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
Peripherals	Brown-out Detect/Reset, DMA, POR, WDT
Number of I/O	52
Program Memory Size	32KB (32K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	4K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 5.5V
Data Converters	A/D 12x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	64-TQFP
Supplier Device Package	64-TQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atsamc20j15a-aut

	Pin[1]		I/O Pin	Supply	A				B[2][3]				C	D	E	F	G	H	I
SAM C21E	SAM C21G	SAM C21J			EIC	REF	ADC0	ADC1	AC	PTC	DAC	SDADC	SERCOM[2][3][4]	SERCOM-ALT[4]	TC TCC	TCC	COM	AC/GCLK	CCL
																			IN[3]
12	14	18	PA09	VDDIO	EXTINT[9]		AIN[9]	AIN[11]		X[1]/Y[17]			SERCOM0/ PAD[1]	SERCOM2/ PAD[1]	TCC0/WO[1]	TCC1/ WO[3]			CCL1/ IN[4]
13	15	19	PA10	VDDIO	EXTINT[10]		AIN[10]			X[2]/Y[18]			SERCOM0/ PAD[2]	SERCOM2/ PAD[2]	TCC1/WO[0]	TCC0/ WO[2]		GCLK_IO[4]	CCL1/ IN[5]
14	16	20	PA11	VDDIO	EXTINT[11]		AIN[11]			X[3]/Y[19]			SERCOM0/ PAD[3]	SERCOM2/ PAD[3]	TCC1/WO[1]	TCC0/ WO[3]		GCLK_IO[5]	CCL1/ OUT[1]
	19	23	PB10	VDDIO	EXTINT[10]									SERCOM4/ PAD[2]	TC1/WO[0]	TCC0/ WO[4]	CAN1/TX	GCLK_IO[4]	CCL1/ IN[5]
	20	24	PB11	VDDIO	EXTINT[11]									SERCOM4/ PAD[3]	TC1/WO[1]	TCC0/ WO[5]	CAN1/RX	GCLK_IO[5]	CCL1/ OUT[1]
		25	PB12	VDDIO	EXTINT[12]					X[12]/Y[28]			SERCOM4/ PAD[0]		TC0/WO[0]	TCC0/ WO[6]		GCLK_IO[6]	
		26	PB13	VDDIO	EXTINT[13]					X[13]/Y[29]			SERCOM4/ PAD[1]		TC0/WO[1]	TCC0/ WO[7]		GCLK_IO[7]	
		27	PB14	VDDIO	EXTINT[14]					X[14]/Y[30]			SERCOM4/ PAD[2]		TC1/WO[0]		CAN1/TX	GCLK_IO[0]	CCL3/ IN[9]
		28	PB15	VDDIO	EXTINT[15]					X[15]/Y[31]			SERCOM4/ PAD[3]		TC1/WO[1]		CAN1/RX	GCLK_IO[1]	CCL3/ IN[10]
	21	29	PA12	VDDIO	EXTINT[12]								SERCOM2/ PAD[0]	SERCOM4/ PAD[0]	TCC2/WO[0]	TCC0/ WO[6]		AC/CMP[0]	
	22	30	PA13	VDDIO	EXTINT[13]								SERCOM2/ PAD[1]	SERCOM4/ PAD[1]	TCC2/WO[1]	TCC0/ WO[7]		AC/CMP[1]	
15	23	31	PA14	VDDIO	EXTINT[14]								SERCOM2/ PAD[2]	SERCOM4/ PAD[2]	TC4/WO[0]	TCC0/ WO[4]		GCLK_IO[0]	
16	24	32	PA15	VDDIO	EXTINT[15]								SERCOM2/ PAD[3]	SERCOM4/ PAD[3]	TC4/WO[1]	TCC0/ WO[5]		GCLK_IO[1]	
17	25	35	PA16	VDDIO	EXTINT[0]					X[4]/Y[20]			SERCOM1/ PAD[0]	SERCOM3/ PAD[0]	TCC2/WO[0]	TCC0/ WO[6]		GCLK_IO[2]	CCL0/ IN[0]
18	26	36	PA17	VDDIO	EXTINT[1]					X[5]/Y[21]			SERCOM1/ PAD[1]	SERCOM3/ PAD[1]	TCC2/WO[1]	TCC0/ WO[7]		GCLK_IO[3]	CCL0/ IN[1]
19	27	37	PA18	VDDIO	EXTINT[2]					X[6]/Y[22]			SERCOM1/ PAD[2]	SERCOM3/ PAD[2]	TC4/WO[0]	TCC0/ WO[2]		AC/CMP[0]	CCL0/ IN[2]
20	28	38	PA19	VDDIO	EXTINT[3]					X[7]/Y[23]			SERCOM1/ PAD[3]	SERCOM3/ PAD[3]	TC4/WO[1]	TCC0/ WO[3]		AC/CMP[1]	CCL0/ OUT[0]
		39	PB16	VDDIO	EXTINT[0]								SERCOM5/ PAD[0]		TC2/WO[0]	TCC0/ WO[4]		GCLK_IO[2]	CCL3/ IN[11]
		40	PB17	VDDIO	EXTINT[1]								SERCOM5/ PAD[1]		TC2/WO[1]	TCC0/ WO[5]		GCLK_IO[3]	CCL3/ OUT[3]
	29	41	PA20	VDDIO	EXTINT[4]					X[8]/Y[24]			SERCOM5/ PAD[2]	SERCOM3/ PAD[2]	TC3/WO[0]	TCC0/ WO[6]		GCLK_IO[4]	
	30	42	PA21	VDDIO	EXTINT[5]					X[9]/Y[25]			SERCOM5/ PAD[3]	SERCOM3/ PAD[3]	TC3/WO[1]	TCC0/ WO[7]		GCLK_IO[5]	
21	31	43	PA22	VDDIO	EXTINT[6]					X[10]/Y[26]			SERCOM3/ PAD[0]	SERCOM5/ PAD[0]	TC0/WO[0]	TCC0/ WO[4]		GCLK_IO[6]	CCL2/ IN[6]
22	32	44	PA23	VDDIO	EXTINT[7]					X[11]/Y[27]			SERCOM3/ PAD[1]	SERCOM5/ PAD[1]	TC0/WO[1]	TCC0/ WO[5]		GCLK_IO[7]	CCL2/ IN[7]
23	33	45	PA24	VDDIO	EXTINT[12]								SERCOM3/ PAD[2]	SERCOM5/ PAD[2]	TC1/WO[0]	TCC1/ WO[2]	CAN0/TX	AC/CMP[2]	CCL2/ IN[8]
24	34	46	PA25	VDDIO	EXTINT[13]								SERCOM3/ PAD[3]	SERCOM5/ PAD[3]	TC1/WO[1]	TCC1/ WO[3]	CAN0/RX	AC/CMP[3]	CCL2/ OUT[2]
	37	49	PB22	VDDIN	EXTINT[6]									SERCOM5/ PAD[2]	TC3/WO[0]		CAN0/TX	GCLK_IO[0]	CCL0/ IN[0]
	38	50	PB23	VDDIN	EXTINT[7]									SERCOM5/ PAD[3]	TC3/WO[1]		CAN0/RX	GCLK_IO[1]	CCL0/ OUT[0]
25	39	51	PA27	VDDIN	EXTINT[15]													GCLK_IO[0]	
27	41	53	PA28	VDDIN	EXTINT[8]													GCLK_IO[0]	
31	45	57	PA30	VDDIN	EXTINT[10]									SERCOM1/ PAD[2]	TCC1/WO[0]		CORTEX_M0P/ SWCLK	GCLK_IO[0]	CCL1/ IN[3]
32	46	58	PA31	VDDIN	EXTINT[11]									SERCOM1/ PAD[3]	TCC1/WO[1]		CORTEX_M0P/ SWDIO		CCL1/ OUT[1]
		59	PB30	VDDIN	EXTINT[14]									SERCOM5/ PAD[0]	TCC0/WO[0]	TCC1/ WO[2]		AC/CMP[2]	
		60	PB31	VDDIN	EXTINT[15]									SERCOM5/ PAD[1]	TCC0/WO[1]	TCC1/ WO[3]		AC/CMP[3]	
		61	PB00	VDDANA	EXTINT[0]			AIN[0]		Y[6]				SERCOM5/ PAD[2]	TC3/WO[0]				CCL0/ IN[1]
		62	PB01	VDDANA	EXTINT[1]			AIN[1]		Y[7]				SERCOM5/ PAD[3]	TC3/WO[1]				CCL0/ IN[2]
	47	63	PB02	VDDANA	EXTINT[2]			AIN[2]		Y[8]				SERCOM5/ PAD[0]	TC2/WO[0]				CCL0/ OUT[0]
	48	64	PB03	VDDANA	EXTINT[3]			AIN[3]		Y[9]				SERCOM5/ PAD[1]	TC2/WO[1]				

Table 10-4. Interrupt Line Mapping, SAM C20

Peripheral Source	NVIC Line
EIC NMI – External Interrupt Controller	NMI
PM – Power Manager MCLK - Main Clock OSCCTRL - Oscillators Controller OSC32KCTRL - 32kHz Oscillators Controller SUPC - Supply Controller PAC - Protection Access Controller	0
WDT – Watchdog Timer	1
RTC – Real Time Clock	2
EIC – External Interrupt Controller	3
FREQM – Frequency Meter	4
Reserved	5
NVMCTRL – Non-Volatile Memory Controller	6
DMAC - Direct Memory Access Controller	7
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
Reserved	15
Reserved	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	21

The “set protection” operation will set the write access protection for the peripheral selected by WRCTRL.PERID. Write accesses are not allowed for the registers with write protection property in this peripheral.

The “set and lock protection” operation will set the write access protection for the peripheral selected by WRCTRL.PERID and locks the access rights of the selected peripheral registers. The write access protection will only be cleared by a hardware reset.

The peripheral access control status can be read from the corresponding STATUSn register.

11.5.2.6 Write Access Protection Management Errors

Only word-wise writes to the WRCTRL register will effectively change the access protection. Other type of accesses will have no effect and will cause a PAC write access error. This error is reported in the INTFLAGn.PAC bit corresponding to the PAC module.

PAC also offers an additional safety feature for correct program execution with an interrupt generated on double write clear protection or double write set protection. If a peripheral is write protected and a subsequent set protection operation is detected then the PAC returns an error, and similarly for a double clear protection operation.

In addition, an error is generated when writing a “set and lock” protection to a write-protected peripheral or when a write access is done to a locked set protection. This can be used to ensure that the application follows the intended program flow by always following a write protect with an unprotect and conversely. However in applications where a write protected peripheral is used in several contexts, e.g. interrupt, care should be taken so that either the interrupt can not happen while the main application or other interrupt levels manipulates the write protection status or when the interrupt handler needs to unprotect the peripheral based on the current protection status by reading the STATUS register.

The errors generated while accessing the PAC module registers (eg. key error, double protect error...) will set the INTFLAGn.PAC flag.

11.5.2.7 AHB Slave Bus Errors

The PAC module reports errors occurring at the AHB Slave bus level. These errors are generated when an access is performed at an address where no slave (bridge or peripheral) is mapped . These errors are reported in the corresponding bits of the INTFLAGAHB register.

11.5.2.8 Generating Events

The PAC module can also generate an event when any of the Interrupt Flag registers bit are set. To enable the PAC event generation, the control bit EVCTRL.ERREO must be set a '1'.

11.5.3 DMA Operation

Not applicable.

11.5.4 Interrupts

The PAC has the following interrupt source:

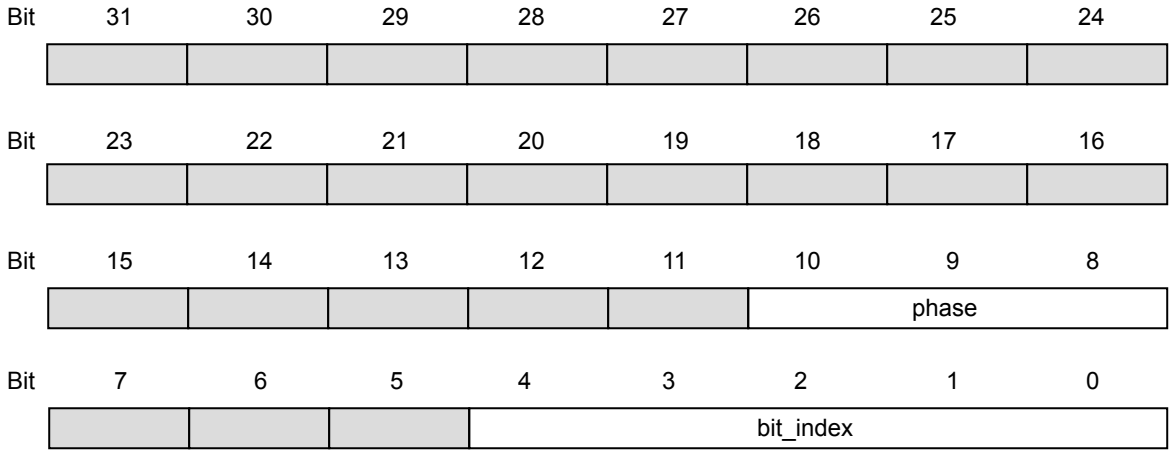
- Error (ERR): Indicates that a peripheral access violation occurred in one of the peripherals controlled by the PAC module, or a bridge error occurred in one of the bridges reported by the PAC
  - This interrupt is a synchronous wake-up source.

Each interrupt source has an interrupt flag associated with it. The interrupt flag in the Interrupt Flag Status and Clear (INTFLAGAHB and INTFLAGn) registers 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,

- ADDR.AMOD=0: exit-on-error (default)
  In this mode, the algorithm terminates either when a fault is detected or on successful completion. In both cases, STATUSA.DONE is set. If an error was detected, STATUSA.FAIL will be set. User then can read the DATA and ADDR registers to locate the fault.
- ADDR.AMOD=1: pause-on-error
  In this mode, the MBIST algorithm is paused when an error is detected. In such a situation, only STATUSA.FAIL is asserted. The state machine waits for user to clear STATUSA.FAIL by writing a '1' in STATUSA.FAIL to resume. Prior to resuming, user can read the DATA and ADDR registers to locate the fault.

4. Locating Faults
   If the test stops with STATUSA.FAIL set, one or more bits failed the test. The test stops at the first detected error. The position of the failing bit can be found by reading the following registers:
   - ADDR: Address of the word containing the failing bit
   - DATA: contains data to identify which bit failed, and during which phase of the test it failed. The DATA register will in this case contains the following bit groups:

Figure 13-6. DATA bits Description When MBIST Operation Returns an Error



- bit_index: contains the bit number of the failing bit
- phase: indicates which phase of the test failed and the cause of the error, as listed in the following table.

Table 13-4. MBIST Operation Phases

Phase	Test actions
0	Write all bits to zero. This phase cannot fail.
1	Read '0', write '1', increment address
2	Read '1', write '0'
3	Read '0', write '1', decrement address
4	Read '1', write '0', decrement address
5	Read '0', write '1'
6	Read '1', write '0', decrement address
7	Read all zeros. bit_index is not used

Offset	Name	Bit Pos.								
0x2C ... 0x2F	Reserved									
0x30	DEBOUNCEN	7:0	DEBOUNCEN[7:0]							
0x31		15:8	DEBOUNCEN[15:8]							
0x32		23:16	DEBOUNCEN[23:16]							
0x33		31:24	DEBOUNCEN[31:24]							
0x34	DPRESCALER	7:0	STATESx	PRESCALERx[2:0]			STATESx	PRESCALERx[2:0]		
0x35		15:8	STATESx	PRESCALERx[2:0]			STATESx	PRESCALERx[2:0]		
0x36		23:16								TICKON
0x37		31:24								
0x38	PINSTATE	7:0	PINSTATE[7:0]							
0x39		15:8	PINSTATE[15:8]							
0x3A		23:16	PINSTATE[23:16]							
0x3B		31:24	PINSTATE[31:24]							

26.8 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.

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.

26.8.1 Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00
Property: PAC Write-Protection

Bit	7	6	5	4	3	2	1	0
				CKSEL			ENABLE	SWRST
Access				RW			RW	W
Reset				0			0	0

Bit 4 – CKSEL: Clock Selection
The EIC can be clocked either by GCLK_EIC (when a frequency higher than 32KHz is required for filtering) or by CLK_ULP32K (when power consumption is the priority).

This bit is not Write-Synchronized.

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)

Offset	Name	Bit Pos.								
0x2E		23:16	PBLDATA[23:16]							
0x2F		31:24	PBLDATA[31:24]							

27.8 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.

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.

27.8.1 Control A

Name: CTRLA
Offset: 0x00 [ID-00000b2c]
Reset: 0x0000
Property: PAC Write-Protection

Bit	15	14	13	12	11	10	9	8
	CMDEX[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

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

Bits 15:8 – CMDEX[7:0]: Command Execution
When this bit group is written to the key value 0xA5, the command written to CMD will be executed. If a value different from the key value is tried, the write will not be performed and the Programming Error bit in the Status register (STATUS.PROGE) will be set. PROGE is also set if a previously written command is not completed yet.

The key value must be written at the same time as CMD. If a command is issued through the APB bus on the same cycle as an AHB bus access, the AHB bus access will be given priority. The command will then be executed when the NVM block and the AHB bus are idle.

INTFLAG.READY must be '1' when the command is issued.

Bit 0 of the CMDEX bit group will read back as '1' until the command is issued.

Note: The NVM Address bit field in the Address register (ADDR.ADDR) uses 16-bit addressing.

Bits 6:0 – CMD[6:0]: Command
These bits define the command to be executed when the CMDEX key is written.

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

32.8.9 Address

Name: ADDR
Offset: 0x24 [ID-00000e74]
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
	ADDRMASK[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

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

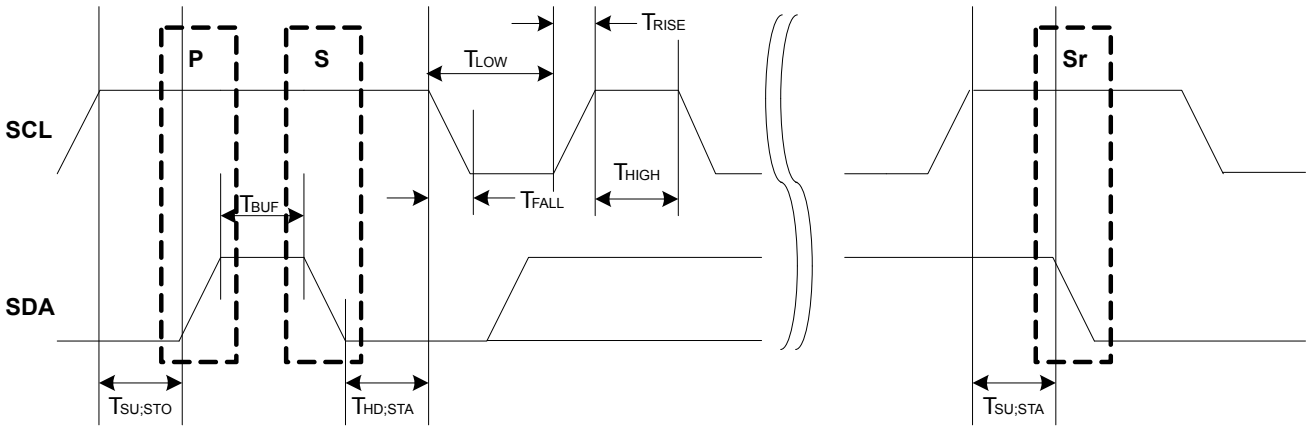
Bits 23:16 – ADDRMASK[7:0]: Address Mask
These bits hold the address mask when the transaction format with address is used (CTRLA.FORM, CTRLB.AMODE).

Bits 7:0 – ADDR[7:0]: Address
These bits hold the address when the transaction format with address is used (CTRLA.FORM, CTRLB.AMODE).

32.8.10 Data

Name: DATA
Offset: 0x28 [ID-00000e74]
Reset: 0x0000
Property: –

Figure 33-7. SCL Timing



The following parameters are timed using the SCL low time period T_{LOW}. This comes from the Master Baud Rate Low bit group in the Baud Rate register (BAUD.BAUDLOW). When BAUD.BAUDLOW=0, or the Master Baud Rate bit group in the Baud Rate register (BAUD.BAUD) determines it.

- T_{LOW} – Low period of SCL clock
- $T_{SU;STO}$ – Set-up time for stop condition
- T_{BUF} – Bus free time between stop and start conditions
- $T_{HD;STA}$ – Hold time (repeated) start condition
- $T_{SU;STA}$ – Set-up time for repeated start condition
- T_{HIGH} is timed using the SCL high time count from BAUD.BAUD
- T_{RISE} is determined by the bus impedance; for internal pull-ups. Refer to *Electrical Characteristics*.
- T_{FALL} is determined by the open-drain current limit and bus impedance; can typically be regarded as zero. Refer to *Electrical Characteristics* for details.

The SCL frequency is given by:

$$f_{\mathrm{SCL}} = \frac{1}{T_{\mathrm{LOW}} + T_{\mathrm{HIGH}} + T_{\mathrm{RISE}}}$$

When BAUD.BAUDLOW is zero, the BAUD.BAUD value is used to time both SCL high and SCL low. In this case the following formula will give the SCL frequency:

$$f_{\mathrm{SCL}} = \frac{f_{\mathrm{GCLK}}}{10 + 2BAUD + f_{\mathrm{GCLK}} \cdot T_{\mathrm{RISE}}}$$

When BAUD.BAUDLOW is non-zero, the following formula determines the SCL frequency:

$$f_{\mathrm{SCL}} = \frac{f_{\mathrm{GCLK}}}{10 + BAUD + BAUDLOW + f_{\mathrm{GCLK}} \cdot T_{\mathrm{RISE}}}$$

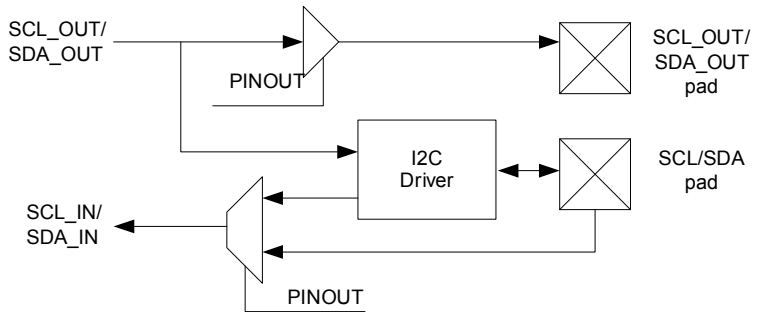
The following formulas can determine the SCL T_{LOW} and T_{HIGH} times:

$$T_{\mathrm{LOW}} = \frac{BAUDLOW + 5}{f_{\mathrm{GCLK}}}$$

$$T_{\mathrm{HIGH}} = \frac{BAUD + 5}{f_{\mathrm{GCLK}}}$$

Note: The I²C 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.

Figure 33-14. I²C Pad Interface



33.6.3.4 Quick Command

Setting the Quick Command Enable bit in the Control B register (CTRLB.QCEN) enables quick command. When quick command is enabled, the corresponding interrupt flag (INTFLAG.SB or INTFLAG.MB) is set immediately after the slave acknowledges the address. At this point, the software can either issue a stop command or a repeated start by writing CTRLB.CMD or ADDR.ADDR.

33.6.4 DMA, Interrupts and Events

Each interrupt source has its own interrupt flag. The interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG) will be set when the interrupt condition is meet. Each interrupt can be individually enabled by writing '1' to the corresponding bit in the Interrupt Enable Set register (INTENSET), and disabled by writing '1' to the corresponding bit in the Interrupt Enable Clear register (INTENCLR). An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request active until the interrupt flag is cleared, the interrupt is disabled or the I²C is reset. See INTFLAG register for details on how to clear interrupt flags.

Table 33-1. Module Request for SERCOM I²C Slave

Condition	Request		
	DMA	Interrupt	Event
Data needed for transmit (TX) (Slave transmit mode)	Yes (request cleared when data is written)		NA
Data received (RX) (Slave receive mode)	Yes (request cleared when data is read)		
Data Ready (DRDY)		Yes	
Address Match (AMATCH)		Yes	
Stop received (PREC)		Yes	
Error (ERROR)		Yes	

Table 34-7. Tx Buffer / FIFO / Queue Element Size

TXESC.TBDS[2:0]	Data Field [bytes]	Element Size [RAM words]
000	8	4
001	12	5
010	16	6
011	20	7
100	24	8
101	32	10
110	48	14
111	64	18

34.6.6.3 Tx FIFO

Tx FIFO operation is configured by programming TXBC.TFQM to '0'. Messages stored in the Tx FIFO are transmitted starting with the message referenced by the Get Index TXFQS.TFGI. After each transmission the Get Index is incremented cyclically until the Tx FIFO is empty. The Tx FIFO enables transmission of messages with the same Message ID from different Tx Buffers in the order these messages have been written to the Tx FIFO. The CAN calculates the Tx FIFO Free Level TXFQS.TFFL as difference between Get and Put Index. It indicates the number of available (free) Tx FIFO elements.

New transmit messages have to be written to the Tx FIFO starting with the Tx Buffer referenced by the Put Index TXFQS.TFQPI. An Add Request increments the Put Index to the next free Tx FIFO element. When the Put Index reaches the Get Index, Tx FIFO Full (TXFQS.TFQF = '1') is signaled. In this case no further messages should be written to the Tx FIFO until the next message has been transmitted and the Get Index has been incremented.

When a single message is added to the Tx FIFO, the transmission is requested by writing a '1' to the TXBAR bit related to the Tx Buffer referenced by the Tx FIFO's Put Index.

When multiple (n) messages are added to the Tx FIFO, they are written to n consecutive Tx Buffers starting with the Put Index. The transmissions are then requested via TXBAR. The Put Index is then cyclically incremented by n. The number of requested Tx buffers should not exceed the number of free Tx Buffers as indicated by the Tx FIFO Free Level.

When a transmission request for the Tx Buffer referenced by the Get Index is canceled, the Get Index is incremented to the next Tx Buffer with pending transmission request and the Tx FIFO Free Level is recalculated. When transmission cancellation is applied to any other Tx Buffer, the Get Index and the FIFO Free Level remain unchanged.

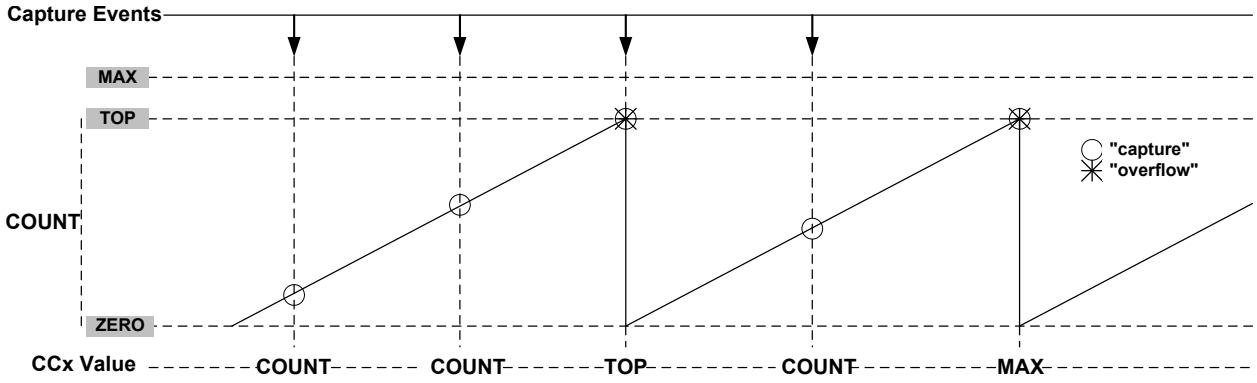
A Tx FIFO element allocates Element Size 32-bit words in the Message RAM (refer to Table 34-7). Therefore the start address of the next available (free) Tx FIFO Buffer is calculated by adding Tx FIFO/ Queue Put Index TXFQS.TFQPI (0…31) • Element Size to the Tx Buffer Start Address TXBC.TBSA.

34.6.6.4 Tx Queue

Tx Queue operation is configured by programming TXBC.TFQM to '1'. Messages stored in the Tx Queue are transmitted starting with the message with the lowest Message ID (highest priority). In case that multiple Queue Buffers are configured with the same Message ID, the Queue Buffer with the lowest buffer number is transmitted first.

New messages have to be written to the Tx Buffer referenced by the Put Index TXFQS.TFQPI. An Add Request cyclically increments the Put Index to the next free Tx Buffer. In case that the Tx Queue is full

Figure 35-15. Time-Stamp



35.6.3.3 Minimum Capture

The minimum capture is enabled by writing the CAPTMIN mode in the Channel n Capture Mode bits in the Control A register (CTRLA.CAPTMODEn = CAPTMIN).

CCx Content:

In CAPTMIN operations, CCx keeps the Minimum captured values. Before enabling this mode of capture, the user must initialize the corresponding CCx register value to a value different from zero. If the CCx register initial value is zero, no captures will be performed using the corresponding channel.

MCx Behaviour:

In CAPTMIN operation, capture is performed only when on capture event time, the counter value is lower than the last captured value. The MCx interrupt flag is set only when on capture event time, the counter value is upper or equal to the value captured on the previous event. So interrupt flag is set when a new absolute local Minimum value has been detected.

35.6.3.4 Maximum Capture

The maximum capture is enabled by writing the CAPTMAX mode in the Channel n Capture Mode bits in the Control A register (CTRLA.CAPTMODEn = CAPTMAX).

CCx Content:

In CAPTMAX operations, CCx keeps the Maximum captured values. Before enabling this mode of capture, the user must initialize the corresponding CCx register value to a value different from TOP. If the CCx register initial value is TOP, no captures will be performed using the corresponding channel.

MCx Behaviour:

In CAPTMAX operation, capture is performed only when on capture event time, the counter value is upper than the last captured value. The MCx interrupt flag is set only when on capture event time, the counter value is lower or equal to the value captured on the previous event. So interrupt flag is set when a new absolute local Maximum value has been detected.

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

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

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

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

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

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

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

35.7.1.6 Interrupt Enable Set

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

Name: INTENSET
Offset: 0x09
Reset: 0x00
Property: PAC Write-Protection

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

Bit 4 – MCx: Match or Capture Channel x Interrupt Enable
Writing a '0' to these bits has no effect.

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

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

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

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

The Channel x Compare/Capture Buffer Value (CCBUFx) registers provide double buffer capability. The double buffering synchronizes the update of the CCx register with the buffer value at the UPDATE condition or a force update command (CTRLBSET.CMD=0x3, UPDATE). For further details, refer to Double Buffering. The synchronization prevents the occurrence of odd-length, non-symmetrical pulses and ensures glitch-free output.

Waveform Output Generation Operations

The compare channels can be used for waveform generation on output port pins. To make the waveform available on the connected pin, the following requirements must be fulfilled:

1. Choose a waveform generation mode in the Waveform Generation Operation bit in Waveform register (WAVE.WAVEGEN).
2. Optionally invert the waveform output WO[x] by writing the corresponding Waveform Output x Inversion bit in the Driver Control register (DRVCTRL.INVENx).
3. Configure the pins with the I/O Pin Controller. Refer to *PORT - I/O Pin Controller* for details.

The counter value is continuously compared with each CCx value. On a comparison match, the Match or Capture Channel x bit in the Interrupt Flag Status and Clear register (INTFLAG.MCx) will be set on the next zero-to-one transition of CLK_TCC_COUNT (see Normal Frequency Operation). An interrupt and/or event can be generated on the same condition if Match/Capture occurs, i.e. INTENSET.MCx and/or EVCTRL.MCEOx is '1'. Both interrupt and event can be generated simultaneously. The same condition generates a DMA request.

There are seven waveform configurations for the Waveform Generation Operation bit group in the Waveform register (WAVE.WAVEGEN). This will influence how the waveform is generated and impose restrictions on the top value. The configurations are:

- Normal Frequency (NFRQ)
- Match Frequency (MFRQ)
- Normal Pulse-Width Modulation (NPWM)
- Dual-slope, interrupt/event at TOP (DSTOP)
- Dual-slope, interrupt/event at ZERO (DSBOTTOM)
- Dual-slope, interrupt/event at Top and ZERO (DSBOTH)
- Dual-slope, critical interrupt/event at ZERO (DSCRITICAL)

When using MFRQ configuration, the TOP value is defined by the CC0 register value. For the other waveform operations, the TOP value is defined by the Period (PER) register value.

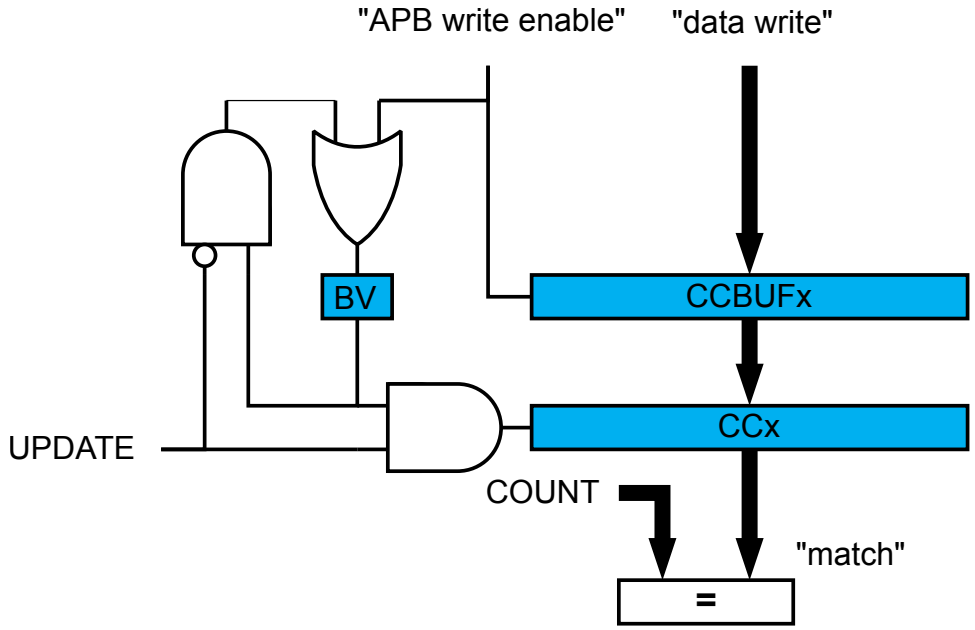
For dual-slope waveform operations, the update time occurs when the counter reaches ZERO. For the other waveforms generation modes, the update time occurs on counter wraparound, on overflow, underflow, or re-trigger.

The table below shows the update counter and overflow event/interrupt generation conditions in different operation modes.

Table 36-2. Counter Update and Overflow Event/interrupt Conditions

Name	Operation	TOP	Update	Output Waveform		OVFIF/Event	
				On Match	On Update	Up	Down
NFRQ	Normal Frequency	PER	TOP/ ZERO	Toggle	Stable	TOP	ZERO
MFRQ	Match Frequency	CC0	TOP/ ZERO	Toggle	Stable	TOP	ZERO

Figure 36-9. Compare Channel Double Buffering



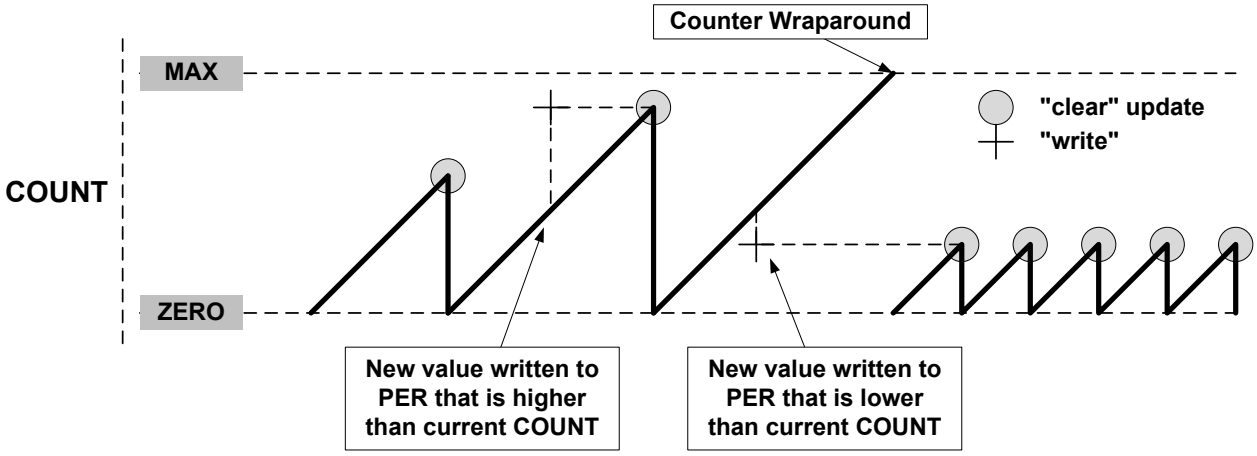
Both the registers (PATT/WAVE/PER/CCx) and corresponding buffer registers (PATTBUF/WAVEBUFV/PERBUF/CCBUFx) are available in the I/O register map, and the double buffering feature is not mandatory. The double buffering is disabled by writing a '1' to CTRLSET.LUPD.

Note: In NFRQ, MFRQ or PWM down-counting counter mode (CTRLBSET.DIR=1), when double buffering is enabled (CTRLBCLR.LUPD=1), PERBUF register is continuously copied into the PER independently of update conditions.

Changing the Period

The counter period can be changed by writing a new Top value to the Period register (PER or CC0, depending on the waveform generation mode), any period update on registers (PER or CCx) is effective after the synchronization delay, whatever double buffering enabling is.

Figure 36-10. Unbuffered Single-Slope Up-Counting Operation



Number of Accumulated Samples	AVGCTRL. SAMPLENUM	Number of Automatic Right Shifts	Final Result Precision	Automatic Division Factor
4	0x2	0	14 bits	0
8	0x3	0	15 bits	0
16	0x4	0	16 bits	0
32	0x5	1	16 bits	2
64	0x6	2	16 bits	4
128	0x7	3	16 bits	8
256	0x8	4	16 bits	16
512	0x9	5	16 bits	32
1024	0xA	6	16 bits	64
Reserved	0xB –0xF		12 bits	0

38.6.2.10 Averaging

Averaging is a feature that increases the sample accuracy, at the cost of a reduced sampling rate. This feature is suitable when operating in noisy conditions.

Averaging is done by accumulating m samples, as described in Accumulation, and dividing the result by m. The averaged result is available in the RESULT register. The number of samples to be accumulated is specified by writing to AVGCTRL.SAMPLENUM as shown in Table 38-2.

The division is obtained by a combination of the automatic right shift described above, and an additional right shift that must be specified by writing to the Adjusting Result/Division Coefficient field in AVGCTRL (AVGCTRL.ADJRES), as described in Table 38-2.

Note: To perform the averaging of two or more samples, the Conversion Result Resolution field in the Control C register (CTRLC.RESSEL) must be set.

Averaging AVGCTRL.SAMPLENUM samples will reduce the un-averaged sampling rate by a factor $\frac{1}{\text{AVGCTRL.SAMPLENUM}}$.

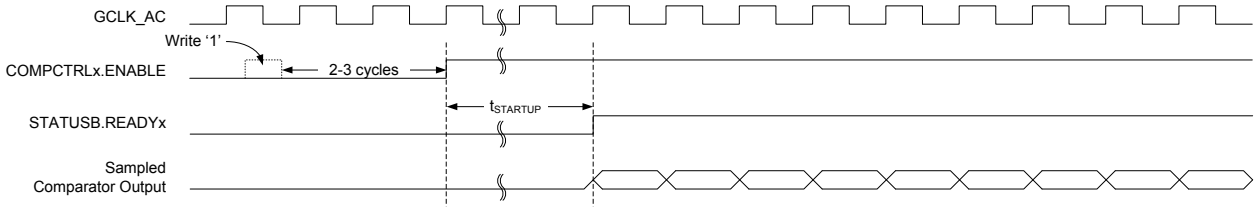
When the averaged result is available, the INTFLAG.RESRDY bit will be set.

Table 38-2. Averaging

Number of Accumulated Samples	AVGCTRL. SAMPLENUM	Intermediate Result Precision	Number of Automatic Right Shifts	Division Factor	AVGCTRL.ADJRES	Total Number of Right Shifts	Final Result Precision	Automatic Division Factor
1	0x0	12 bits	0	1	0x0		12 bits	0
2	0x1	13	0	2	0x1	1	12 bits	0
4	0x2	14	0	4	0x2	2	12 bits	0
8	0x3	15	0	8	0x3	3	12 bits	0
16	0x4	16	0	16	0x4	4	12 bits	0
32	0x5	17	1	16	0x4	5	12 bits	2

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

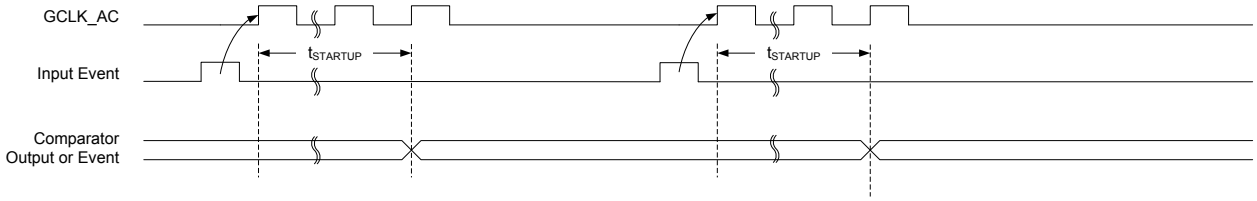
Figure 40-10. Continuous Mode SleepWalking



40.6.14.2 Single-Shot Measurement during Sleep

For low-power operation, event-triggered measurements can be performed during sleep modes. When the event occurs, the Power Manager will start GCLK_AC. The comparator is enabled, and after the start-up time has passed, a comparison is done, with filtering if desired, and the appropriate peripheral events and interrupts are also generated, as shown in Figure 40-11. The comparator and GCLK_AC are then disabled again automatically, unless configured to wake the system from sleep. Filtering is allowed with this configuration.

Figure 40-11. Single-Shot SleepWalking



40.6.15 Synchronization

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

The following bits are synchronized when written:

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

The following registers are synchronized when written:

- Window Control register (WINCTRL)

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

Related Links

Register Synchronization

43.5.8 Register Access Protection

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

- Control B (CTRLB) register
- Interrupt Flag Status and Clear (INTFLAG) register

Write-protection is denoted by the PAC Write-Protection property in the register description.

Write-protection does not apply to accesses through an external debugger. Refer to the Peripheral Access Controller chapter for details.

43.5.9 Calibration

The GAIN, OFFSET, FCAL, and TCAL calibration values from the production test must be loaded from the NVM Temperature Calibration Area into the TSENS Gain register (GAIN), Offset register (OFFSET) and Calibration register (CAL) by software to achieve specified accuracy.

Related Links

NVM Software Calibration Area Mapping

Temperature Sensor Characteristics

43.6 Functional Description

43.6.1 Principle of Operation

The TSENS accurately measures the operating temperature of the device by comparing the difference in two temperature dependent frequencies to a known frequency. The frequency of the temperature dependent oscillator (TOSC) is measured twice: first with the min configuration and next with the max configuration. The number of periods of GCLK_TSENS used for the measurement is defined by the GAIN register. The width of the resulting pulse is measured using a counter clocked by GCLK_TSENS in the up direction for the 1st phase and in the down 2nd phase.

The resulting signed value is proportional to the temperature and is corrected for offset by the contents of the OFFSET register.

$$\mathrm{VALUE} = \mathrm{OFFSET} + \mathrm{GAIN} \times \left(\frac{f_{\mathrm{TOSCMIN}}}{f_{\mathrm{GCLK}}} + - \frac{f_{\mathrm{TOSCMAX}}}{f_{\mathrm{GCLK}}} \right)$$

Note:

- The values of GAIN and OFFSET are factory programmed to give a specific temperature slope when using the undivided internal 48MHz oscillator (OSC48M) as the GCLK_TSENS source. Other frequencies/sources may be used, but the GAIN setting and/or expected slope will need to be scaled accordingly.
- The calibration value should be copied and written into the GAIN and OFFSET registers to get the specified accuracy.

Related Links

VALUE

43.6.2 Basic Operation

43.6.2.1 Initialization

The generic clocks (GCLK_TSENS) should be configured and enabled. Refer to the Generic Clock Controller chapter for details.

2. External Anti-alias filter must be placed in front of each SDADC input to ensure high-frequency signals to not alias into measurement bandwidth. Use capacitors of X5R type for DC measurement. or capacitors of COG or NPO type for AC measurement.

Table 47-9. SDADC DC Performance: Differential Input Mode. Chopper ON(1)

Symbol	Parameters	Conditions (2)	Min	Typ	Max	Unit
INL	Integral Non Linearity	CLK_SDADC = 3MHz VREF = 1.2V	-	+/-2.9	+/-3.9	LSB
		CLK_SDADC = 3MHz INT VREF = 5.5V	-	+/-8.4	+/-9.3	
DNL	Differential Non Linearity	CLK_SDADC = 3MHz VREF = 1.2V	-	+/-1.5	+/-2.1	LSB
		CLK_SDADC = 3MHz INT VREF = 5.5V	-	+/-1.7	+/-2.3	
Eg	Gain Errors	CLK_SDADC = 3MHz VREF = 1.2V	-	+/-0.3	+/-1.9	%
		CLK_SDADC = 3MHz INT VREF = 5.5V	-	+/-0.3	+/-1.7	
TCg	Gain Drift	CLK_SDADC = 3MHz VREF = 1.2V	-0.9	3.9	17.5	ppm/°C
Off	Offset Error	CLK_SDADC = 3MHz VREF = 1.2V	-	+/-2.3	+/-3.7	mV
		CLK_SDADC = 3MHz INT VREF = 5.5V	-	+/-0.3	+/-2.4	
Tco	Offset Error Drift	CLK_SDADC = 3MHz VREF = 1.2V	-1.4	0.01	0.6	uV/°C

1. OSR=256

Table 47-10. SDADC DC Performance: Differential Input Mode. Chopper OFF(1)

Symbol	Parameters	Conditions (2)	Min	Typ	Max	Unit
INL	Integral Non Linearity	CLK_SDADC = 6MHz VREF = 1.2V	-	+/-5.5	+/-9.3	LSB
		CLK_SDADC = 6MHz INT VREF = 5.5V	-	+/-8.9	+/-10.1	
DNL	Differential Non Linearity	CLK_SDADC = 6MHz VREF = 1.2V	-	+/-2.8	+/-4.1	LSB
		CLK_SDADC = 6MHz INT VREF = 5.5V	-	+/-1.8	+/-3	
Eg	Gain Errors	CLK_SDADC = 6MHz VREF = 1.2V	-	+/-0.6	+/-2.1	%
		CLK_SDADC = 6MHz INT VREF = 5.5V	-	+/-0.3	+/-1.7	
TCg	Gain Drift	CLK_SDADC = 6MHz VREF = 1.2V	-19.7	2.2	20.9	ppm/°C
Off	Offset Error	CLK_SDADC = 6MHz VREF = 1.2V	-	+/-1.7	+/-14.3	mV
		CLK_SDADC = 6MHz INT VREF = 5.5V	-	+/-4.9	+/-13.2	
Tco	Offset Error Drift	CLK_SDADC = 6MHz VREF = 1.2V	-14	12.4	60	µV/°C
Input noise rms	AC Input noise rms	OSR = 256 VREF = 1.2V	-	19	20	
		OSR = 256 VREF = 5.5V	-	59	76	mVrms

1. OSR=256

Table 47-11. SDADC AC Performance: : Differential Input Mode(1)

Symbol	Parameters	Conditions (2)	Min	Typ	Max	Unit
ENOB	Effective Number Of Bits	Ext ref = 1.2V	12	15.3	15.4	dB
		Int Ref = 5.5V	12.9	13.1	14	
DR	Dynamic Range	Ext ref = 1.2V	90.5	92.4	93.2	dB
		Int Ref = 5.5V	83.0	95.6	97.0	
SNR	Signal to Noise Ratio	Ext ref = 1.2V	68.7	88.7	89	dB

SERCOM SPI – SERCOM Serial Peripheral Interface	Features: Updated references to serial clock speed in master and slave operation.
CAN - Control Area Network	CREL: Updated reset value from 0x31000000 (device rev B) to 0x32100000 (device rev C and newer).
TC – Timer/Counter	Added register property "Write-Synchronized" to the CCBUFx and PERBUF registers.
TCC – Timer/Counter for Control Applications	• Updated number of TCC instances from one to three. • Counter Operation: 'Stop Command and Event Action' split into 'Stop Command' and 'Pause Event Action' • Capture Operations: Value 0 in CAPTMIN mode is captured only in down-counting mode. • Ramp Operations: RAMP2C Operation added. • Compare Operations: Reorganization of section. • Corrected bit names in the WAVE register: CIRCCENx -> CICCENx and CIRPEREN -> CIPEREN.
CCL – Configurable Custom Logic	• Number of LUTCTRL registers changed from eight to four. • Number of SEQCTRL registers changed from four to two. • Truth Table Inputs Selection: Updated description and figure in Analog Comparator Inputs (AC).
SDADC – Sigma-Delta Analog-to-Digital Converter	• Resolution corrected from 24-bit to 16-bit. • Conversion range updated from "0V to V_{ref} "to "0V to 0.7xV_{ref}" • Test Mode section removed. • Updated operation formula in the following registers: – OFFSETCORR – GAINCORR – SHIFTCORR • Updated RESULT bit description in RESULT.
AC – Analog Comparators	COMPCTRL: SPEED bit description updated. Values 0x1 and 0x2 is reserved.
DAC – Digital-to-Analog Converter	Updated DATA register: DATA bits access corrected from read/write (R/W) to write (W).TPUBSAMD-354
TSENS – Temperature Sensor	Measurement: Added temperature measurement recommendation to avoid discrepancies.
Electrical Characteristics 85°C (SAM C20/C21 E/G/J)	• Added electrical characteristics for 85°C. • GPIO Clusters moved to I/O Multiplexing and Considerations.
Electrical Characteristics 105°C (SAM C20/C21 E/G/J)	• Added electrical characteristics for 105°C.