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	CANbus, I²C, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, DMA, POR, WDT
Number of I/O	38
Program Memory Size	64KB (64K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	8K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 5.5V
Data Converters	A/D 14x12b, 2x16b; D/A 1x10b
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/atsamc21g16a-ant

Peripheral Source	NVIC Line
EVSYS – Event System	8
SERCOM0 – Serial Communication Controller 0 SERCOM6 – Serial Communication Controller 6	9
SERCOM1 – Serial Communication Controller 1 SERCOM7 – Serial Communication Controller 7	10
SERCOM2 – Serial Communication Controller 2	11
SERCOM3 – Serial Communication Controller 3	12
SERCOM4 – Serial Communication Controller 4	13
SERCOM5 – Serial Communication Controller 5	14
CAN0 – Controller Area Network 0	15
CAN1 – Controller Area Network 1	16
TCC0 – Timer Counter for Control 0	17
TCC1 – Timer Counter for Control 1	18
TCC2 – Timer Counter for Control 2	19
TC0 – Timer Counter 0 TC5 – Timer Counter 5	20
TC1 – Timer Counter 1 TC6 – Timer Counter 6	21
TC2 – Timer Counter 2 TC7 – Timer Counter 7	22
TC3 – Timer Counter 3Reserved	23
TC4 – Timer Counter 4Reserved	24
ADC0 – Analog-to-Digital Converter 0	25
ADC1 – Analog-to-Digital Converter 1Reserved	26
AC – Analog Comparator	27
DAC – Digital-to-Analog Converter	28
SDADC – Sigma-Delta Analog-to-Digital Converter 1	29
PTC – Peripheral Touch Controller	30
Reserved	31

IOBUS. When the dividend and divide registers are programmed, the division starts and the result will be stored in the Result and Remainder registers. The Busy and Divide-by-zero status can be read from STATUS register.

When the square root input register (SQRNUM) is programmed, the square root function starts and the result will be stored in the Result and Remainder registers. The Busy status can be read from STATUS register.

14.6.2 Basic Operation

14.6.2.1 Initialization

The DIVAS configuration cannot be modified while a divide operation is ongoing. The following bits must be written prior to starting a division:

- Sign selection bit in Control A register (CTRLA.SIGNED)
- Leading zero mode bit in Control A register (CTRLA.DLZ)

14.6.2.2 Performing Division

First write the dividend to DIVIDEND register. Writing the divisor to DIVISOR register starts the division and sets the busy bit in the Status register (STATUS.BUSY). When the division has completed, the STATUS.BUSY bit is cleared and the result will be stored in RESULT and REMAINDER registers.

The RESULT and REMAINDER registers can be read directly via the high-speed bus without checking first STATUS.BUSY. Wait states will be inserted on the high-speed bus until the operation is complete. The IOBUS does not support wait states. For accesses via the IOBUS, the STATUS.BUSY bit must be polled before reading the result from the RESULTand REMAINDER registers.

14.6.2.3 Operand Size

Divide

The DIVAS can perform 32-bit signed and unsigned division and the operation follows the equation as below.

$$RESULT[31:0] = DIVIDEND[31:0]/DIVISOR[31:0]$$

$$REMAINDER[31:0] = DIVIDEND[31:0]\,\%\,DIVISOR[31:0]$$

DIVAS completes 32-bit division in 2-16 cycles.

Square Root

The DIVAS can perform 32-bit unsigned division and the operation follows the equation as below.

$$RESULT[31:0] = \sqrt{SQRNUM[31:0]}$$

$$REMAINDER\left[31:0\right] = SQRNUM\left[31:0\right] - RESULT[31:0]^2$$

14.6.2.4 Signed Division

When CTRLA.SIGNED is one, both the input and the result will be in 2's complement format. The results of signed division are such that the remainder and dividend have the same sign and the quotient is negative if the dividend and divisor have opposite signs. 16-bit results are sign extended to 32-bits. Note that when the maximum negative number is divided by the minimum negative number, the resulting quotient overflows the signed integer range and will return the maximum negative number with no indication of the overflow. This occurs for 0x80000000 / 0xFFFFFFFF in 32-bit operation and 0x8000 / 0xFFFF in 16-bit operation.

21.7 Register Summary

Offset	Name	Bit Pos.								
0x00	INTENCLR	7:0						CLKFAIL	OSC32KRDY	XOSC32KRDY
0x01		15:8								
0x02		23:16								
0x03		31:24								
0x04	INTENSET	7:0						CLKFAIL	OSC32KRDY	XOSC32KRDY
0x05		15:8								
0x06		23:16								
0x07		31:24								
0x08	INTFLAG	7:0						CLKFAIL	OSC32KRDY	XOSC32KRDY
0x09		15:8								
0x0A		23:16								
0x0B		31:24								
0x0C	STATUS	7:0					CLKSW	CLKFAIL	OSC32KRDY	XOSC32KRDY
0x0D		15:8								
0x0E		23:16								
0x0F		31:24								
0x10 ... 0x13	Reserved									
0x14	XOSC32K	7:0	ONDEMAND	RUNSTDBY		EN1K	EN32K	XTALEN	ENABLE	
0x15		15:8				WRTLOCK		STARTUP[2:0]		
0x16	CFDCTRL	7:0						CFDPRESC	SWBACK	CFDEN
0x17	EVCTRL	7:0								CFDEO
0x18	OSC32K	7:0	ONDEMAND	RUNSTDBY			EN1K	EN32K	ENABLE	
0x19		15:8				WRTLOCK		STARTUP[2:0]		
0x1A		23:16		CALIB[6:0]						
0x1B		31:24								
0x1C	OSCULP32K	7:0								
0x1D		15:8	WRTLOCK			CALIB[4:0]				
0x1E		23:16								
0x1F		31:24								

21.8 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.

All registers with write-access can be write-protected optionally by the peripheral access controller (PAC). Optional Write-Protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write-

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
	WRTLOCK			CALIB[4:0]				
Access	R/W			R/W	R/W	R/W	R/W	R/W
Reset	0			x	x	x	x	x

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bit 15 – WRTLOCK: Write Lock

This bit locks the OSCULP32K register for future writes to fix the OSCULP32K configuration.

Value	Description
0	The OSCULP32K configuration is not locked.
1	The OSCULP32K configuration is locked.

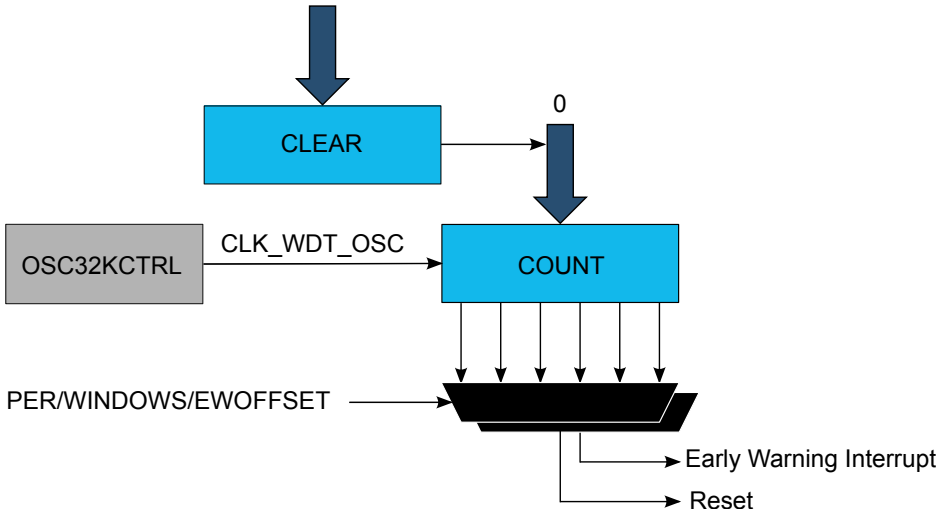
Bits 12:8 – CALIB[4:0]: Oscillator Calibration

These bits control the oscillator calibration.

These bits are loaded from Flash Calibration at startup.

23.3 Block Diagram

Figure 23-1. WDT Block Diagram



23.4 Signal Description

Not applicable.

23.5 Product Dependencies

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

23.5.1 I/O Lines

Not applicable.

23.5.2 Power Management

The WDT can continue to operate in any sleep mode where the selected source clock is running. The WDT interrupts can be used to wake up the device from sleep modes. The events can trigger other operations in the system without exiting sleep modes.

Related Links

PM – Power Manager

23.5.3 Clocks

The WDT bus clock (CLK_WDT_APB) can be enabled and disabled (masked) in the Main Clock module (MCLK).

A 1 kHz oscillator clock (CLK_WDT_OSC) is required to clock the WDT internal counter.

CLK_WDT_OSC is sourced from the clock of the internal ultra-low-power oscillator, OSCULP32K. Due to the ultra-low-power design, the oscillator is not very accurate, and so the exact time-out period may vary from device to device. This variation must be kept in mind when designing software that uses the WDT to ensure that the time-out periods used are valid for all devices.

The counter clock CLK_WDT_OSC is asynchronous to the bus clock (CLK_WDT_APB). Due to this asynchronicity, writing to certain registers will require synchronization between the clock domains. Refer to Synchronization for further details.

Name: EVCTRL
Offset: 0x04
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

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
	OVFEO							CMPEO0
Access	R/W							R/W
Reset	0							0

Bit	7	6	5	4	3	2	1	0
	PEREOn	PEREOn	PEREOn	PEREOn	PEREOn	PEREOn	PEREOn	PEREOn
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 – CMPEO0: Compare 0 Event Output Enable

Value	Description
0	Compare 0 event is disabled and will not be generated.
1	Compare 0 event is enabled and will be generated for every compare match.

Bits 7:0 – PEREOn: Periodic Interval n Event Output Enable [n = 7..0]

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

24.8.3 Interrupt Enable Clear in COUNT32 mode (CTRLA.MODE=0)

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 (INTENSET) register.

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.

24.10.1 Control A in COUNT16 mode (CTRLA.MODE=1)

Name: CTRLA
Offset: 0x00
Reset: 0x0000
Property: PAC Write-Protection, Enable-Protected, Write-Synchronized

Bit	15	14	13	12	11	10	9	8
	COUNTSYNC				PRESCALER[3:0]			
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
					MODE[1:0]		ENABLE	SWRST
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit 15 – COUNTSYNC: COUNT Read Synchronization Enable
The COUNT register requires synchronization when reading. Disabling the synchronization will prevent reading valid values from the COUNT register.
This bit is not enable-protected.

Value	Description
0	COUNT read synchronization is disabled
1	COUNT read synchronization is enabled

Bits 11:8 – PRESCALER[3:0]: Prescaler
These bits define the prescaling factor for the RTC clock source (GCLK_RTC) to generate the counter clock (CLK_RTC_CNT). Periodic events and interrupts are not available when the prescaler is off. These bits are not synchronized.

Value	Name	Description
0x0	OFF	CLK_RTC_CNT = GCLK_RTC/1
0x1	DIV1	CLK_RTC_CNT = GCLK_RTC/1
0x2	DIV2	CLK_RTC_CNT = GCLK_RTC/2
0x3	DIV4	CLK_RTC_CNT = GCLK_RTC/4
0x4	DIV8	CLK_RTC_CNT = GCLK_RTC/8
0x5	DIV16	CLK_RTC_CNT = GCLK_RTC/16

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

25.6.3.7 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.

- CRC-16:
  - Polynomial: $x^{16}+ x^{12}+ x^5+ 1$
  - Hex value: 0x1021
- CRC-32:
  - Polynomial: $x^{32}+x^{26}+ x^{23}+ x^{22}+x^{16}+ x^{12}+ x^{11}+ x^{10}+ x^8+ x^7+ x^5+ x^4+ x^2+ x + 1$
  - Hex value: 0x04C11DB7

The data source for the CRC engine can either be one of the DMA channels or the APB bus interface, and must be selected by writing to the CRC Input Source bits in the CRC Control register (CRCCTRL.CRCSRC). The CRC engine then takes data input from the selected source and generates a checksum based on these data. The checksum is available in the CRC Checksum register (CRCCHKSUM). When CRC-32 polynomial is used, the final checksum read is bit reversed and complemented, as shown in Figure 25-16.

The CRC polynomial is selected by writing to the CRC Polynomial Type bit in the CRC Control register (CRCCTRL.CRCPOLY), the default is CRC-16. The CRC engine operates on byte only. When the DMA is used as data source for the CRC engine, the DMA channel beat size setting will be used. When used with APB bus interface, the application must select the CRC Beat Size bit field of CRC Control register (CRCCTRL.CRCBEATSIZE). 8-, 16-, or 32-bit bus transfer access type is supported. The corresponding number of bytes will be written in the CRCDATAIN register and the CRC engine will operate on the input data in a byte by byte manner.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

25.8.16 Write-Back Memory Section Base Address

Name: WRBADDR
Offset: 0x38
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

25.8.17 Channel ID

27.4 Signal Description

Not applicable.

27.5 Product Dependencies

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

27.5.1 Power Management

The NVMCTRL will continue to operate in any sleep mode where the selected source clock is running. The NVMCTRL interrupts can be used to wake up the device from sleep modes.

The Power Manager will automatically put the NVM block into a low-power state when entering sleep mode. This is based on the Control B register (CTRLB) SLEEPPRM bit setting. Refer to the CTRLB.SLEEPPRM register description for more details. The NVM block goes into low-power mode automatically when the device enters STANDBY mode regardless of SLEEPPRM. The NVM Page Buffer is lost when the NVM goes into low power mode therefore a write command must be issued prior entering the NVM low power mode. NVMCTRL SLEEPPRM can be disabled to avoid such loss when the CPU goes into sleep except if the device goes into STANDBY mode for which there is no way to retain the Page Buffer.

Related Links

PM – Power Manager

27.5.2 Clocks

Two synchronous clocks are used by the NVMCTRL. One is provided by the AHB bus (CLK_NVMCTRL_AHB) and the other is provided by the APB bus (CLK_NVMCTRL_APB). For higher system frequencies, a programmable number of wait states can be used to optimize performance. When changing the AHB bus frequency, the user must ensure that the NVM Controller is configured with the proper number of wait states. Refer to the Electrical Characteristics for the exact number of wait states to be used for a particular frequency range.

Related Links

Electrical Characteristics 85°C (SAM C20/C21 E/G/J)

27.5.3 Interrupts

The NVM Controller interrupt request line is connected to the interrupt controller. Using the NVMCTRL interrupt requires the interrupt controller to be programmed first.

27.5.4 Debug Operation

When an external debugger forces the CPU into debug mode, the peripheral continues normal operation.

Access to the NVM block can be protected by the security bit. In this case, the NVM block will not be accessible. See the section on the NVMCTRL Security Bit for details.

27.5.5 Register Access Protection

All registers with write-access are optionally write-protected by the Peripheral Access Controller (PAC), except the following registers:

- Interrupt Flag Status and Clear register (INTFLAG)
- Status register (STATUS)

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.

Refer to Synchronization

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.

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

Refer to Register Access Protection.

32.8.1 Control A

Name: CTRLA
Offset: 0x00 [ID-00000e74]
Reset: 0x00000000
Property: PAC Write-Protection, Enable-Protected, Write-Synchronized

Bit	31	30	29	28	27	26	25	24
		DORD	CPOL	CPHA	FORM[3:0]			
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
			DIPO[1:0]				DOPO[1:0]	
Access			R/W	R/W			R/W	R/W
Reset			0	0			0	0

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

Bit	7	6	5	4	3	2	1	0
	RUNSTDBY			MODE[2:0]			ENABLE	SWRST
Access	R/W			R/W	R/W	R/W	R/W	R/W
Reset	0			0	0	0	0	0

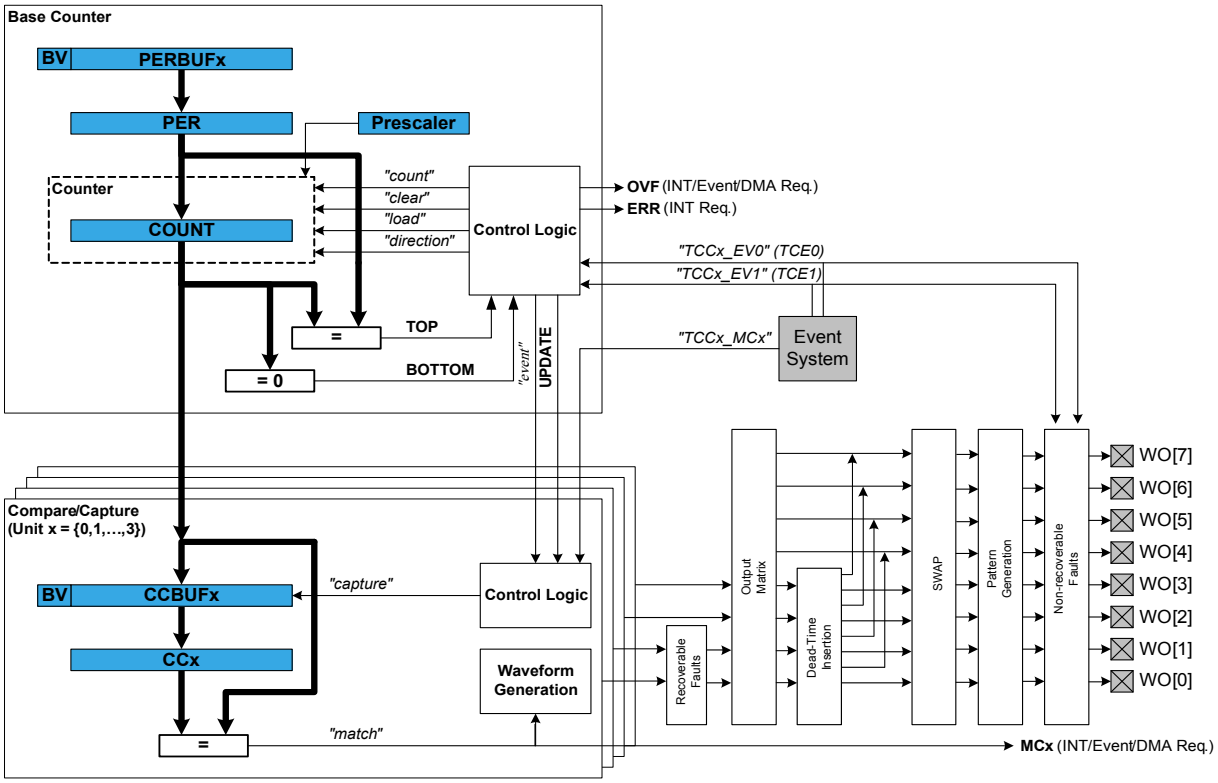
Bit 30 – DORD: Data Order
This bit selects the data order when a character is shifted out from the shift register.

This bit is not synchronized.

  - Two non-recoverable fault sources
  - Debugger can be source of non-recoverable fault
- Input events:
  - Two input events for counter
  - One input event for each channel
- Output events:
  - Three output events (Count, Re-Trigger and Overflow) available for counter
  - One Compare Match/Input Capture event output for each channel
- Interrupts:
  - Overflow and Re-Trigger interrupt
  - Compare Match/Input Capture interrupt
  - Interrupt on fault detection
- Can be used with DMA and can trigger DMA transactions

36.3 Block Diagram

Figure 36-1. Timer/Counter for Control Applications - Block Diagram



36.4 Signal Description

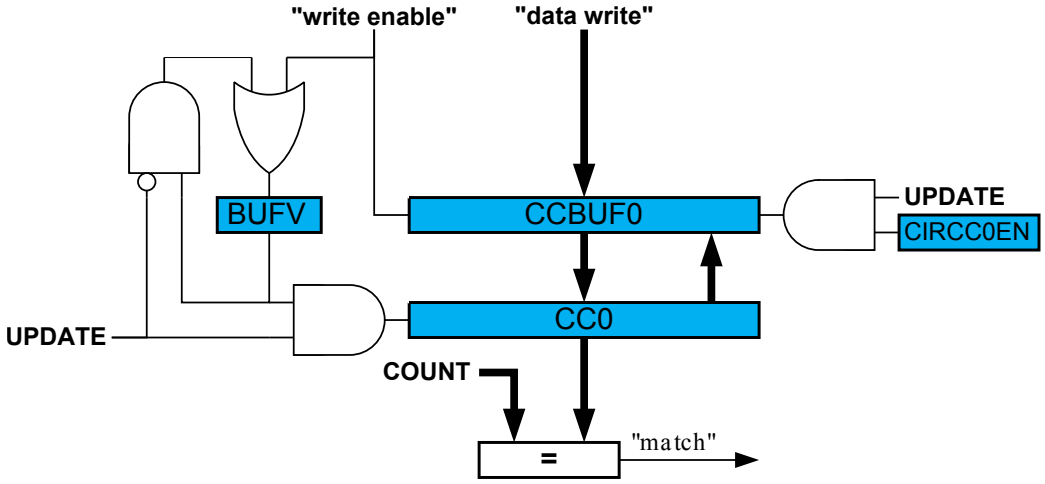
Pin Name	Type	Description
TCCx/WO[0]	Digital output	Compare channel 0 waveform output
TCCx/WO[1]	Digital output	Compare channel 1 waveform output

One-shot operation can be enabled by writing a '1' to the One-Shot bit in the Control B Set register (CTRLBSET.ONESHOT) and disabled by writing a '1' to CTRLBCLR.ONESHOT. When enabled, the TCC will count until an overflow or underflow occurs and stop counting. The one-shot operation can be restarted by a re-trigger software command, a re-trigger event or a start event. When the counter restarts its operation, STATUS.STOP is automatically cleared.

36.6.3.2 Circular Buffer

The Period register (PER) and the compare channels register (CC0 to CC3) support circular buffer operation. When circular buffer operation is enabled, the PER or CCx values are copied into the corresponding buffer registers at each update condition. Circular buffering is dedicated to RAMP2, RAMP2A, and DSBOTH operations.

Figure 36-17. Circular Buffer on Channel 0



36.6.3.3 Dithering Operation

The TCC supports dithering on Pulse-width or Period on a 16, 32 or 64 PWM cycles frame.

Dithering consists in adding some extra clocks cycles in a frame of several PWM cycles, and can improve the accuracy of the *average* output pulse width and period. The extra clock cycles are added on some of the compare match signals, one at a time, through a "blue noise" process that minimizes the flickering on the resulting dither patterns.

Dithering is enabled by writing the corresponding configuration in the Enhanced Resolution bits in CTRLA register (CTRLA.RESOLUTION):

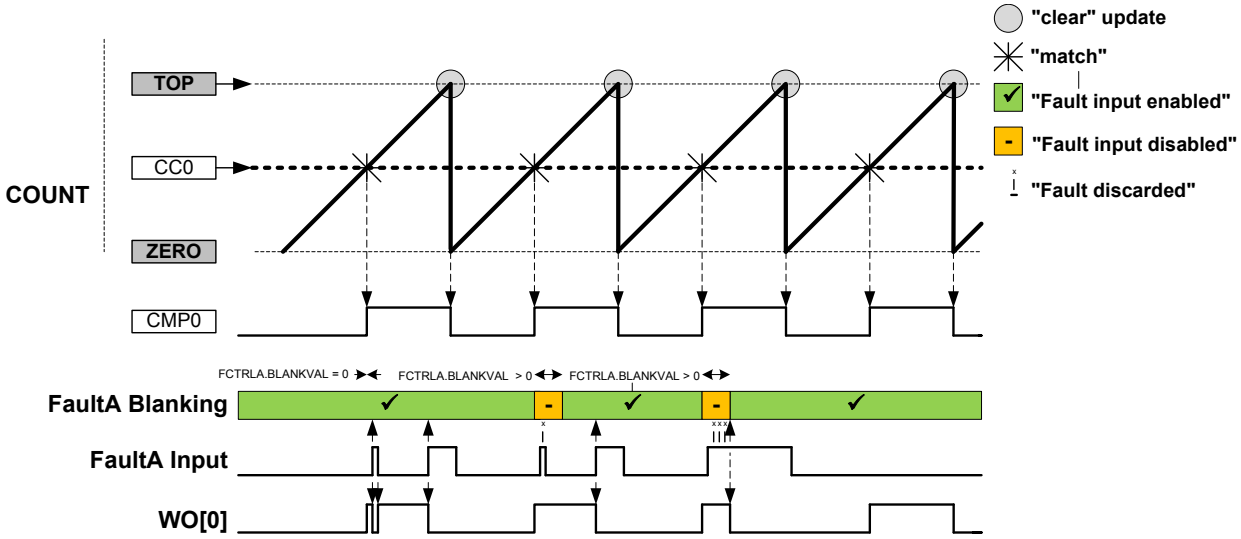
- DITH4 enable dithering every 16 PWM frames
- DITH5 enable dithering every 32 PWM frames
- DITH6 enable dithering every 64 PWM frames

The DITHERCY bits of COUNT, PER and CCx define the number of extra cycles to add into the frame (DITHERCY bits from the respective COUNT, PER or CCx registers). The remaining bits of COUNT, PER, CCx define the compare value itself.

The pseudo code, giving the extra cycles insertion regarding the cycle is:

```
int extra_cycle(resolution, dithercy, cycle){
  int MASK;
  int value
  switch (resolution){
    DITH4: MASK = 0x0f;
    DITH5: MASK = 0x1f;
    DITH6: MASK = 0x3f;
  }
```

Figure 36-22. Fault Blanking in RAMP1 Operation with Inverted Polarity



Fault Qualification

This is enabled by writing a '1' to the Fault n Qualification bit in the Recoverable Fault n Configuration register (FCTRLn.QUAL). When the recoverable fault qualification is enabled (FCTRLn.QUAL=1), the fault input is disabled all the time the corresponding channel output has an inactive level, as shown in the figures below.

Figure 36-23. Fault Qualification in RAMP1 Operation

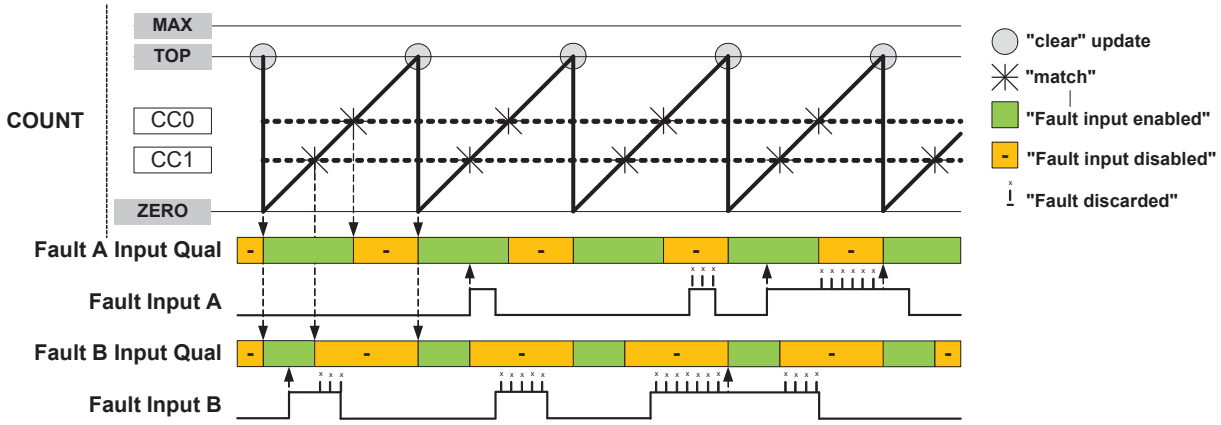
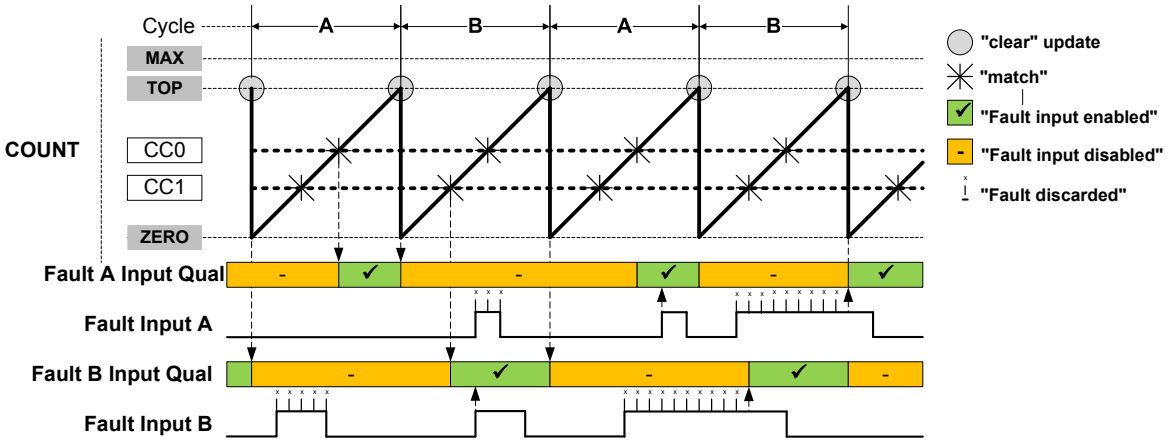


Figure 36-24. Fault Qualification in RAMP2 Operation with Inverted Polarity



Bit	15	14	13	12	11	10	9	8
			DUALSEL[1:0]			WINMODE[2:0]		
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
	R2R		RESSEL[1:0]		CORREN	FREERUN	LEFTADJ	DIFFMODE
Access	R/W		R/W	R/W	R/W	R/W	R/W	R/W
Reset	0		0	0	0	0	0	0

Bits 13:12 – DUALSEL[1:0]: Dual Mode Trigger Selection
These bits define the trigger mode. These bits are available in the master ADC and have no effect if the master-slave operation is disabled (ADC1.CTRLA.SLAVEEN=0).

Value	Name	Description
0x0	BOTH	Start event or software trigger will start a conversion on both ADCs.
0x1	INTERLEAVE	Start event or software trigger will alternatingly start a conversion on ADC0 and ADC1.
0x2 - 0x3	-	Reserved

Bits 10:8 – WINMODE[2:0]: Window Monitor Mode
These bits enable and define the window monitor mode.

Value	Name	Description
0x0	DISABLE	No window mode (default)
0x1	MODE1	RESULT > WINLT
0x2	MODE2	RESULT < WINUT
0x3	MODE3	WINLT < RESULT < WINUT
0x4	MODE4	WINUT < RESULT < WINLT
0x5 - 0x7		Reserved

Bit 7 – R2R: Rail-to-Rail Operation

Value	Description
0	Disable rail-to-rail operation.
1	Enable rail-to-rail operation to increase the allowable range of the input common mode voltage (V_{CMIN}). When R2R is one, a sampling period of four cycles is required. Offset compensation (SAMPCTRL.OFFCOMP) must be written to one when using this period.

Bits 5:4 – RESSEL[1:0]: Conversion Result Resolution
These bits define whether the ADC completes the conversion 12-, 10- or 8-bit result resolution.

Value	Name	Description
0x0	12BIT	12-bit result
0x1	16BIT	For averaging mode output
0x2	10BIT	10-bit result
0x3	8BIT	8-bit result

Bit 3 – CORREN: Digital Correction Logic Enabled

Value	Description
0	Disable the digital result correction.
1	Enable the digital result correction. The ADC conversion result in the RESULT register is then corrected for gain and offset based on the values in the GAINCORR and OFFSETCORR registers. Conversion time will be increased by 13 cycles according to the value in the Offset Correction Value bit group in the Offset Correction register.

Bit 2 – FREERUN: Free Running Mode

Value	Description
0	The ADC run in single conversion mode.
1	The ADC is in free running mode and a new conversion will be initiated when a previous conversion completes.

Bit 1 – LEFTADJ: Left-Adjusted Result

Value	Description
0	The ADC conversion result is right-adjusted in the RESULT register.
1	The ADC conversion result is left-adjusted in the RESULT register. The high byte of the 12-bit result will be present in the upper part of the result register. Writing this bit to zero (default) will right-adjust the value in the RESULT register.

Bit 0 – DIFFMODE: Differential Mode

Value	Description
0	The ADC is running in singled-ended mode.
1	The ADC is running in differential mode. In this mode, the voltage difference between the MUXPOS and MUXNEG inputs will be converted by the ADC.

38.8.11 Average Control

Name: AVGCTRL
Offset: 0x0C [ID-0000120e]
Reset: 0x00
Property: PAC Write-Protection, Write-Synchronized

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

Bits 6:4 – ADJRES[2:0]: Adjusting Result / Division Coefficient
These bits define the division coefficient in 2n steps.

Bits 3:0 – SAMPLENUM[3:0]: Number of Samples to be Collected
These bits define how many samples are added together. The result will be available in the Result register (RESULT). Note: if the result width increases, CTRLC.RESSEL must be changed.

Value	Description
0x0	1 sample
0x1	2 samples

Bit	15	14	13	12	11	10	9	8
	SKPCNT[3:0]					OSR[2:0]		
Access	R/W	R/W	R/W	R/W		R/W	R/W	R/W
Reset	0	0	0	0		0	0	0

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

Bits 15:12 – SKPCNT[3:0]: Skip Count
How many skip samples before retrieve the first valid sample.

The first valid sample starts from the third sample onward.

Bits 10:8 – OSR[2:0]: Over Sampling Ratio
OSR is the Over Sampling Ratio which can be modified to change the output data rate.

The OSR must never be changed while the SDADC is running. One must first place the SDADC in reset state, modify the OSR and then run the SDADC again.

Example: The sampling rate of the SDADC is 1.5Msps/OSR. The maximum sampling rate is then 1.5MSPS/OSR64 ≅ 23.4ksps and the minimum sampling rate is 1.5Msps/OSR1024 ≅ 1.5ksps

Value	Name	Description
0x0	OSR64	Over Sampling Ratio is 64
x01	OSR128	Over Sampling Ratio is 128
0x2	OSR256	Over Sampling Ratio is 256
0x3	OSR512	Over Sampling Ratio is 512
0x4	OSR1024	Over Sampling Ratio is 1024
0x4 - 0xF	-	Reserved

Bits 7:0 – PRESCALER[7:0]: Prescaler Configuration
The ADC uses the SDADC Clock to perform conversions.

The CLK_SDADC_PRESCAL clock range is between CLK_GEN_SDADC/2, if PRESCAL is 0, and CLK_GEN_SDADC/512, if PRESCAL is set to 255 (0xFF). PRESCAL must be programmed in order to provide an CLK_SDADC_PRESCAL clock frequency according to the parameters given in the product Electrical Characteristics section.

39.8.4 Event Control

Name: EVCTRL
Offset: 0x04 [ID-0000243d]
Reset: 0x00
Property: PAC Write-Protection, Enable-Protected

40.7 Register Summary

Offset	Name	Bit Pos.								
0x00	CTRLA	7:0							ENABLE	SWRST
0x01	CTRLB	7:0					STARTx	STARTx	STARTx	STARTx
0x02	EVCTRL	7:0			WINEOx	WINEOx	COMPEOx	COMPEOx	COMPEOx	COMPEOx
0x03		15:8	INVEIx	INVEIx	INVEIx	INVEIx	COMPEIx	COMPEIx	COMPEIx	COMPEIx
0x04	INTENCLR	7:0			WINx	WINx	COMPx	COMPx	COMPx	COMPx
0x05	INTENSET	7:0			WINx	WINx	COMPx	COMPx	COMPx	COMPx
0x06	INTFLAG	7:0			WINx	WINx	COMPx	COMPx	COMPx	COMPx
0x07	STATUSA	7:0	WSTATE1[1:0]		WSTATE0[1:0]		STATEx	STATEx	STATEx	STATEx
0x08	STATUSB	7:0					READYx	READYx	READYx	READYx
0x09	DBGCTRL	7:0								DBGRUN
0x0A	WINCTRL	7:0		WINTSEL1[1:0]		WEN1		WINTSEL0[1:0]		WEN0
0x0B	Reserved									
0x0C	SCALERn0	7:0	VALUE[5:0]							
0x0D	SCALERn1	7:0	VALUE[5:0]							
0x0E	SCALERn2	7:0	VALUE[5:0]							
0x0F	SCALERn3	7:0	VALUE[5:0]							
0x10	COMPCTRL0	7:0		RUNSTDBY		INTSEL[1:0]		SINGLE	ENABLE	
0x11		15:8	SWAP	MUXPOS[2:0]				MUXNEG[2:0]		
0x12		23:16					HYSTEN		SPEED[1:0]	
0x13		31:24			OUT[1:0]			FLEN[2:0]		
0x14	COMPCTRL1	7:0		RUNSTDBY		INTSEL[1:0]		SINGLE	ENABLE	
0x15		15:8	SWAP	MUXPOS[2:0]				MUXNEG[2:0]		
0x16		23:16					HYSTEN		SPEED[1:0]	
0x17		31:24			OUT[1:0]			FLEN[2:0]		
0x18	COMPCTRL2	7:0		RUNSTDBY		INTSEL[1:0]		SINGLE	ENABLE	
0x19		15:8	SWAP	MUXPOS[2:0]				MUXNEG[2:0]		
0x1A		23:16					HYSTEN		SPEED[1:0]	
0x1B		31:24			OUT[1:0]			FLEN[2:0]		
0x1C	COMPCTRL3	7:0		RUNSTDBY		INTSEL[1:0]		SINGLE	ENABLE	
0x1D		15:8	SWAP	MUXPOS[2:0]				MUXNEG[2:0]		
0x1E		23:16					HYSTEN		SPEED[1:0]	
0x1F		31:24			OUT[1:0]			FLEN[2:0]		
0x20	SYNCBUSY	7:0		COMPCTRLx	COMPCTRLx	COMPCTRLx	COMPCTRLx	WINCTRL	ENABLE	SWRST
0x21		15:8								
0x22		23:16								
0x23		31:24								

40.8 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.

Mode	conditions	Ta	Vcc	Typ.	Max.	Units
	CPU running a While 1 algorithm, with GCLKIN as reference	25°C	5.0V	71*Freq+160	78*Freq+162	µA (with freq in MHz)
		85°C	5.0V	71*Freq+253	74*Freq+447	
	CPU running a Fibonacci algorithm	25°C	5.0V	4.7	5.2	mA
		85°C	5.0V	4.8	5.3	
	CPU running a Fibonacci algorithm	25°C	3.0V	4.7	5.1	mA
		85°C	3.0V	4.8	5.3	
	CPU running a Fibonacci algorithm, with GCLKIN as reference	25°C	5.0V	90*Freq+163	99*Freq+168	µA (with freq in MHz)
		85°C	5.0V	90*Freq+258	95*Freq+450	
	CPU running a CoreMark algorithm	25°C	5.0V	5.9	6.4	mA
		85°C	5.0V	6.1	6.6	
	CPU running a CoreMark algorithm	25°C	3.0V	5.2	5.7	mA
		85°C	3.0V	5.4	5.8	
	CPU running a CoreMark algorithm, with GCLKIN as reference	25°C	5.0V	115*Freq+167	126*Freq+167	µA (with freq in MHz)
		85°C	5.0V	117*Freq+261	122*Freq+454	
IDLE		25°C	5.0V	1.2	1.3	mA
		85°C	5.0V	1.3	2.3	
STANDBY	XOSC32K running RTC running at 1kHz	25°C	5.0V	15.9	37.0	µA
		85°C	5.0V	89.9	302.0	
	XOSC32K and RTC stopped	25°C	5.0V	14.6	35.0	
		85°C	5.0V	87.8	300.0	

1. These are based on characterization.

45.8 Wake-Up Time

Conditions:

- V_{DD} = 5.0V
- CPU clock = OSC48M @8Mhz
- 0 Wait-state
- Cache enabled
- Flash in WAKEUPINSTANT mode (NVMCTRL.CTRLB.SLEEPPRM=1)

CPU sets an IO by writing PORT->IOBUS without jumping in an interrupt handler (Cortex M0+ register PRIMASK=1). The wakeup time is measured between the edge of the wakeup input signal and the edge of the GPIO pin.

Figure 49-4. External Analog Reference Schematic With One Reference

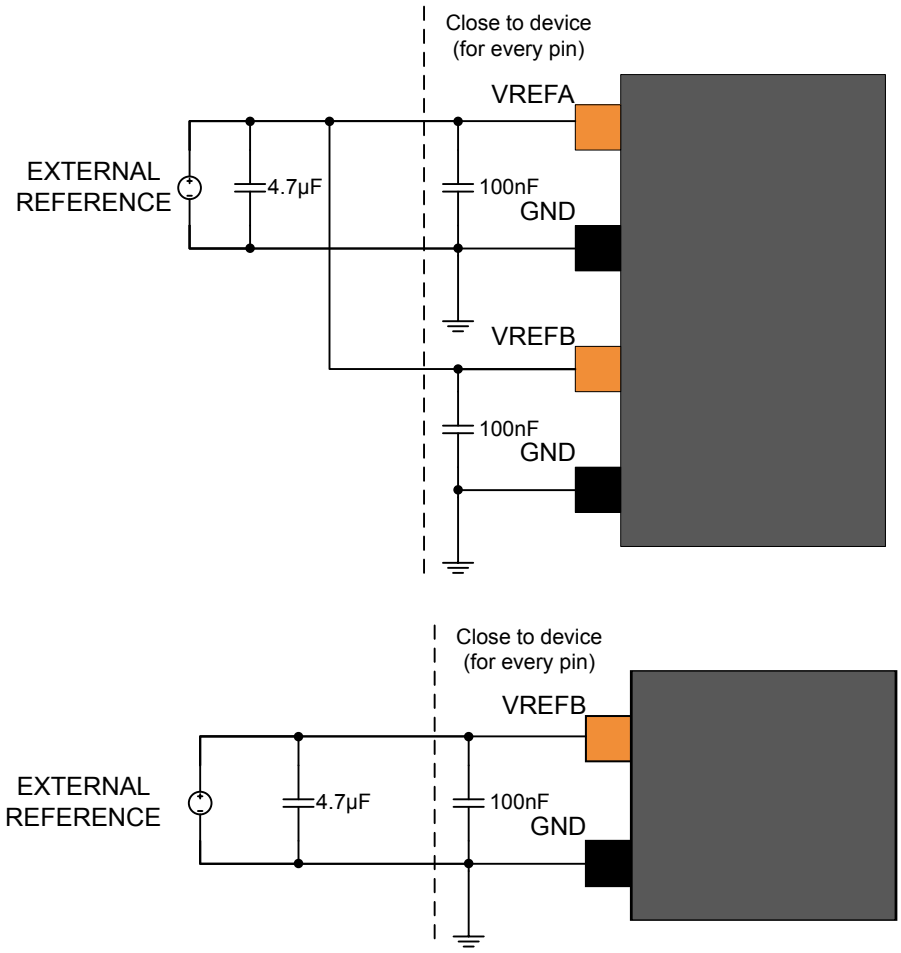


Table 49-2. External Analog Reference Connections

Signal Name	Recommended Pin Connection	Description
VREFA	2.0V 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 VREFA pin on the analog port.
VREFB	1.0V to 5.5V for SDADC Decoupling/filtering capacitors: 100nF(1)(2) and 4.7µF(1)	External reference from VREFB pin on the analog port.
GND		Ground

Note:

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.