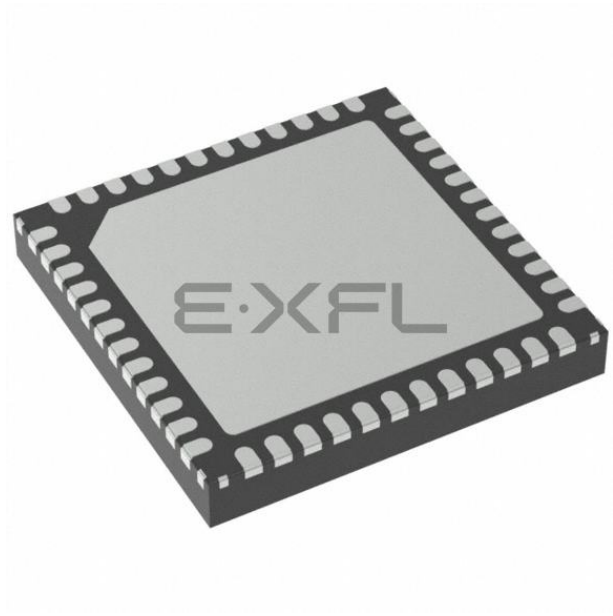


Welcome to E-XFL.COM

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	Obsolete
Core Processor	ARM® Cortex®-M0+
Core Size	32-Bit Single-Core
Speed	48MHz
Connectivity	I²C, LINbus SBC, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, DMA, POR, PWM, WDT
Number of I/O	32
Program Memory Size	16KB (16K x 8)
Program Memory Type	FLASH
EEPROM Size	512 x 8
RAM Size	4K x 8
Voltage - Supply (Vcc/Vdd)	0V ~ 3.8V
Data Converters	A/D 13x12b; D/A 1x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TC)
Mounting Type	Surface Mount, Wettable Flank
Package / Case	48-VFQFN Exposed Pad
Supplier Device Package	48-VQFN (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atsamha1g14a-mbt

15.13.16Peripheral Identification 3

Name: PID3

Offset: 0x1FEC

Reset: 0x00000000

Property: -

Bit	31	30	29	28	27	26	25	24
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

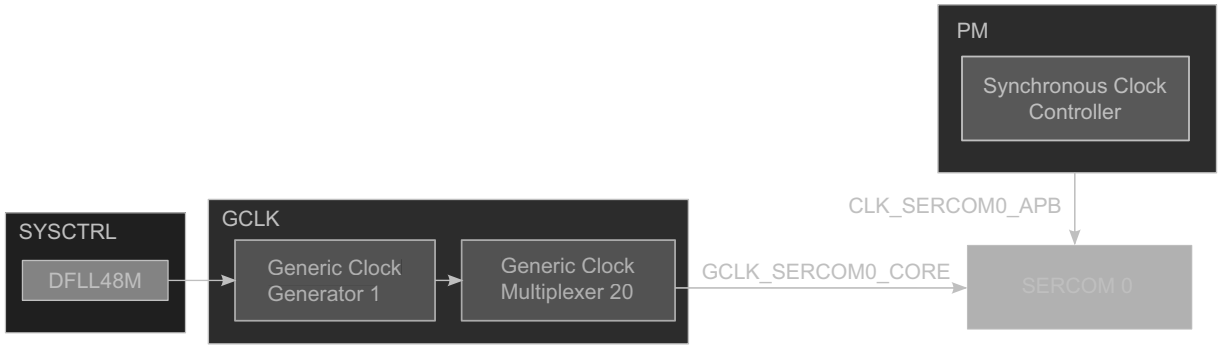
Bit	23	22	21	20	19	18	17	16
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

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

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

- **Bits 31:8 – Reserved**
  These bits are unused and reserved for future use. For compatibility with future devices, always write these bits to zero when this register is written. These bits will always return zero when read.
- **Bits 7:4 – REVAND[3:0]: Revision Number**
  These bits will always return 0x0 when read.
- **Bits 3:0 – CUSMOD[3:0]: ARM CUSMOD**
  These bits will always return 0x0 when read.

Figure 16-2. Example of SERCOM clock



16.2 Synchronous and Asynchronous Clocks

As the CPU and the peripherals can be clocked from different clock sources, possibly with widely different clock speeds, some peripheral accesses by the CPU needs to be synchronized between the different clock domains. In these cases the peripheral includes a SYNCBUSY status flag that can be used to check if a sync operation is in progress. As the nature of the synchronization might vary between different peripherals, detailed description for each peripheral can be found in the sub-chapter “synchronization” for each peripheral where this is necessary.

In the datasheet references to synchronous clocks are referring to the CPU and bus clocks, while asynchronous clocks are clock generated by generic clocks.

16.3 Register Synchronization

There are two different register synchronization schemes implemented on this device: some modules use a common synchronizer register synchronization scheme, while other modules use a distributed synchronizer register synchronization scheme.

The modules using a common synchronizer register synchronization scheme are: GCLK, WDT, RTC, EIC, TC, ADC, AC, DAC.

The modules using a distributed synchronizer register synchronization scheme are: SERCOM USART, SERCOM SPI, SERCOM I2C, TCC.

16.3.1 Common Synchronizer Register Synchronization

16.3.1.1 Overview

All peripherals are composed of one digital bus interface, which is connected to the APB or AHB bus and clocked using a corresponding synchronous clock, and one core clock, which is clocked using a generic clock. Access between these clock domains must be synchronized. As this mechanism is implemented in hardware the synchronization process takes place even if the different clocks domains are clocked from the same source and on the same frequency. All registers in the bus interface are accessible without synchronization. All core registers in the generic clock domain must be synchronized when written. Some core registers must be synchronized when read. Registers that need synchronization has this denoted in each individual register description. Two properties are used: write-synchronization and read-synchronization.

A common synchronizer is used for all registers in one peripheral, as shown in Figure 16-3. Therefore, only one register per peripheral can be synchronized at a time.

Table 17-4. Generic Clock Selection ID (Continued)

Value	Name	Description
0x13	GCLK_SERCOMx_SLOW	SERCOMx_SLOW
0x14	GCLK_SERCOM0_CORE	SERCOM0_CORE
0x15	GCLK_SERCOM1_CORE	SERCOM1_CORE
0x16	GCLK_SERCOM2_CORE	SERCOM2_CORE
0x17	GCLK_SERCOM3_CORE	SERCOM3_CORE
0x18	GCLK_SERCOM4_CORE	SERCOM4_CORE
0x19	GCLK_SERCOM5_CORE	SERCOM5_CORE
0x1A	GCLK_TCC0, GCLK_TCC1	TCC0,TCC1
0x1B	GCLK_TCC2, GCLK_TC3	TCC2,TC3
0x1C	GCLK_TC4, GCLK_TC5	TC4,TC5
0x1D	GCLK_TC6, GCLK_TC7	TC6,TC7
0x1E	GCLK_ADC	ADC
0x1F	GCLK_AC_DIG	AC_DIG
0x20	GCLK_AC_ANA	AC_ANA
0x21	GCLK_DAC	DAC
0x22	GCLK_PTC	PTC
0x23-0x3F	Reserved	

A power reset will reset the CLKCTRL register for all IDs, including the RTC. If the WRTLOCK bit of the corresponding ID is zero and the ID is not the RTC, a user reset will reset the CLKCTRL register for this ID.

After a power reset, the reset value of the CLKCTRL register versus module instance is as shown in Table 17-5.

Table 17-5. CLKCTRL Reset Value after a Power Reset

Module Instance	Reset Value after Power Reset		
	CLKCTRL.GEN	CLKCTRL.CLKEN	CLKCTRL.WRTLOCK
RTC	0x00	0x00	0x00
WDT	0x02	0x01 if WDT Enable bit in NVM User Row written to one 0x00 if WDT Enable bit in NVM User Row written to zero	0x01 if WDT Always-On bit in NVM User Row written to one 0x00 if WDT Always-On bit in NVM User Row written to zero
Others	0x00	0x00	0x00

18.8.5 APBB Clock Select

Name: APBBSEL

Offset: 0x0A

Reset: 0x00

Property: Write-Protected

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

- **Bits 7:3 – Reserved**
  These bits are unused and reserved for future use. For compatibility with future devices, always write these bits to zero when this register is written. These bits will always return zero when read.

- **Bits 2:0 – APBBDIV[2:0]: APBB Prescaler Selection**
  These bits define the division ratio of the APBB clock prescaler (2^n).

Table 18-9. APBB Prescaler Selection

APBBDIV[2:0]	Name	Description
0x0	DIV1	Divide by 1
0x1	DIV2	Divide by 2
0x2	DIV4	Divide by 4
0x3	DIV8	Divide by 8
0x4	DIV16	Divide by 16
0x5	DIV32	Divide by 32
0x6	DIV64	Divide by 64
0x7	DIV128	Divide by 128

Writing a zero to this bit has no effect.

Writing a one to this bit will clear the DPLL Lock Fall Interrupt Enable bit, which disables the DPLL Lock Fall interrupt.

- **Bit 15 – DPLLLCKR: DPLL Lock Rise Interrupt Enable**
  0: The DPLL Lock Rise interrupt is disabled.

  1: The DPLL Lock Rise interrupt is enabled, and an interrupt request will be generated when the DPLL Lock Rise Interrupt flag is set.

  Writing a zero to this bit has no effect.

  Writing a one to this bit will clear the DPLL Lock Rise Interrupt Enable bit, which disables the DPLL Lock Rise interrupt.

- **Bits 14:12 – Reserved**
  These bits are unused and reserved for future use. For compatibility with future devices, always write these bits to zero when this register is written. These bits will always return zero when read.

- **Bit 11 – B33SRDY: BOD33 Synchronization Ready Interrupt Enable**
  0: The BOD33 Synchronization Ready interrupt is disabled.

  1: The BOD33 Synchronization Ready interrupt is enabled, and an interrupt request will be generated when the BOD33 Synchronization Ready Interrupt flag is set.

  Writing a zero to this bit has no effect.

  Writing a one to this bit will clear the BOD33 Synchronization Ready Interrupt Enable bit, which disables the BOD33 Synchronization Ready interrupt.

- **Bit 10 – BOD33DET: BOD33 Detection Interrupt Enable**
  0: The BOD33 Detection interrupt is disabled.

  1: The BOD33 Detection interrupt is enabled, and an interrupt request will be generated when the BOD33 Detection Interrupt flag is set.

  Writing a zero to this bit has no effect.

  Writing a one to this bit will clear the BOD33 Detection Interrupt Enable bit, which disables the BOD33 Detection interrupt.

- **Bit 9 – BOD33RDY: BOD33 Ready Interrupt Enable**
  0: The BOD33 Ready interrupt is disabled.

  1: The BOD33 Ready interrupt is enabled, and an interrupt request will be generated when the BOD33 Ready Interrupt flag is set.

  Writing a zero to this bit has no effect.

  Writing a one to this bit will clear the BOD33 Ready Interrupt Enable bit, which disables the BOD33 Ready interrupt.

- **Bit 8 – DFLLRCS: DFLL Reference Clock Stopped Interrupt Enable**
  0: The DFLL Reference Clock Stopped interrupt is disabled.

  1: The DFLL Reference Clock Stopped interrupt is enabled, and an interrupt request will be generated when the DFLL Reference Clock Stopped Interrupt flag is set.

  Writing a zero to this bit has no effect.

  Writing a one to this bit will clear the DFLL Reference Clock Stopped Interrupt Enable bit, which disables the DFLL Reference Clock Stopped interrupt.

- **Bit 7 – DFLLLCKC: DFLL Lock Coarse Interrupt Enable**
  0: The DFLL Lock Coarse interrupt is disabled.

  1: The DFLL Lock Coarse interrupt is enabled, and an interrupt request will be generated when the DFLL Lock Coarse Interrupt flag is set.

1: The XOSC Ready interrupt is enabled, and an interrupt request will be generated when the XOSC Ready Interrupt flag is set.

Writing a zero to this bit has no effect.

Writing a one to this bit will set the XOSC Ready Interrupt Enable bit, which enables the XOSC Ready interrupt.

- **Bit 3 – EN32K: 32kHz Output Enable**
  0: The 32kHz output is disabled.
  1: The 32kHz output is enabled.
- **Bit 2 – XTALEN: Crystal Oscillator Enable**
  This bit controls the connections between the I/O pads and the external clock or crystal oscillator:
  0: External clock connected on XIN32. XOUT32 can be used as general-purpose I/O.
  1: Crystal connected to XIN32/XOUT32.
- **Bit 1 – ENABLE: Oscillator Enable**
  0: The oscillator is disabled.
  1: The oscillator is enabled.
- **Bit 0 – Reserved**
  This bit is unused and reserved for future use. For compatibility with future devices, always write this bit to zero when this register is written. This bit will always return zero when read.

22.8.1.19 Channel Control B

Name: CHCTRLB

Offset: 0x44

Reset: 0x00000000

Property: Enable-Protected, Write-Protected

Bit	31	30	29	28	27	26	25	24
							CMD[1:0]	
Access	R	R	R	R	R	R	R/W	R/W
Reset	0	0	0	0	0	0	0	0

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

Bit	15	14	13	12	11	10	9	8
			TRIGSRC[5:0]					
Access	R	R	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
		LVL[1:0]		EVOE	EVIE	EVACT[2:0]		
Access	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

- **Bits 31:26 – Reserved**
  These bits are unused and reserved for future use. For compatibility with future devices, always write these bits to zero when this register is written. These bits will always return zero when read.
- **Bits 25:24 – CMD[1:0]: Software Command**
  These bits define the software commands, as shown in Table 22-6.
  These bits are not enable-protected.

Table 22-6. Software Command

CMD[1:0]	Name	Description
0x0	NOACT	No action
0x1	SUSPEND	Channel suspend operation
0x2	RESUME	Channel resume operation
0x3		Reserved

24.8.5 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: 0x10

Reset: 0x00

Property: Write-Protected

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

- **Bits 7:2 – Reserved**
  These bits are unused and reserved for future use. For compatibility with future devices, always write these bits to zero when this register is written. These bits will always return zero when read.

- **Bit 1 – ERROR: Error Interrupt Enable**
  Writing a zero to this bit has no effect.
  Writing a one 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 zero to this bit has no effect.
  Writing a one to this bit sets the READY interrupt enable.
  This bit will read as the current value of the READY interrupt enable.

Writing a zero to this bit has no effect.

Writing a one to this bit will clear the Overrun Channel x interrupt flag.

- **Bits 15:8 – EVDx [x=7..0]: Channel x Event Detection**
  This flag is set on the next CLK_EVSYS_APB cycle when an event is being propagated through the channel, and an interrupt request will be generated if INTENCLR/SET.EVDx is one.

  When the event channel path is asynchronous, the EVDx interrupt flag will not be set.

  Writing a zero to this bit has no effect.

  Writing a one to this bit will clear the Event Detected Channel n interrupt flag.

- **Bits 7:0 – OVRx [x=7..0]: Channel x Overrun**
  This flag is set on the next CLK_EVSYS cycle after an overrun channel condition occurs, and an interrupt request will be generated if INTENCLR/SET.OVRx is one.

  When the event channel path is asynchronous, the OVRx interrupt flag will not be set.

  Writing a zero to this bit has no effect.

  Writing a one to this bit will clear the Overrun Channel x interrupt flag.

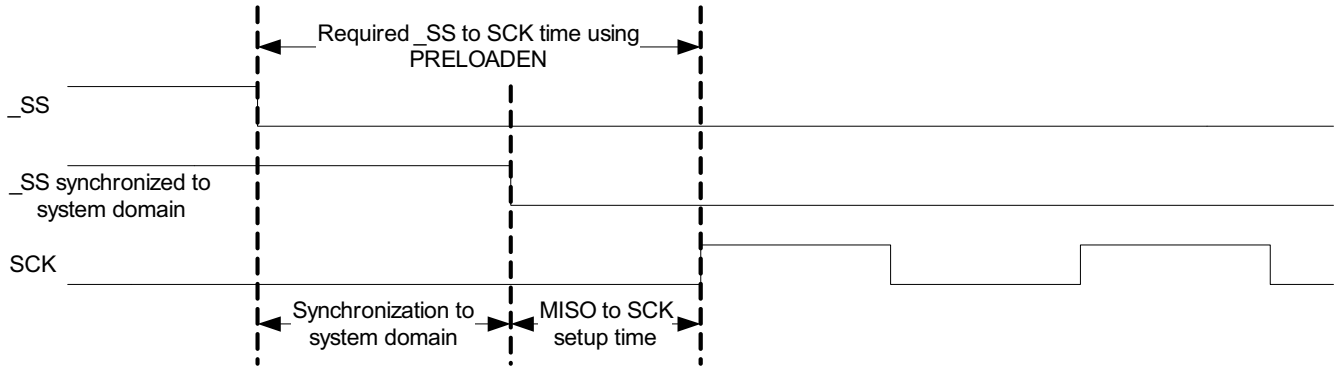
29.6.3.2 Preloading of the Slave Shift Register

When starting a transaction, the slave will first transmit the contents of the shift register before loading new data from DATA. The first character sent can be either the reset value of the shift register (if this is the first transmission since the last reset) or the last character in the previous transmission. Preloading can be used to preload data to the shift register while _SS is high and eliminate sending a dummy character when starting a transaction.

In order to guarantee enough set-up time before the first SCK edge, enough time must be given between _SS going low and the first SCK sampling edge, as shown in Figure 29-4.

Preloading is enabled by setting the Slave Data Preload Enable bit in the Control B register (CTRLB.PLOADEN).

Figure 29-4. Timing Using Preloading

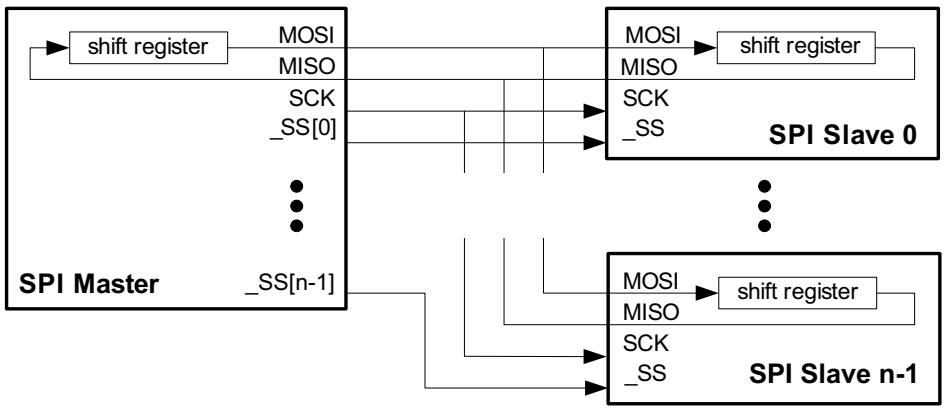


Only one data character written to DATA will be preloaded into the shift register while the synchronized _SS signal (see Figure 29-4) is high. The next character written to DATA before _SS is pulled low will be stored in DATA until transfer begins. If the shift register is not preloaded, the current contents of the shift register will be shifted out.

29.6.3.3 Master with Several Slaves

Master with multiple slaves in parallel feature is available only when Master Slave Select Enable (CTRLB.MSSEN) is set to zero and hardware _SS control is disabled. If the bus consists of several SPI slaves, an SPI master can use general purpose I/O pins to control the _SS line to each of the slaves on the bus, as shown in Figure 29-5. In this configuration, the single selected SPI slave will drive the tri-state MISO line.

Figure 29-5. Multiple Slaves in Parallel



An alternate configuration is shown in Figure 29-6. In this configuration, all n attached slaves are connected in series. A common _SS line is provided to all slaves, enabling them simultaneously. The master must shift n characters for a complete transaction. Depending on the Master Slave Select Enable bit (CTRLB.MSSEN), _SS line is controlled either by hardware or by user software and normal GPIO

1: SCL stretch only after ACK bit according to Figure 30-10.

This bit is not synchronized.

- **Bit 26– Reserved**
  This bit is unused and reserved for future use. For compatibility with future devices, always write this bit to zero when this register is written. This bit will always return zero when read.
- **Bits 25:24 – SPEED[1:0]: Transfer Speed**
  These bits define bus speed.

Table 30-5. Transfer Speed

Value	Description
0x0	Standard-mode (Sm) up to 100 kHz and Fast-mode (Fm) up to 400 kHz
0x1	Fast-mode Plus (Fm+) up to 1 MHz
0x2	High-speed mode (Hs-mode) up to 3.4 MHz
0x3	Reserved

These bits are not synchronized.

- **Bit 23 – SEXTTOEN: Slave SCL Low Extend Time-Out**
  This bit enables the slave SCL low extend time-out. If SCL is cumulatively held low for greater than 25ms from the initial START to a STOP, the slave will release its clock hold if enabled and reset the internal state machine. Any interrupts set at the time of time-out will remain set. If the address was recognized, PREC will be set when a STOP is received.

  0: Time-out disabled

  1: Time-out enabled

  This bit is not synchronized.
- **Bit 22 – Reserved**
  This bit is unused and reserved for future use. For compatibility with future devices, always write this bit to zero when this register is written. This bit will always return zero when read.
- **Bits 21:20 – SDAHOLD[1:0]: SDA Hold Time**
  These bits define the SDA hold time with respect to the negative edge of SCL.

Table 30-6. SDA Hold Time

Value	Name	Description
0x0	DIS	Disabled
0x1	75	50-100ns hold time
0x2	450	300-600ns hold time
0x3	600	400-800ns hold time

These bits are not synchronized.

- **Bits 19:17 – Reserved**
  These bits are unused and reserved for future use. For compatibility with future devices, always write these bits to zero when this register is written. These bits will always return zero when read.

- **Bit 9 – GCMD: PMBus Group Command**
  This bit enables PMBus group command support. When enabled, a STOP interrupt will be generated if the slave has been addressed since the last STOP condition on the bus.

  0: Group command is disabled.

  1: Group command is enabled.

  This bit is not write-synchronized.

- **Bit 8 – SMEN: Smart Mode Enable**
  This bit enables smart mode. When smart mode is enabled, acknowledge action is sent when DATA.DATA is read.

  0: Smart mode is disabled.

  1: Smart mode is enabled.

  This bit is not write-synchronized.

- **Bits 7:0 – Reserved**
  These bits are unused and reserved for future use. For compatibility with future devices, always write these bits to zero when this register is written. These bits will always return zero when read.

Table 31-6. Prescaler

Value	Name	Description
0x0	DIV1	Prescaler: GCLK_TC
0x1	DIV2	Prescaler: GCLK_TC/2
0x2	DIV4	Prescaler: GCLK_TC/4
0x3	DIV8	Prescaler: GCLK_TC/8
0x4	DIV16	Prescaler: GCLK_TC/16
0x5	DIV64	Prescaler: GCLK_TC/64
0x6	DIV256	Prescaler: GCLK_TC/256
0x7	DIV1024	Prescaler: GCLK_TC/1024

- **Bit 7 – Reserved**
  This bit is unused and reserved for future use. For compatibility with future devices, always write this bit to zero when this register is written. This bit will always return zero when read.
- **Bits 6:5 – WAVEGEN[1:0]: Waveform Generation Operation**
  These bits select the waveform generation operation. They affect the top value, as shown in “Waveform Output Operations” on page 582. It also controls whether frequency or PWM waveform generation should be used. How these modes differ can also be seen from “Waveform Output Operations” on page 582.
  These bits are not synchronized.

Table 31-7. Waveform Generation Operation

Value	Name	Operation	Top Value	Waveform Output on Match	Waveform Output on Wraparound
0x0	NFRQ	Normal frequency	PER(1)/Max	Toggle	No action
0x1	MFRQ	Match frequency	CC0	Toggle	No action
0x2	NPWM	Normal PWM	PER(1)/Max	Clear when counting up Set when counting down	Set when counting up Clear when counting down
0x3	MPWM	Match PWM	CC0	Clear when counting up Set when counting down	Set when counting up Clear when counting down

Note: 1. This depends on the TC mode. In 8-bit mode, the top value is the Period Value register (PER). In 16- and 32-bit mode it is the maximum value.

- **Bit 4 – Reserved**
  This bit is unused and reserved for future use. For compatibility with future devices, always write this bit to zero when this register is written. This bit will always return zero when read.
- **Bits 3:2 – MODE[1:0]: TC Mode**
  These bits select the TC mode, as shown in Table 31-8.
  These bits are not synchronized.

32.8.9 Debug Control

Name: DBGCTRL

Offset: 0x1E

Reset: 0x00

Property: Write-Protected

Bit	7	6	5	4	3	2	1	0
						FDDBD		DBGRUN
Access	R	R	R	R	R	R/W	R	R/W
Reset	0	0	0	0	0	0	0	0

- **Bits 7:3 – Reserved**
  These bits are unused and reserved for future use. For compatibility with future devices, always write these bits to zero when this register is written. These bits will always return zero when read.

- **Bit 2 – FDDBD: Fault Detection on Debug Break Detection**
  This bit is not affected by software reset and should not be changed by software while the TCC is enabled.
  By default this bit is zero, the on-chip debug (OCD) fault protection is enabled. OCD break request from the OCD system will trigger non-recoverable fault. When this bit is set, OCD fault protection is disabled and OCD break request will not trigger a fault.
  0: No faults are generated when TCC is halted in debug mode.
  1: A non recoverable fault is generated and DFS flag is set when TCC is halted in debug mode.

- **Bit 1 – Reserved**
  This bit is unused and reserved for future use. For compatibility with future devices, always write this bit to zero when this register is written. This bit will always return zero when read.

- **Bit 0 – DBGRUN: Debug Running Mode**
  This bit is not affected by software reset and should not be changed by software while the TCC is enabled.
  0: The TCC is halted when the device is halted in debug mode.
  1: The TCC continues normal operation when the device is halted in debug mode.

Mode: DITH6

Name: COUNT

Offset: 0x34

Reset: 0x00000000

Property: Read-Synchronized, Write-Protected, Write-Synchronized

Bit	31	30	29	28	27	26	25	24
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	COUNT[17:10]							
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
	COUNT[9:2]							
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
	COUNT[1:0]							
Access	R/W	R/W	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

- **Bits 31:24 – Reserved**
  These bits are unused and reserved for future use. For compatibility with future devices, always write these bits to zero when this register is written. These bits will always return zero when read.
- **Bits 23:6 – COUNT[17:0]: Counter Value**
  These bits hold the value of the counter register.
  The number of bits in this field corresponds to the size of the counter.
- **Bits 5:0 – Reserved**
  These bits are unused and reserved for future use. For compatibility with future devices, always write these bits to zero when this register is written. These bits will always return zero when read.

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
512	0x9	21	5	16	0x4	9	12 bits	32
1024	0xA	22	6	16	0x4	10	12 bits	64
Reserved	0xB –0xF				0x0		12 bits	0

33.6.8 Oversampling and Decimation

By using oversampling and decimation, the ADC resolution can be increased from 12 bits to up to 16 bits. To increase the resolution by *n* bits, 4^n samples must be accumulated. The result must then be shifted right by *n* bits. This right shift is a combination of the automatic right shift and the value written to AVGCTRL.ADJRES. To obtain the correct resolution, the ADJRES must be configured as described in the table below. This method will result in *n* bit extra LSB resolution.

Table 33-4. Configuration Required for Oversampling and Decimation

Result Resolution	Number of Samples to Average	AVGCTRL.SAMPLENUM[3:0]	Number of Automatic Right Shifts	AVGCTRL.ADJRES[2:0]
13 bits	$4^1 = 4$	0x2	0	0x1
14 bits	$4^2 = 16$	0x4	0	0x2
15 bits	$4^3 = 64$	0x6	2	0x1
16 bits	$4^4 = 256$	0x8	4	0x0

33.6.9 Window Monitor

The window monitor allows the conversion result to be compared to some predefined threshold values. Supported modes are selected by writing the Window Monitor Mode bit group in the Window Monitor Control register (WINCTRL.WINMODE[2:0]). Thresholds are given by writing the Window Monitor Lower Threshold register (WINLT) and Window Monitor Upper Threshold register (WINUT).

If differential input is selected, the WINLT and WINUT are evaluated as signed values. Otherwise they are evaluated as unsigned values.

Another important point is that the significant WINLT and WINUT bits are given by the precision selected in the Conversion Result Resolution bit group in the Control B register (CTRLB.RESSEL). This means that if 8-bit mode is selected, only the eight lower bits will be considered. In addition, in differential mode, the eighth bit will be considered as the sign bit even if the ninth bit is zero.

The INTFLAG.WINMON interrupt flag will be set if the conversion result matches the window monitor condition.

33.6.10 Offset and Gain Correction

Inherent gain and offset errors affect the absolute accuracy of the ADC. The offset error is defined as the deviation of the actual ADC's transfer function from an ideal straight line at zero input voltage. The offset error cancellation is handled by the Offset Correction register (OFFSETCORR). The offset correction value is subtracted from the converted data before writing the Result register (RESULT). The gain error is defined as the deviation of the last output step's midpoint from the ideal straight line, after compensating for offset error. The gain error cancellation is handled by the Gain Correction register (GAINCORR). To correct these two errors, the Digital Correction Logic Enabled bit in the Control B register (CTRLB.CORREN) must be written to one.

Offset and gain error compensation results are both calculated according to:

34.5.10 Other Dependencies

Not applicable.

34.6 Functional Description

34.6.1 Principle of Operation

Each comparator has one positive input and one negative input. Each positive input may be chosen from a selection of analog input pins. Each negative input may be chosen from a selection of analog input pins or internal inputs, such as a bandgap reference voltage. The digital output from the comparator is one when the difference between the positive and the negative input voltage is positive, and zero otherwise.

The individual comparators can be used independently (normal mode) or grouped in pairs to generate a window comparison (window mode).

34.6.2 Basic Operation

34.6.2.1 Initialization

Before enabling the AC, the input and output events must be configured in the Event Control register (EVCTRL). These settings cannot be changed while the AC is enabled.

Each individual comparator must also be configured by its respective Comparator Control register (COMPCTRL0) before that comparator is enabled. These settings cannot be changed while the comparator is enabled.

- Select the desired measurement mode with COMPCTRLx.SINGLE. See “Starting a Comparison” on page 753 for more details
- Select the desired hysteresis with COMPCTRLx.HYST. See “Input Hysteresis” on page 757 for more details
- Select the comparator speed versus power with COMPCTRLx.SPEED. See “Propagation Delay vs. Power Consumption” on page 757 for more details
- Select the interrupt source with COMPCTRLx.INTSEL
- Select the positive and negative input sources with the COMPCTRLx.MUXPOS and COMPCTRLx.MUXNEG bits. See section “Selecting Comparator Inputs” on page 755 for more details
- Select the filtering option with COMPCTRLx.FLEN

34.6.2.2 Enabling, Disabling and Resetting

The AC is enabled by writing a one to the Enable bit in the Control A register (CTRLA.ENABLE). The individual comparators must be also enabled by writing a one to the Enable bit in the Comparator x Control registers (COMPCTRLx.ENABLE). The AC is disabled by writing a zero to CTRLA.ENABLE. This will also disable the individual comparators, but will not clear their COMPCTRLx.ENABLE bits.

The AC is reset by writing a one to the Software Reset bit in the Control A register (CTRLA.SWRST). All registers in the AC, except DEBUG, will be reset to their initial state, and the AC will be disabled. Refer to the CTRLA register for details.

34.6.2.3 Starting a Comparison

Each comparator channel can be in one of two different measurement modes, determined by the Single bit in the Comparator x Control register (COMPCTRLx.SINGLE):

- Continuous measurement
- Single-shot

After being enabled, a start-up delay is required before the result of the comparison is ready. This start-up time is measured automatically to account for environmental changes, such as temperature or voltage supply level, and is specified in “Electrical Characteristics” on page 801.

During the start-up time, the COMP output is not available. If the supply voltage is below 2.5V, the start-up time is also dependent on the voltage doubler. If the supply voltage is guaranteed to be above 2.5V, the voltage doubler can be disabled by writing the Low-Power Mux bit in the Control A register (CTRLA.LPMUX) to one.

Table 37-9. Typical Peripheral Current Consumption

Peripheral	Conditions	Typ.	Unit
RTC	f_{GCLK_RTC} = 32kHz, 32bit counter mode	7.4	µA
WDT	f_{GCLK_WDT} = 32kHz, normal mode with EW	5.5	
AC	Both f_{GCLK} = 8MHz, Enable both COMP	31.3	
TCx(1)	f_{GCLK} = 8MHz, Enable + COUNTER in 8bit mode	50	
TCC2	f_{GCLK} = 8MHz, Enable + COUNTER	95.5	
TCC1	f_{GCLK} = 8MHz, Enable + COUNTER	167.5	
TCC0	f_{GCLK} = 8MHz, Enable + COUNTER	180.3	
SERCOMx.I2CM(2)	f_{GCLK} = 8MHz, Enable	69.7	
SERCOMx.I2CS	f_{GCLK} = 8MHz, Enable	29.2	
SERCOMx.SPI	f_{GCLK} = 8MHz, Enable	64.6	
SERCOMx.USART	f_{GCLK} = 8MHz, Enable	65.5	
DMAC(3)	RAM to RAM transfer	399.5	

Notes:
1. All TCs from 4 to 7 share the same power consumption values.
2. All SERCOMs from 0 to 5 share the same power consumption values.
3. The value includes the power consumption of the R/W access to the RAM.

37.9.2 I^2C Pins

Refer to "I/O Multiplexing and Considerations" on page 23 to get the list of I^2C pins.

Table 37-11. I^2C Pins Characteristics in I^2C Configuration

Parameter	Condition	Symbol	Min.	Typ.	Max.	Unit
Pull-up - Pull-down resistance		R_{PULL}	20	40	60	kΩ
Input low-level voltage	V_{DD} = 2.7V-3.63V	V_{IL}	-	-	$0.3 \times V_{DD}$	V
Input high-level voltage	V_{DD} = 2.7V-3.63V	V_{IH}	$0.55 \times V_{DD}$	-	-	
Hysteresis of Schmitt trigger inputs		V_{HYS}	$0.08 \times V_{DD}$	-	-	
Output low-level voltage	V_{DD} > 2.0V I_{OL} = 3mA	V_{OL}	-	-	0.4	
	$V_{DD} \leq$ 2.0V I_{OL} = 2mA		-	-	$0.2 \times V_{DD}$	
Capacitance for each I/O Pin		C_I				pF
Output low-level current	V_{OL} = 0.4V Standard, Fast and HS Modes	I_{OL}	3			mA
	V_{OL} = 0.4V Fast Mode +		20	-	-	
	V_{OL} = 0.6V		6	-	-	
SCL clock frequency		f_{SCL}	-	-	3.4	MHz
Value of pull-up resistor	$f_{SCL} \leq$ 100kHz	R_P				Ω
	f_{SCL} > 100kHz					

I^2C pins timing characteristics can be found in "SERCOM in I2C Mode Timing" on page 842.

Table 37-12. I^2C Pins Characteristics in I/O Configuration

Parameter	Conditions	Symbol	Min.	Typ.	Max.	Unit
Pull-up - Pull-down resistance		R_{PULL}	20	40	60	kΩ
Input low-level voltage	V_{DD} = 2.7V-3.63V	V_{IL}	-	-	$0.3 \times V_{DD}$	V
Input high-level voltage	V_{DD} = 2.7V-3.63V	V_{IH}	$0.55 \times V_{DD}$	-	-	
Output low-level voltage	V_{DD} > 2.7V, IOL max	V_{OL}	-	$0.1 \times V_{DD}$	$0.2 \times V_{DD}$	
Output high-level voltage	V_{DD} > 2.7V, IOH max	V_{OH}	$0.8*V_{DD}$	$0.9 \times V_{DD}$	-	