Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

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

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	ARM® Cortex®-M0+
Core Size	32-Bit Single-Core
Speed	48MHz
Connectivity	I²C, LINbus, SPI, UART/USART, USB
Peripherals	Brown-out Detect/Reset, DMA, I²S, POR, PWM, WDT
Number of I/O	52
Program Memory Size	128KB (128K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	16K x 8
Voltage - Supply (Vcc/Vdd)	1.62V ~ 3.6V
Data Converters	A/D 20x12b; D/A 1x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°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/atsamd21j17a-aft

Bit 20 – I2S
Writing a zero to these bits has no effect.

Writing a one to these bits will clear the Write Protect bit for the corresponding peripherals.

Value	Description
0	Write-protection is disabled.
1	Write-protection is enabled.

Bit 19 – PTC
Writing a zero to these bits has no effect.

Writing a one to these bits will clear the Write Protect bit for the corresponding peripherals.

Value	Description
0	Write-protection is disabled.
1	Write-protection is enabled.

Bit 18 – DAC:
Writing a zero to these bits has no effect.

Writing a one to these bits will clear the Write Protect bit for the corresponding peripherals.

Value	Description
0	Write-protection is disabled.
1	Write-protection is enabled.

Bit 17 – AC
Writing a zero to these bits has no effect.

Writing a one to these bits will clear the Write Protect bit for the corresponding peripherals.

Value	Description
0	Write-protection is disabled.
1	Write-protection is enabled.

Bit 16 – ADC
Writing a zero to these bits has no effect.

Writing a one to these bits will clear the Write Protect bit for the corresponding peripherals.

Value	Description
0	Write-protection is disabled.
1	Write-protection is enabled.

Bits 11, 12, 13, 14, 15 – TC3, TC4, TC5, TC4, TC7
Writing a zero to these bits has no effect.

Writing a one to these bits will clear the Write Protect bit for the corresponding peripherals.

Value	Description
0	Write-protection is disabled.
1	Write-protection is enabled.

Bits 8, 9, 10 – TCCn
Writing a zero to these bits has no effect.

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
	FKBC[3:0]				JEPCC[3:0]			
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 7:4 – FKBC[3:0]: 4KB Count
These bits will always return zero when read, indicating that this debug component occupies one 4KB block.

Bits 3:0 – JEPCC[3:0]: JEP-106 Continuation Code
These bits will always return zero when read, indicating an Atmel device.

13.13.15 Peripheral Identification 0

Name: PID0
Offset: 0x1FE0
Reset: 0x000000D0
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

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

16.6.2.3 Selecting the Main Clock Source

Refer to *GCLK – Generic Clock Controller* for details on how to configure the main clock source.

Related Links

GCLK - Generic Clock Controller

16.6.2.4 Selecting the Synchronous Clock Division Ratio

The main clock feeds an 8-bit prescaler, which can be used to generate the synchronous clocks. By default, the synchronous clocks run on the undivided main clock. The user can select a prescaler division for the CPU clock by writing the CPU Prescaler Selection bits in the CPU Select register (CPUSEL.CPUDIV), resulting in a CPU clock frequency determined by this equation:

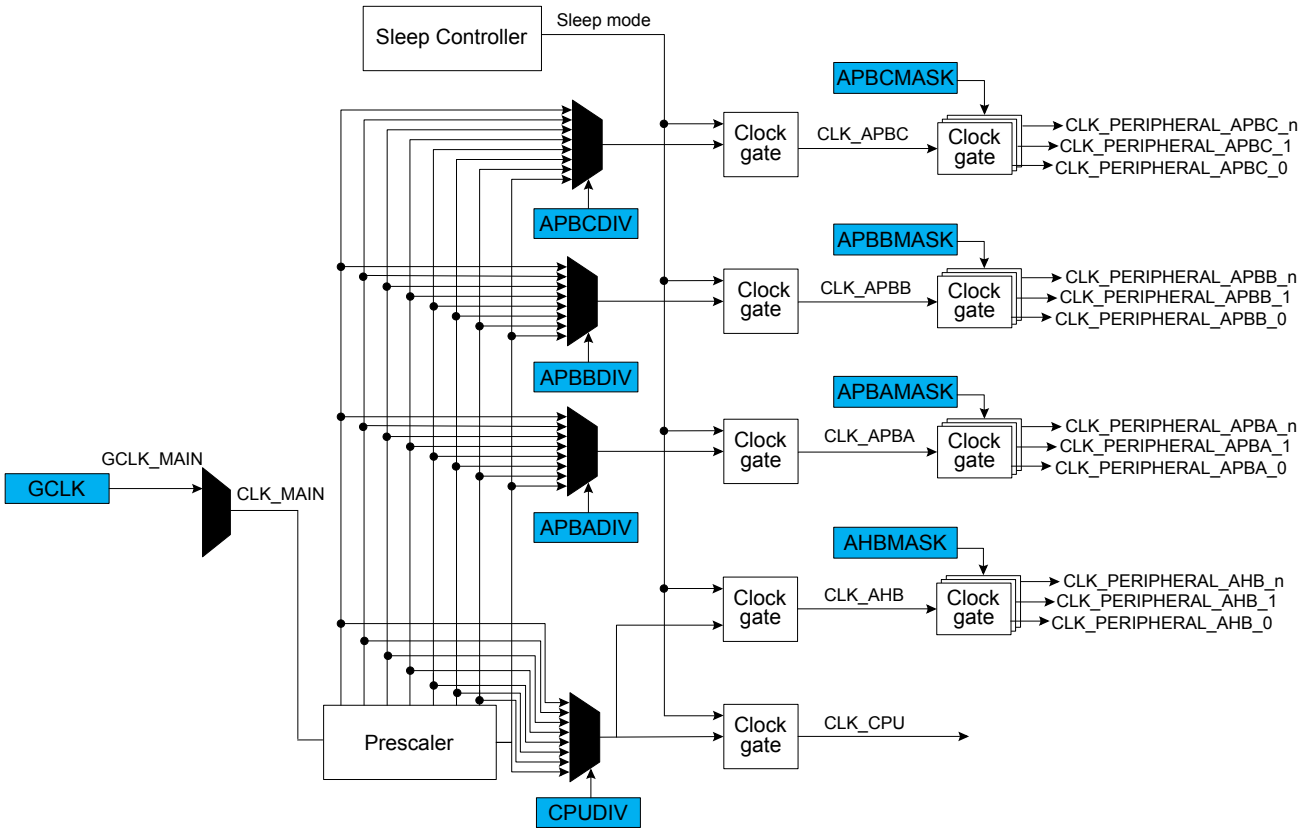
$$f_{CPU} = \frac{f_{main}}{2^{CPUDIV}}$$

Similarly, the clock for the APBx can be divided by writing their respective registers (APBxSEL.APBxDIV). To ensure correct operation, frequencies must be selected so that $f_{CPU} \geq f_{APBx}$. Also, frequencies must never exceed the specified maximum frequency for each clock domain.

Note: The AHB clock is always equal to the CPU clock.

CPUSEL and APBxSEL can be written without halting or disabling peripheral modules. Writing CPUSEL and APBxSEL allows a new clock setting to be written to all synchronous clocks at the same time. It is possible to keep one or more clocks unchanged. This way, it is possible to, for example, scale the CPU speed according to the required performance, while keeping the APBx frequency constant.

Figure 16-2. Synchronous Clock Selection and Prescaler



Offset	Name	Bit Pos.								
0x26 ... 0x27	Reserved									
0x28	DFLLVAL	7:0	FINE[7:0]							
0x29		15:8	COARSE[5:0]						FINE[9:8]	
0x2A		23:16	DIFF[7:0]							
0x2B		31:24	DIFF[15:8]							
0x2C	DFLLMUL	7:0	MUL[7:0]							
0x2D		15:8	MUL[15:8]							
0x2E		23:16	FSTEP[7:0]							
0x2F		31:24	CSTEP[5:0]						FSTEP[9:8]	
0x30	DFLLSYNC	7:0	READREQ							
0x31 ... 0x33	Reserved									
0x34	BOD33	7:0		RUNSTDBY		ACTION[1:0]		HYST	ENABLE	
0x35		15:8	PSEL[3:0]						CEN	MODE
0x36		23:16			LEVEL[5:0]					
0x37		31:24								
0x38 ... 0x3B	Reserved									
0x3C	VREG	7:0		RUNSTDBY						
0x3D		15:8			FORCELDO					
0x3E ... 0x3F	Reserved									
0x40	VREF	7:0						BGOUTEN	TSEN	
0x41		15:8								
0x42		23:16	CALIB[7:0]							
0x43		31:24						CALIB[10:8]		
0x44	DPLLCTRLA	7:0	ONDEMAND	RUNSTDBY					ENABLE	
0x45 ... 0x47	Reserved									
0x48	DPLLRATIO	7:0	LDR[7:0]							
0x49		15:8					LDR[11:8]			
0x4A		23:16					LDRFRAC[3:0]			
0x4B		31:24								
0x4C	DPLLCTRLB	7:0			REFCLK[1:0]		WUF	LPEN	FILTER[1:0]	
0x4D		15:8				LBYPASS		LTIME[2:0]		
0x4E		23:16	DIV[7:0]							
0x4F		31:24						DIV[10:8]		
0x50	DPLLSTATUS	7:0					DIV	ENABLE	CLKRDY	LOCK

CRC on DMA data CRC-16 or CRC-32 calculations can be performed on data passing through any DMA channel. Once a DMA channel is selected as the source, the CRC engine will continuously generate the CRC on the data passing through the DMA channel. The checksum is available for readout once the DMA transaction is completed or aborted. A CRC can also be generated on SRAM, Flash, or I/O memory by passing these data through a DMA channel. If the latter is done, the destination register for the DMA data can be the data input (CRCDATAIN) register in the CRC engine.

CRC using the I/O interface Before using the CRC engine with the I/O interface, the application must set the CRC Beat Size bits in the CRC Control register (CRCCTRL.CRCBEATSIZE). 8/16/32-bit bus transfer type can be selected.

CRC can be performed on any data by loading them into the CRC engine using the CPU and writing the data to the CRCDATAIN register. Using this method, an arbitrary number of bytes can be written to the register by the CPU, and CRC is done continuously for each byte. This means if a 32-bit data is written to the CRCDATAIN register the CRC engine takes four cycles to calculate the CRC. The CRC complete is signaled by a set CRCBUSY bit in the CRCSTATUS register. New data can be written only when CRCBUSY flag is not set.

20.6.4 DMA Operation

Not applicable.

20.6.5 Interrupts

The DMAC channels have the following interrupt sources:

- Transfer Complete (TCMPL): Indicates that a block transfer is completed on the corresponding channel. Refer to Data Transmission for details.
- Transfer Error (TERR): Indicates that a bus error has occurred during a burst transfer, or that an invalid descriptor has been fetched. Refer to Error Handling for details.
- Channel Suspend (SUSP): Indicates that the corresponding channel has been suspended. Refer to Channel Suspend and Data Transmission for details.

Each interrupt source has an interrupt flag associated with it. The interrupt flag in the Channel Interrupt Flag Status and Clear (CHINTFLAG) register is set when the interrupt condition occurs. Each interrupt can be individually enabled by setting the corresponding bit in the Channel Interrupt Enable Set register (CHINTENSET=1), and disabled by setting the corresponding bit in the Channel Interrupt Enable Clear register (CHINTENCLR=1).

An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until the interrupt flag is cleared, the interrupt is disabled, the DMAC is reset or the corresponding DMA channel is reset. See CHINTFLAG for details on how to clear interrupt flags. All interrupt requests are ORed together on system level to generate one combined interrupt request to the NVIC.

The user must read the Channel Interrupt Status (INTSTATUS) register to identify the channels with pending interrupts and must read the Channel Interrupt Flag Status and Clear (CHINTFLAG) register to determine which interrupt condition is present for the corresponding channel. It is also possible to read the Interrupt Pending register (INTPEND), which provides the lowest channel number with pending interrupt and the respective interrupt flags.

Note: Interrupts must be globally enabled for interrupt requests to be generated.

Related Links

Nested Vector Interrupt Controller

Property: PAC Write-Protection, Enable-Protected

Bit	31	30	29	28	27	26	25	24
	BASEADDR[31:24]							
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	23	22	21	20	19	18	17	16
	BASEADDR[23:16]							
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
	BASEADDR[15:8]							
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
	BASEADDR[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 31:0 – BASEADDR[31:0]: Descriptor Memory Base Address
These bits store the Descriptor memory section base address. The value must be 128-bit aligned.

20.8.16 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
	WRBADDR[31:24]							
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	23	22	21	20	19	18	17	16
	WRBADDR[23:16]							
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
	WRBADDR[15:8]							
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

Offset: 0x08
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
	DIRSET[31:24]							
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	23	22	21	20	19	18	17	16
	DIRSET[23:16]							
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
	DIRSET[15:8]							
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
	DIRSET[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 31:0 – DIRSET[31:0]: Port Data Direction Set
Writing '0' to a bit has no effect.

Writing '1' to a bit will set the corresponding bit in the DIR register, which configures the I/O pin as an output.

Value	Description
0	The corresponding I/O pin in the PORT group will keep its configuration.
1	The corresponding I/O pin in the PORT group is configured as an output.

23.8.4 Data Direction Toggle

This register allows the user to toggle the direction of one or more I/O pins, without doing a read-modify-write operation. Changes in this register will also be reflected in the Data Direction (DIR), Data Direction Set (DIRSET) and Data Direction Clear (DIRCLR) registers.

Name: DIRTGL
Offset: 0x0C
Reset: 0x00000000
Property: PAC Write-Protection

Bit	31	30	29	28	27	26	25	24
	DIRTGL[31:24]							
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

Writing a one to this bit will clear the Event Detected Channel n Interrupt Enable bit, which disables the Event Detected Channel n interrupt.

Value	Description
0	The Event Detected Channel n interrupt is disabled.
1	The Event Detected Channel n interrupt is enabled.

Bits 19,18,17,16,7,6,5,4,3,2,1,0 – OVRn : Channel n Overrun Interrupt Enable [n=11..0]
Writing a zero to this bit has no effect.

Writing a one to this bit will clear the Overrun Channel n Interrupt Enable bit, which disables the Overrun Channel n interrupt.

Value	Description
0	The Overrun Channel n interrupt is disabled.
1	The Overrun Channel n interrupt is enabled.

24.8.6 Interrupt Enable Set

Name: INTENSET
Offset: 0x14
Reset: 0x00000000
Property: Write-Protected

Bit	31	30	29	28	27	26	25	24
					EVD11	EVD10	EVD9	EVD8
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	23	22	21	20	19	18	17	16
					OVR11	OVR10	OVR9	OVR8
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
	EVD7	EVD6	EVD5	EVD4	EVD3	EVD2	EVD1	EVD0
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
	OVR7	OVR6	OVR5	OVR4	OVR3	OVR2	OVR1	OVR0
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 27,26,25,24,15,14,13,12,11,10,9,8 – EVDn : Channel n Event Detection Interrupt Enable [n=11..0]
Writing a zero to this bit has no effect.

Writing a one to this bit will set the Event Detected Channel n Interrupt Enable bit, which enables the Event Detected Channel n interrupt.

The SERCOM interrupts can be used to wake up the device from sleep modes. Refer to the different SERCOM mode chapters for details.

25.6.8 Synchronization

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

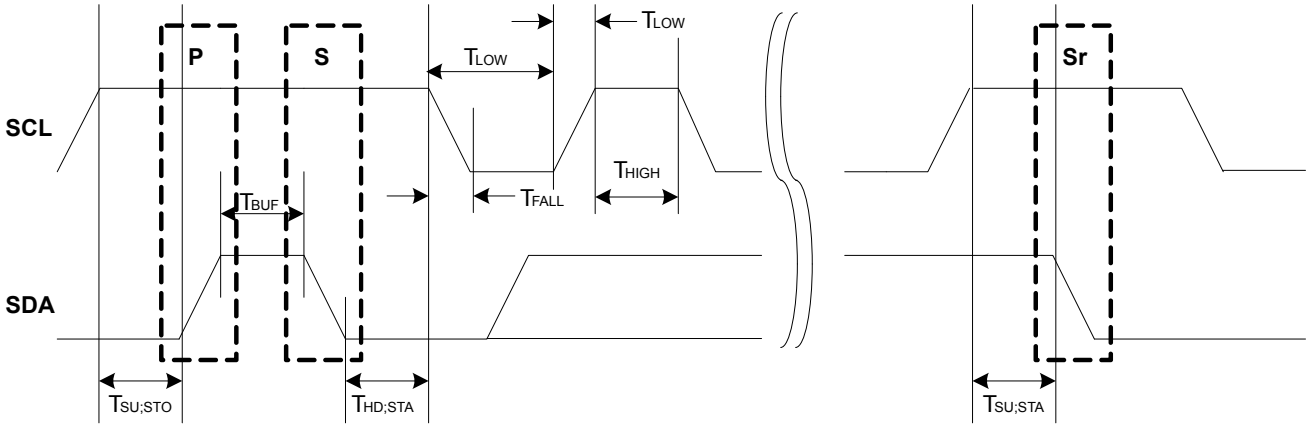
Required write-synchronization is denoted by the "Write-Synchronized" property in the register description.

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

Related Links

Register Synchronization

Figure 28-6. 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.

Name	Description
Counter	The clock control is handled externally (e.g. counting external events)
CC	For compare operations, the CC are referred to as “compare channels” For capture operations, the CC are referred to as “capture channels.”

The counter in the TC can either count events from the Event System, or clock ticks of the GCLK_TCx clock, which may be divided by the prescaler.

The counter value is passed to the CCx where it can be either compared to user-defined values or captured.

The compare and capture registers (CCx) and counter register (COUNT) can be configured as 8-, 16- or 32-bit registers, with according MAX values. Mode settings determine the maximum range of the counter.

In 8-bit mode, Period Value (PER) is also available. The counter range and the operating frequency determine the maximum time resolution achievable with the TC peripheral.

The TC can be set to count up or down. Under normal operation, the counter value is continuously compared to the TOP or ZERO value to determine whether the counter has reached that value. On a comparison match the TC can request DMA transactions, or generate interrupts or events for the Event System. On a comparison match the TC can request DMA transactions, or generate interrupts or events for the Event System.

In compare operation, the counter value is continuously compared to the values in the CCx registers. In case of a match the TC can request DMA transactions, or generate interrupts or events for the Event System. In waveform generator mode, these comparisons are used to set the waveform period or pulse width.

Capture operation can be enabled to perform input signal period and pulse width measurements, or to capture selectable edges from an IO pin or internal event from Event System.

30.6.2 Basic Operation

30.6.2.1 Initialization

The following registers are enable-protected, meaning that they can only be written when the TC is disabled (CTRLA.ENABLE =0):

- Control A register (CTRLA), except the Run Standby (RUNSTDBY), Enable (ENABLE) and Software Reset (SWRST) bits

Enable-protected bits in the CTRLA register can be written at the same time as CTRLA.ENABLE is written to '1', but not at the same time as CTRLA.ENABLE is written to '0'. Enable-protection is denoted by the "Enable-Protected" property in the register description. The following bits are enable-protected:

- Event Action bits in the Event Control register (EVCTRL.EVACT)

Before enabling the TC, the peripheral must be configured by the following steps:

1. Enable the TC bus clock (CLK_TCx_APB).
2. Select 8-, 16- or 32-bit counter mode via the TC Mode bit group in the Control A register (CTRLA.MODE). The default mode is 16-bit.
3. Select one wave generation operation in the Waveform Generation Operation bit group in the Control A register (CTRLA.WAVEGEN).
4. If desired, the GCLK_TCx clock can be prescaled via the Prescaler bit group in the Control A register (CTRLA.PRESCALER).

Name: CCx
Offset: 0x18+i*0x2 [i=0..1]
Reset: 0x0000
Property: Write-Synchronized

Bit	15	14	13	12	11	10	9	8
	CC[15:8]							
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
	CC[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 15:0 – CC[15:0]: Channel x Compare/Capture Value
These bits contain the compare/capture value in 16-bit TC mode. In Match frequency (MFRQ) or Match PWM (MPWM) waveform operation (CTRLA.WAVEGEN), the CC0 register is used as a period register.

30.8.14.3 Channel x Compare/Capture Value, 32-bit Mode

Name: CCx
Offset: 0x18+i*0x4 [i=0..1]
Reset: 0x00000000
Property: Write-Synchronized

Bit	31	30	29	28	27	26	25	24
	CC[31:24]							
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	23	22	21	20	19	18	17	16
	CC[23:16]							
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
	CC[15:8]							
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
	CC[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 31:0 – CC[31:0]: Channel x Compare/Capture Value
These bits contain the compare/capture value in 32-bit TC mode. In Match frequency (MFRQ) or Match PWM (MPWM) waveform operation (CTRLA.WAVEGEN), the CC0 register is used as a period register.

31.6.3.5 Recoverable Faults

Recoverable faults can restart or halt the timer/counter. Two faults, called Fault A and Fault B, can trigger recoverable fault actions on the compare channels CC0 and CC1 of the TCC. The compare channels' outputs can be clamped to inactive state either as long as the fault condition is present, or from the first valid fault condition detection on until the end of the timer/counter cycle.

Fault Inputs

The first two channel input events (TCCxMC0 and TCCxMC1) can be used as Fault A and Fault B inputs, respectively. Event system channels connected to these fault inputs must be configured as asynchronous. The TCC must work in a PWM mode.

Fault Filtering

There are three filters available for each input Fault A and Fault B. They are configured by the corresponding Recoverable Fault n Configuration registers (FCTRLA and FCTRLB). The three filters can either be used independently or in any combination.

Input Filtering

By default, the event detection is asynchronous. When the event occurs, the fault system will immediately and asynchronously perform the selected fault action on the compare channel output, also in device power modes where the clock is not available. To avoid false fault detection on external events (e.g. due to a glitch on an I/O port) a digital filter can be enabled and configured by the Fault B Filter Value bits in the Fault n Configuration registers (FCTRLn.FILTERVAL). If the event width is less than FILTERVAL (in clock cycles), the event will be discarded. A valid event will be delayed by FILTERVAL clock cycles.

Fault Blanking

This ignores any fault input for a certain time just after a selected waveform output edge. This can be used to prevent false fault triggering due to signal bouncing, as shown in the figure below. Blanking can be enabled by writing an edge triggering configuration to the Fault n Blanking Mode bits in the Recoverable Fault n Configuration register (FCTRLn.BLANK). The desired duration of the blanking must be written to the Fault n Blanking Time bits (FCTRLn.BLANKVAL).
The blanking time $t_b$is calculated by

$$t_b = \frac{1 + \mathrm{BLANKVAL}}{f_{\mathrm{GCLK\_TCCx\_PRESC}}}$$

Here, $f_{GCLK\_TCCx\_PRESC}$ is the frequency of the prescaled peripheral clock frequency f_{GCLK_TCCx}.

The maximum blanking time (FCTRLn.BLANKVAL= 255) at f_{GCLK_TCCx}=96MHz is 2.67µs (no prescaler) or 170µs (prescaling). For f_{GCLK_TCCx}=1MHz, the maximum blanking time is either 170µs (no prescaling) or 10.9ms (prescaling enabled).

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

Bit 7 – FLENCE: Frame Length Control Enable
When this bit is '1', the time between Start-of-Frames can be tuned by up to +/-0.06% using FLENC[3:0].
Note: In Low Speed mode, FLENCE must be '0'.

Value	Description
0	Start-of-Frame is generated every 1ms.
1	Start-of-Frame generation depends on the signed value of FLENC[3:0]. USB Start-of-Frame period equals 1ms + (FLENC[3:0]/12000)ms

Bits 3:0 – FLENC[3:0]: Frame Length Control
These bits define the signed value of the 4-bit FLENC that is added to the Internal Frame Length when FLENCE is '1'. The internal Frame length is the top value of the frame counter when FLENCE is zero.

32.8.5.3 Status

Name: STATUS
Offset: 0x0C
Reset: 0x00
Property: Read only

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

Bits 7:6 – LINESTATE[1:0]: USB Line State Status
These bits define the current line state DP/DM.

LINESTATE[1:0]	USB Line Status
0x0	SE0/RESET
0x1	FS-J or LS-K State
0x2	FS-K or LS-J State

Bits 3:2 – SPEED[1:0]: Speed Status
These bits define the current speed used by the host.

SPEED[1:0]	Speed Status
0x0	Full-speed mode
0x1	Low-speed mode
0x2	Reserved
0x3	Reserved

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Units
$I_{XOSC32K}$	Current consumption		-	1.22	2.19	µA
ESR	Crystal equivalent series resistance f=32.768kHz , Safety Factor = 3	C_L=12.5pF	-	-	100	kΩ

37.12.3 Digital Frequency Locked Loop (DFLL48M) Characteristics

Table 37-50. DFLL48M Characteristics - Open Loop Mode(1)

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Units
f_{OUT}	Output frequency	DFLLVAL.COARSE = DFLL48M COARSE CAL DFLLVAL.FINE = 512	47	48	49	MHz
I_{DFLL}	Power consumption on V_{DDIN}	IDFLLVAL.COARSE = DFLL48M COARSE CAL DFLLVAL.FINE = 512	-	403	453	µA
$t_{STARTUP}$	Start-up time	DFLLVAL.COARSE = DFLL48M COARSE CAL DFLLVAL.FINE = 512 f_{OUT} within 90 % of final value	7	8	9	µs

Note: 1. DFLL48M in Open loop after calibration at room temperature.

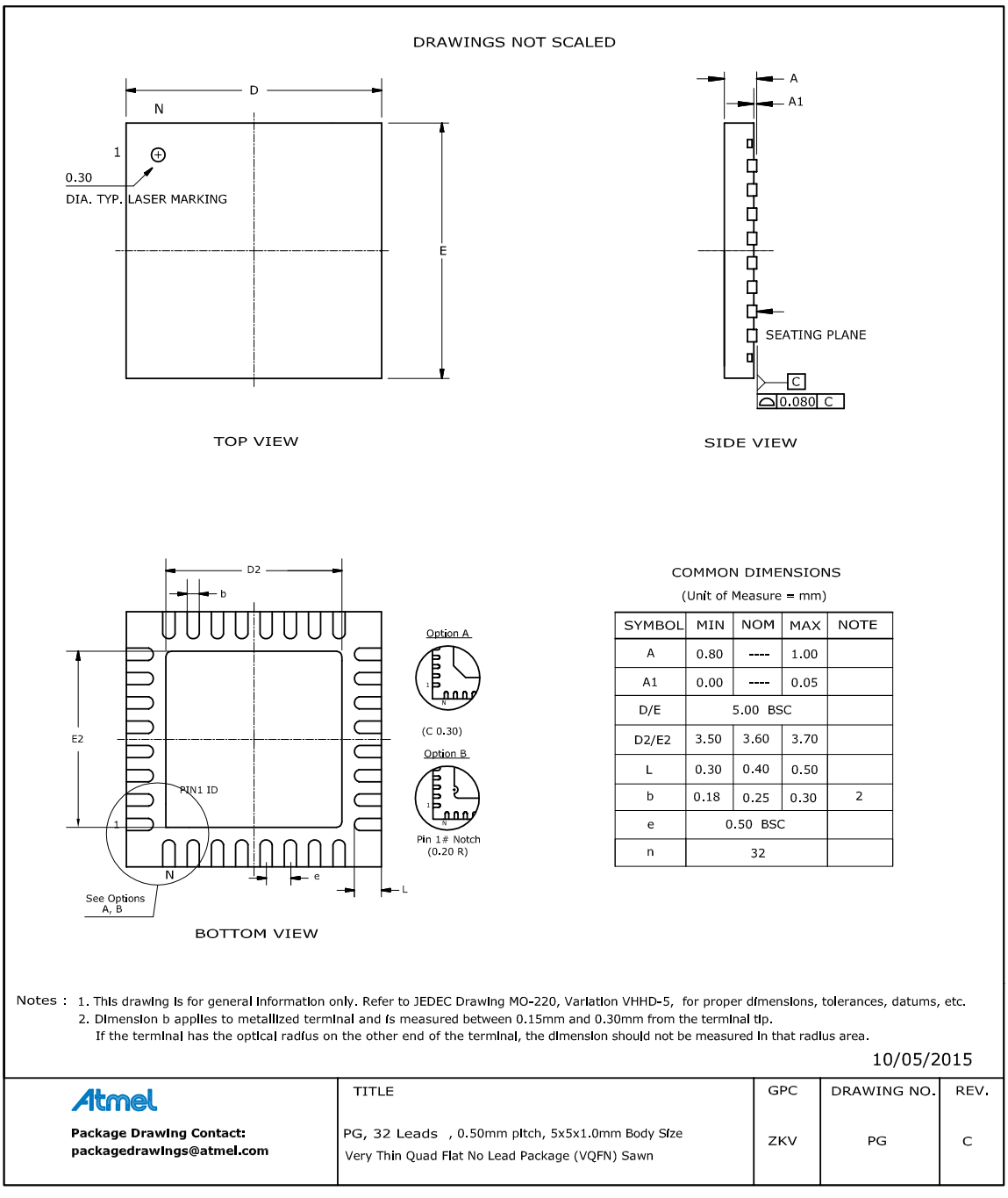
Table 37-51. DFLL48M Characteristics - Closed Loop Mode(1) (Device Variant A)

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Units
f_{OUT}	Average Output frequency	f_{REF} = XTAL, 32 .768kHz, 100ppm DFLLMUL = 1464	47.963	47.972	47.981	MHz
f_{REF}	Reference frequency		0.732	32.768	33	kHz
Jitter	Cycle to Cycle jitter	f_{REF} = XTAL, 32 .768kHz, 100ppm DFLLMUL = 1464	-	-	0.42	ns

Table 38-22. Package Reference

JEDEC Drawing Reference	MS-026
JESD97 Classification	E3

38.2.8 32 pin QFN



Note: The exposed die attach pad is connected inside the device to GND and GNDANA.

Table 38-23. Device and Package Maximum Weight

90	mg

11 – When the Peripheral Access Controller (PAC) protection is enabled, writing to WAVE or WAVEB registers will not cause a hardware exception.
Errata reference: 11468
Fix/Workaround:
None

12 – If the MCx flag in the INTFLAG register is set when enabling the DMA, this will trigger an immediate DMA transfer and overwrite the current buffered value in the TCC register.
Errata reference: 12155
Fix/Workaround:
None

40.1.2.13 PTC

1 – WCOMP interrupt flag is not stable. The WCOMP interrupt flag will not always be set as described in the datasheet.
Errata reference: 12860
Fix/Workaround:
Do not use the WCOMP interrupt. Use the WCOMP event.

40.1.3 Die Revision C

40.1.3.1 Device

1 – When VDDIN is lower than the POR threshold during power rise or fall, an internal pull-up resistor is enabled on pins with PTC functionality (see PORT Function Multiplexing). Note that this behavior will be present even if PTC functionality is not enabled on the pin. The POR level is defined in the "Power-On Reset (POR) Characteristics" chapter.
Errata reference: 12117
Fix/Workaround:
Use a pin without PTC functionality if the pull-up could damage your application during power up.

2 – In single shot mode and at 125°C, the ADC conversions have linearity errors.
Errata reference: 13277
Fix/Workaround:
- Workaround 1: At 125°C, do not use the ADC in single shot mode; use the ADC in free running mode only.
- Workaround 2: At 125°C, use the ADC in single shot mode only with VDDANA > 3V.

3 – In the table ""NVM User Row Mapping"", the WDT Window bitfield default value on silicon is not as specified in the datasheet. The datasheet defines the default value as 0x5, while it is 0xB on silicon.
Errata reference: 13951
Fix/Workaround:
None.

4 – On pin PA24 and PA25 the pull-up and pull-down configuration is not disabled automatically when alternative pin function is enabled except for USB.

6 – In I2C Slave mode, writing the CTRLB register when in the AMATCH or DRDY interrupt service routines can cause the state machine to reset.

Errata reference: 13574

Fix/Workaround:

Write CTRLB.ACKACT to 0 using the following sequence:

```
// If higher priority interrupts exist, then disable so that the
// following two writes are atomic.
SERCOM - STATUS.reg = 0;
SERCOM - CTRLB.reg = 0;
// Re-enable interrupts if applicable.
```

Write CTRLB.ACKACT to 1 using the following sequence:

```
// If higher priority interrupts exist, then disable so that the
// following two writes are atomic.
SERCOM - STATUS.reg = 0;
SERCOM - CTRLB.reg = SERCOM_I2CS_CTRLB_ACKACT;
// Re-enable interrupts if applicable.
```

Otherwise, only write to CTRLB in the AMATCH or DRDY interrupts if it is to close out a transaction.

When not closing a transaction, clear the AMATCH interrupt by writing a 1 to its bit position instead of using CTRLB.CMD. The DRDY interrupt is automatically cleared by reading/writing to the DATA register in smart mode. If not in smart mode, DRDY should be cleared by writing a 1 to its bit position.

Code replacements examples:

Current:

```
SERCOM - CTRLB.reg |= SERCOM_I2CS_CTRLB_ACKACT;
```

Change to:

```
// If higher priority interrupts exist, then disable so that the
// following two writes are atomic.
SERCOM - STATUS.reg = 0;
SERCOM - CTRLB.reg = SERCOM_I2CS_CTRLB_ACKACT;
// Re-enable interrupts if applicable.
```

Current:

```
SERCOM - CTRLB.reg &= ~SERCOM_I2CS_CTRLB_ACKACT;
```

Change to:

```
// If higher priority interrupts exist, then disable so that the
// following two writes are atomic.
SERCOM - STATUS.reg = 0;
SERCOM - CTRLB.reg = 0;
// Re-enable interrupts if applicable.
```

Current:

```
/* ACK or NACK address */
SERCOM - CTRLB.reg |= SERCOM_I2CS_CTRLB_CMD(0x3);
```

Change to:

```
// CMD=0x3 clears all interrupts, so to keep the result similar,
// PREC is cleared if it was set.
if (SERCOM - INTFLAG.bit.PREC) SERCOM - INTFLAG.reg =
SERCOM_I2CS_INTFLAG_PREC;
SERCOM - INTFLAG.reg = SERCOM_I2CS_INTFLAG_AMATCH;
```

41. Conventions

41.1 Numerical Notation

Table 41-1. Numerical Notation

Symbol	Description
165	Decimal number
0b0101	Binary number (example 0b0101 = 5 decimal)
'0101'	Binary numbers are given without prefix if unambiguous.
0x3B24	Hexadecimal number
X	Represents an unknown or don't care value
Z	Represents a high-impedance (floating) state for either a signal or a bus

41.2 Memory Size and Type

Table 41-2. Memory Size and Bit Rate

Symbol	Description
KB (kbyte)	kilobyte (2^{10} = 1024)
MB (Mbyte)	megabyte (2^{20} = 1024*1024)
GB (Gbyte)	gigabyte (2^{30} = 1024*1024*1024)
b	bit (binary '0' or '1')
B	byte (8 bits)
1kbit/s	1,000 bit/s rate (not 1,024 bit/s)
1Mbit/s	1,000,000 bit/s rate
1Gbit/s	1,000,000,000 bit/s rate
word	32 bit
half-word	16 bit

41.3 Frequency and Time

Symbol	Description
kHz	1kHz = 10^3Hz = 1,000Hz
KHz	1KHz = 1,024Hz, 32KHz = 32,768Hz
MHz	10^6 = 1,000,000Hz