TXUR0			TXRDY1	TXRDY0
0x12 ... 0x13	Reserved									
0x14	INTFLAG	7:0			RXOR1	RXOR0			RXRDY1	RXRDY0
0x15		15:8			TXUR1	TXUR0			TXRDY1	TXRDY0
0x16 ... 0x17	Reserved									
0x18	SYNCBUSY	7:0			SEREN1	SEREN0	CKEN1	CKEN0	ENABLE	SWRST
0x19		15:8							DATA1	DATA0
0x1A ... 0x1F	Reserved									
0x20	SERCTRLn0	7:0	SLOTADJ		CLKSEL	TXSAME	TXDEFAULT[1:0]		SERMODE[1:0]	
0x21		15:8	BITREV	EXTEND[1:0]		WORDADJ		DATASIZE[2:0]		
0x22		23:16	SLOTDIS8	SLOTDIS7	SLOTDIS6	SLOTDIS5	SLOTDIS4	SLOTDIS3	SLOTDIS1	SLOTDIS0
0x23		31:24						RXLOOP	DMA	MONO
0x24	SERCTRLn1	7:0	SLOTADJ		CLKSEL	TXSAME	TXDEFAULT[1:0]		SERMODE[1:0]	
0x25		15:8	BITREV	EXTEND[1:0]		WORDADJ		DATASIZE[2:0]		
0x26		23:16	SLOTDIS8	SLOTDIS7	SLOTDIS6	SLOTDIS5	SLOTDIS4	SLOTDIS3	SLOTDIS1	SLOTDIS0
0x27		31:24						RXLOOP	DMA	MONO
0x28 ... 0x2F	Reserved									

Figure 30-9. Input Capture Timing



The TC can detect capture overflow of the input capture channels: When a new capture event is detected while the Capture Interrupt flag (INTFLAG.MCx) is still set, the new timestamp will not be stored and INTFLAG.ERR will be set.

Period and Pulse-Width (PPW) Capture Action

The TC can perform two input captures and restart the counter on one of the edges. This enables the TC to measure the pulse width and period and to characterize the frequency *f* and duty cycle of an input signal:

$$f = \frac{1}{T} \qquad \text{dutyCycle} = \frac{t_p}{T}$$

Selecting PWP (pulse-width, period) in the Event Action bit group in the Event Control register (EVCTRL.EVACT) enables the TC to perform one capture action on the rising edge and the other one on the falling edge. The period *T* will be captured into CC1 and the pulse width t_p in CC0. EVCTRL.EVACT=PPW (period and pulse-width)offers identical functionality, but will capture *T* into CC0 and t_p into CC1.

The TC Event Input Invert Enable bit in the Event Control register (EVCTRL.TCINV) is used to select whether the wraparound should occur on the rising edge or the falling edge. If EVCTRL.TCINV=1, the wraparound will happen on the falling edge.

To fully characterize the frequency and duty cycle of the input signal, activate capture on CC0 and CC1 by writing 0x3 to the Capture Channel x Enable bit group in the Control C register (CTRLC.CPTEN). When only one of these measurements is required, the second channel can be used for other purposes.

The TC can detect capture overflow of the input capture channels: When a new capture event is detected while the Capture Interrupt flag (INTFLAG.MCx) is still set, the new timestamp will not be stored and INTFLAG.ERR will be set.

- Enable bit in the Control A register (CTRLA.ENABLE)

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

The following registers are synchronized when written:

- Control B Clear register (CTRLBCLR)
- Control B Set register (CTRLBSET)
- Control C register (CTRLC)
- Count Value register (COUNT)
- Period Value register (PERIOD)
- Compare/Capture Value registers (CCx)

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

The following registers are synchronized when read:

- Control B Clear register (CTRLBCLR)
- Control B Set register (CTRLBSET)
- Control C register (CTRLC)
- Count Value register (COUNT)
- Period Value register (PERIOD)
- Compare/Capture Value registers (CCx)

Required read-synchronization is denoted by the "Read-Synchronized" property in the register description.

Related Links

Register Synchronization

30.7 Register Summary

Table 30-4. Register Summary – 8-bit Mode

Offset	Name	Bit Pos.								
0x00	CTRLA	7:0		WAVEGEN[1:0]			MODE[1:0]		ENABLE	SWRST
0x01		15:8			PRESCSYNC[1:0]		RUNSTDBY	PRESCALER[2:0]		
0x02	READREQ	7:0				ADDR[4:0]				
0x03		15:8	RREQ	RCONT						
0x04	CTRLBCLR	7:0	CMD[1:0]					ONESHOT		DIR
0x05	CTRLBSET	7:0	CMD[1:0]					ONESHOT		DIR
0x06	CTRLC	7:0			CPTEN1	CPTEN0			INVEN1	INVEN0
0x07	Reserved									
0x08	DBGCTRL	7:0								DBGRUN
0x09	Reserved									
0x0A	EVCTRL	7:0			TCEI	TCINV		EVACT[2:0]		
0x0B		15:8			MCEO1	MCEO0				OVFEO
0x0C	INTENCLR	7:0			MC1	MC0	SYNCRDY		ERR	OVF
0x0D	INTENSET	7:0			MC1	MC0	SYNCRDY		ERR	OVF
0x0E	INTFLAG	7:0			MC1	MC0	SYNCRDY		ERR	OVF

31.6.4.2 Interrupts

The TCC has the following interrupt sources:

- Overflow/Underflow (OVF)
- Retrigger (TRG)
- Count (CNT) - refer also to description of EVCTRL.CNTSEL.
- Capture Overflow Error (ERR)
- Debug Fault State (DFS)
- Recoverable Faults (FAULTn)
- Non-recoverable Faults (FAULTx)
- Compare Match or Capture Channels (MCx)

These interrupts are asynchronous wake-up sources. See Sleep Mode Entry and Exit Table in PM/Sleep Mode Controller section for details.

Each interrupt source has an interrupt flag associated with it. The interrupt flag in the Interrupt Flag Status and Clear (INTFLAG) register is set when the interrupt condition occurs. Each interrupt can be individually enabled by writing a '1' to the corresponding bit in the Interrupt Enable Set (INTENSET) register, and disabled by writing a '1' to the corresponding bit in the Interrupt Enable Clear (INTENCLR) register. An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until the interrupt flag is cleared, the interrupt is disabled, or the TCC is reset. See INTFLAG for details on how to clear interrupt flags. The TCC has one common interrupt request line for all the interrupt sources. The user must read the INTFLAG register to determine which interrupt condition is present.

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

Related Links

Nested Vector Interrupt Controller

Sleep Mode Controller

IDLE Mode

STANDBY Mode

31.6.4.3 Events

The TCC can generate the following output events:

- Overflow/Underflow (OVF)
- Trigger (TRG)
- Counter (CNT) For further details, refer to EVCTRL.CNTSEL description.
- Compare Match or Capture on compare/capture channels: MCx

Writing a '1' ('0') to an Event Output bit in the Event Control Register (EVCTRL.xxEO) enables (disables) the corresponding output event. Refer also to *EVSYS – Event System*.

The TCC can take the following actions on a channel input event (MCx):

- Capture event
- Generate a recoverable or non-recoverable fault

The TCC can take the following actions on counter Event 1 (TCCx EV1):

- Counter re-trigger
- Counter direction control
- Stop the counter

Value	Name	Description
0x3	CAPTMAX	On rising edge of a valid recoverable Fault n, capture counter value on channel selected by CHSEL[1:0], if COUNT value is higher than the last stored capture value (CC). INTFLAG.FAULTn flag rises on each local maximun detection.
0x4	LOCMIN	On rising edge of a valid recoverable Fault n, capture counter value on channel selected by CHSEL[1:0]. INTFLAG.FAULTn flag rises on each local minimum value detection.
0x5	LOCMAX	On rising edge of a valid recoverable Fault n, capture counter value on channel selected by CHSEL[1:0]. INTFLAG.FAULTn flag rises on each local maximun detection.
0x6	DERIV0	On rising edge of a valid recoverable Fault n, capture counter value on channel selected by CHSEL[1:0]. INTFLAG.FAULTn flag rises on each local maximun or minimum detection.

Bits 11:10 – CHSEL[1:0]: Recoverable Fault n Capture Channel
These bits select the channel for capture operation triggered by recoverable Fault n.

Value	Name	Description
0x0	CC0	Capture value stored into CC0
0x1	CC1	Capture value stored into CC1
0x2	CC2	Capture value stored into CC2
0x3	CC3	Capture value stored into CC3

Bits 9:8 – HALT[1:0]: Recoverable Fault n Halt Operation
These bits select the halt action for recoverable Fault n.

Value	Name	Description
0x0	DISABLE	Halt action disabled
0x1	HW	Hardware halt action
0x2	SW	Software halt action
0x3	NR	Non-recoverable fault

Bit 7 – RESTART: Recoverable Fault n Restart
Setting this bit enables restart action for Fault n.

Value	Description
0	Fault n restart action is disabled.
1	Fault n restart action is enabled.

Bits 6:5 – BLANK[1:0]: Recoverable Fault n Blanking Operation
These bits, select the blanking start point for recoverable Fault n.

An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until the interrupt flag is cleared, the interrupt is disabled or the DAC is reset. See INTFLAG register for details on how to clear interrupt flags.

All interrupt requests from the peripheral are ORed together on system level to generate one combined interrupt request to the NVIC. The user must read the INTFLAG register to determine which interrupt condition is present.

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

Related Links

Nested Vector Interrupt Controller

35.6.5 Events

The DAC Controller can generate the following output events:

- Data Buffer Empty (EMPTY): Generated when the internal data buffer of the DAC is empty. Refer to DMA Operation for details.

Writing a '1' to an Event Output bit in the Event Control register (EVCTRL.EMPTYEO) enables the corresponding output event. Writing a '0' to this bit disables the corresponding output event.

The DAC can take the following action on an input event:

- Start Conversion (START): DATABUF value is transferred into DATA as soon as the DAC is ready for the next conversion, and then conversion is started. START is considered as asynchronous to GCLK_DAC thus it is resynchronized in DAC Controller. Refer to Digital to Analog Conversion for details.

Writing a '1' to an Event Input bit in the Event Control register (EVCTRL.STARTEI) enables the corresponding action on an input event. Writing a '0' to this bit disables the corresponding action on input event.

Note: When several events are connected to the DAC Controller, the enabled action will be taken on any of the incoming events.

By default, DAC Controller detects rising edge events. Falling edge detection can be enabled by writing a '1' to EVCTRL.INVEIx.

Related Links

EVSYS – Event System

35.6.6 Sleep Mode Operation

The generic clock for the DAC is running in idle sleep mode. If the Run In Standby bit in the Control A register (CTRLA.RUNSTDBY) is one, the DAC output buffer will keep its value in standby sleep mode. If CTRLA.RUNSTDBY is zero, the DAC output buffer will be disabled in standby sleep mode.

35.6.7 Synchronization

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

- Synchronization when written
- Synchronization when read
- Synchronization when written and read
- No synchronization

  - DFLL48M using XOSC32K as reference and running at 48 MHz
- Clocks
  - DFLL48M used as main clock source, except otherwise specified
  - CPU, AHB clocks undivided
  - APBA clock divided by 4
  - APBB and APBC bridges off
- The following AHB module clocks are running: NVMCTRL, APBA bridge
  - All other AHB clocks stopped
- The following peripheral clocks running: PM, SYSCTRL, RTC
  - All other peripheral clocks stopped
- I/Os are inactive with internal pull-up
- CPU is running on flash with 1 wait states
- Cache enabled
- BOD33 disabled

Mode	Conditions	T_A	V_{CC}	Typ.	Max.	Units
IDLE0	Default operating conditions	25°C	3.3V	2.4	2.5	mA
		85°C	3.3V	2.5	2.6	
IDLE1	Default operating conditions	25°C	3.3V	1.8	1.9	
		85°C	3.3V	1.9	2.0	
IDLE2	Default operating conditions	25°C	3.3V	1.3	1.4	
		85°C	3.3V	1.4	1.5	
STANDBY	XOSC32K running, RTC running at 1kHz	25°C	3.3V	4	6.2	µA
		85°C	3.3V	54	100	
	XOSC32K and RTC stopped	25°C	3.3V	2.8	5.0	
		85°C	3.3V	52	98.8	

Table 37-9. Current Consumption (Device Variant B and C / Die Revision F)

Mode	Conditions	T_A	V_{CC}	Typ.	Max.	Units
ACTIVE	CPU running a While 1 algorithm	25°C	3.3V	3.7	3.9	mA
		85°C	3.3V	3.8	4	
	CPU running a While 1 algorithm	25°C	1.8V	3.7	3.9	
		85°C	1,8V	3.8	4	
	CPU running a While 1 algorithm, with GCLKIN as reference	25°C	3.3V	72*Freq+107	76*Freq+111	µA (with freq in MHz)
		85°C	3.3V	72*Freq+198	76*Freq+210	
	CPU running a Fibonacci algorithm	25°C	3.3V	4.2	4.6	mA
		85°C	3.3V	4.3	4.7	
	CPU running a Fibonacci algorithm	25°C	1.8V	4.2	4.6	
		85°C	1.8V	4.3	4.7	
	CPU running a Fibonacci algorithm, with GCLKIN as reference	25°C	3.3V	83*Freq+107	87*Freq+111	µA(with freq in MHz)
		85°C	3.3V	84*Freq+199	87*Freq+216	
	CPU running a CoreMark algorithm	25°C	3.3V	5.2	5.7	
		85°C	3.3V	5.3	5.8	
	CPU running a CoreMark algorithm	25°C	1.8V	4.8	5.1	
		85°C	1.8V	4.9	5.2	
	CPU running a CoreMark algorithm, with GCLKIN as reference	25°C	3.3V	104*Freq+109	108*Freq+113	µA (with freq in MHz)
		85°C	3.3V	104*Freq+200	109*Freq+212	

Figure 39-2. External Analog Reference Schematic With Two References

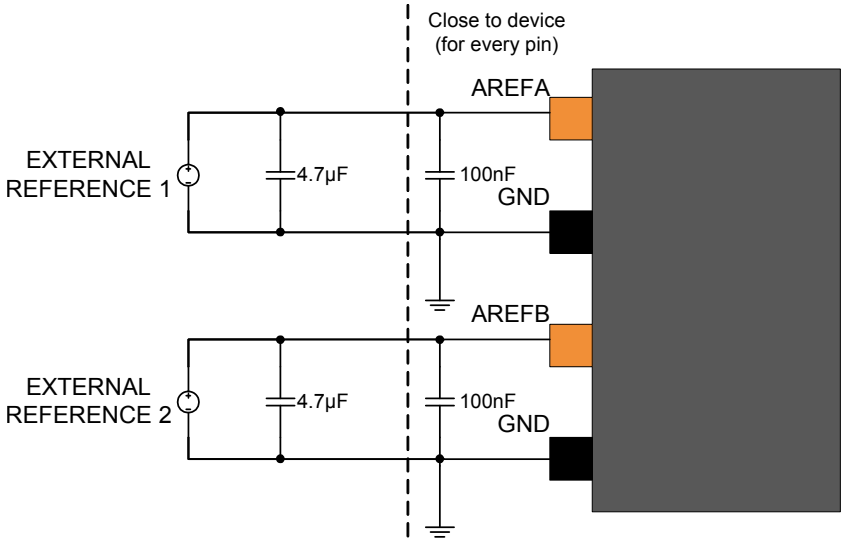


Figure 39-3. External Analog Reference Schematic With One Reference

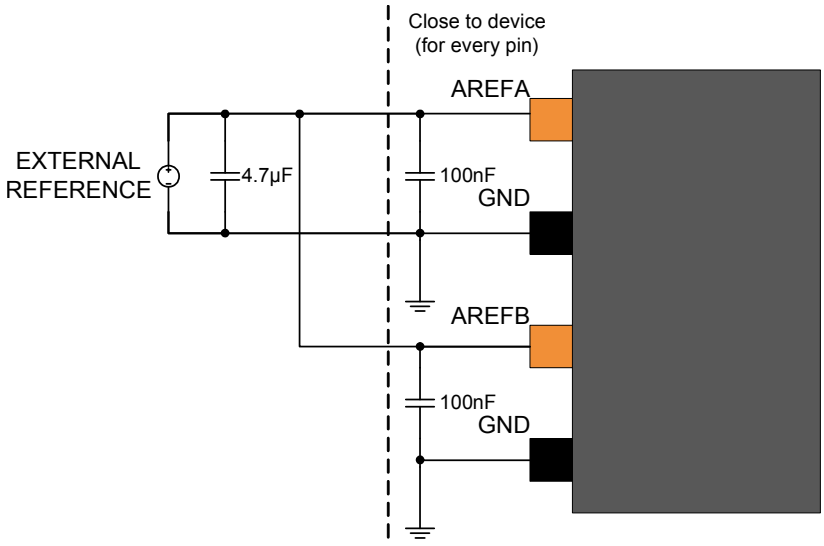


Table 39-2. External Analog Reference Connections

Signal Name	Recommended Pin Connection	Description
AREFx	1.0V to V_{DDANA} - 0.6V for ADC 1.0V to V_{DDANA}- 0.6V for DAC Decoupling/filtering capacitors 100nF(1)(2) and 4.7µF(1)	External reference from AREFx pin on the analog port
GND		Ground

1. These values are given as a typical example.
2. Decoupling capacitor should be placed close to the device for each supply pin pair in the signal group.

Do a power cycle to reset the GCLK generators after an external XOSC32K failure.

40.3.1.2 DSU

1 – The MBIST ""Pause-on-Error"" feature is not functional on this device.
Errata reference: 14324
Fix/Workaround:
Do not use the ""Pause-on-Error"" feature.

40.3.1.3 DFLL48M

1 – The DFLL clock must be requested before being configured otherwise a write access to a DFLL register can freeze the device.
Errata reference: 9905
Fix/Workaround:
Write a zero to the DFLL ONDEMAND bit in the DFLLCTRL register before configuring the DFLL module.

2 – The DFLL status bits in the PCLKSR register during the USB clock recovery mode can be wrong after a USB suspend state.
Errata reference: 11938
Fix/Workaround:
Do not monitor the DFLL status bits in the PCLKSR register during the USB clock recovery mode.

3 – If the DFLL48M reaches the maximum or minimum COARSE or FINE calibration values during the locking sequence, an out of bounds interrupt will be generated. These interrupts will be generated even if the final calibration values at DFLL48M lock are not at maximum or minimum, and might therefore be false out of bounds interrupts.
Errata reference: 10669
Fix/Workaround:
Check that the lockbits: DFLLLCKC and DFLLLCKF in the SYSCTRL Interrupt Flag Status and Clear register (INTFLAG) are both set before enabling the DFLLOOB interrupt.

40.3.1.4 FDPLL

1 – When changing on-the-fly the FDPLL ratio in DPLLnRATIO register, STATUS.DPLLnLDRTO will not be set when the ratio update will be completed.
Errata reference: 15753
Fix/Workaround:
Wait for the interruption flag INTFLAG.DPLLnLDRTO instead.

40.3.1.5 DMAC

1 – When at least one channel using linked descriptors is already active, enabling another DMA channel (with or without linked descriptors) can result in a channel Fetch Error (FERR) or an incorrect descriptor fetch.
This happens if the channel number of the channel being enabled is lower than the channel already active.
Errata reference: 15683
Fix/Workaround:

Updated V_{DD} max from 3.63V to 3.63V in Absolute Maximum Ratings. Updated VDDIN pin from 57 to 56 in GPIO Clusters. Power Consumption: Updated Max values for STANDBY from 190.6µA and 197.3µA to 100µA in Table 37-7. Added Peripheral Power Consumption. I/O Pin Characteristics: tRISE and tFALL updated with different load conditions depending on the DVRSTR value in . I/O Pin Characteristics: Correct typo IOL and IOH Max values inverted between PORT.PINCFG.DRVSTR=0 and 1, tRISE and tFALL updated with different load conditions depending on the DVRSTR value in Table 37-14. Analog Characteristics: Removed note from Table 37-18. Analog-to-Digital (ADC) characteristics: Added Max DC supply current (I_{DD}), R_{SAMPLE} maximum value changed from 2.8kW to 3.5kW, Conversion time Typ value change to Min Value in Table 37-22. Digital to Analog Converter (DAC) Characteristics: Added Max DC supply current (I_{DD}) in Table 37-30. Analog Comparator Characteristics: Added Min and Max values for VSCALE INL, DNL, Offset Error and Gain Error in Table 37-34. Internal 1.1V Bandgap Reference Characteristics: Added Min and Max values, removed accuracy row in Table 37-36. SERCOM in I2C Mode Timing: Add Typical values for t_R in Table 37-62. Removed Asynchronous Watchdog Clock Characterization. 32.768kHz Internal oscillator (OSC32K) Characteristics: Added Max current consumption (I_{OSC32K}) in Table 37-53. Updated Crystal Oscillator Characteristics (XOSC32K) ESR maximum values, Crystal Oscillator Characteristics. Updated Crystal Oscillator Characteristics (XOSC) ESR maximum value, Crystal Oscillator Characteristics from 348kΩ to 141kΩ. Digital Frequency Locked Loop (DFLL48M) Characteristics: Updated presentation, now separating between Open- and Closed Loop Modes. Added f_{REF} Min and Max values to Table 37-50. Updated typical Startup time ($t_{STARTUP}$) from 6.1µs to 8µs in Table 37-51. Updated typical Fine lock time (t_{LFINE}) from 700µs to 600µs in Table 37-51. Fractional Digital Phase Locked Loop (FDPLL96M) Characteristics: Added Current consumption ($I_{FDPLL96M}$), Period Jitter (Jp), Lock time (t_{LOCK}), Duty cycles parameters in Table 37-56. Added USB Characteristics. Timing Characteristics: Added SCK period (t_{SCK}) Typ value in Table 37-60.
Errata
Errata for revision B added.