Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

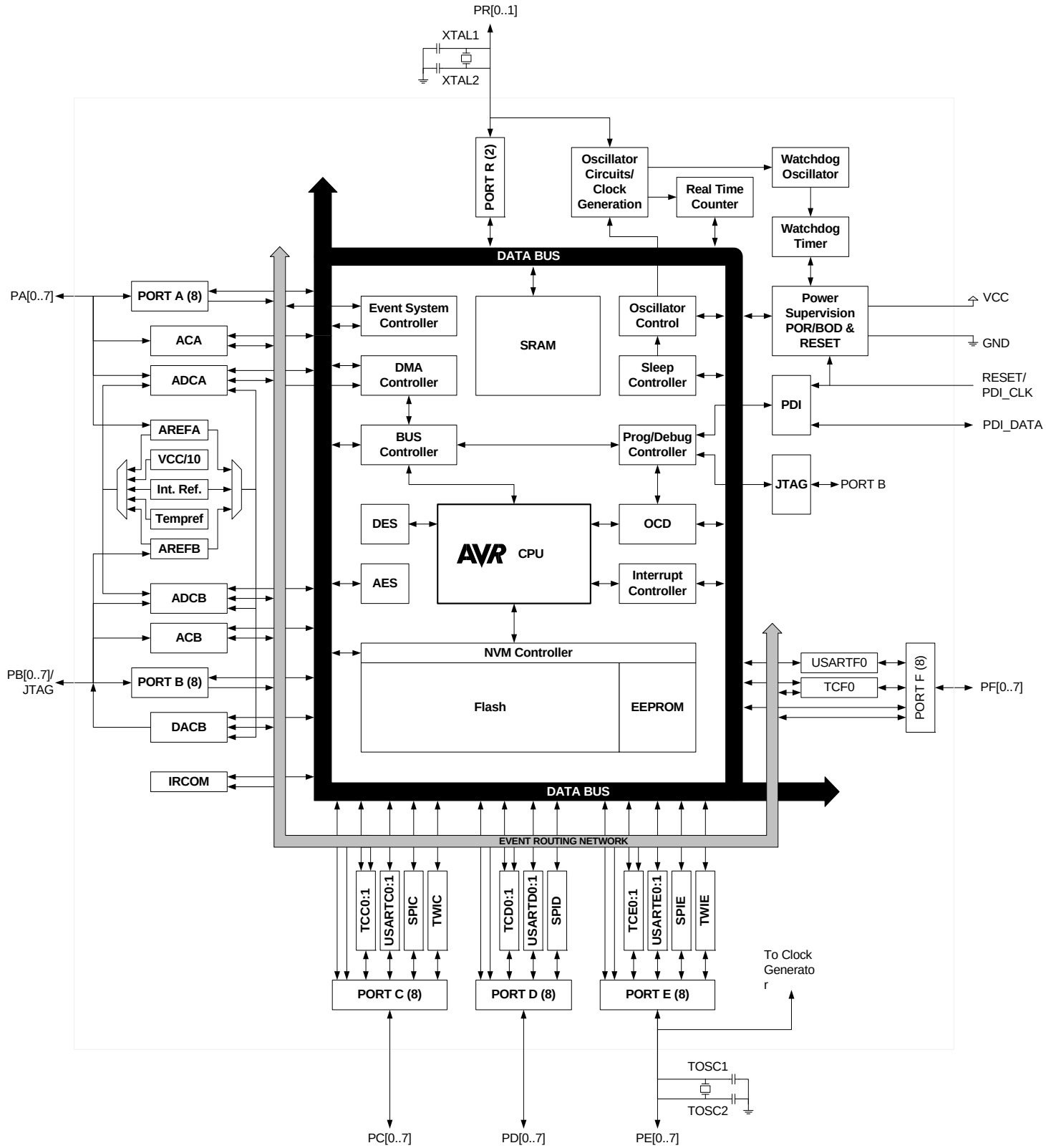
"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" 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	AVR
Core Size	8/16-Bit
Speed	32MHz
Connectivity	I²C, IrDA, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, DMA, POR, PWM, WDT
Number of I/O	50
Program Memory Size	128KB (64K x 16)
Program Memory Type	FLASH
EEPROM Size	2K x 8
RAM Size	8K x 8
Voltage - Supply (Vcc/Vdd)	1.6V ~ 3.6V
Data Converters	A/D 16x12b; D/A 2x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	64-VFQFN Exposed Pad
Supplier Device Package	64-QFN (9x9)
Purchase URL	https://www.e-xfl.com/product-detail/atmel/atxmega128a3-mhr

3.1 Block Diagram

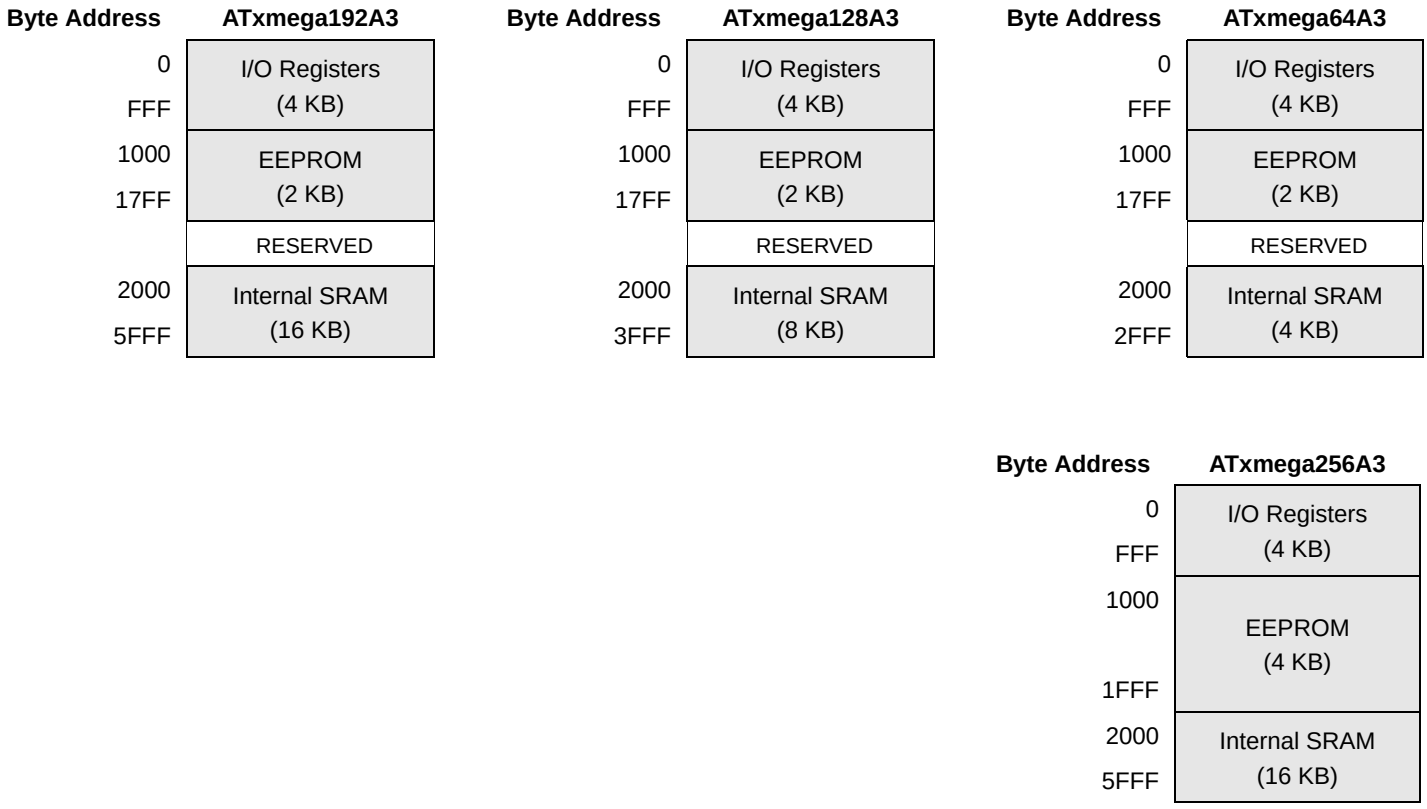
Figure 3-1. XMEGA A3 Block Diagram



7.4 Data Memory

The Data Memory consist of the I/O Memory, EEPROM and SRAM memories, all within one linear address space, see Figure 7-2 on page 11. To simplify development, the memory map for all devices in the family is identical and with empty, reserved memory space for smaller devices.

Figure 7-2. Data Memory Map (Hexadecimal address)



7.7 Flash and EEPROM Page Size

The Flash Program Memory and EEPROM data memory are organized in pages. The pages are word accessible for the Flash and byte accessible for the EEPROM.

Table 7-2 on page 14 shows the Flash Program Memory organization. Flash write and erase operations are performed on one page at a time, while reading the Flash is done one byte at a time. For Flash access the Z-pointer (Z[m:n]) is used for addressing. The most significant bits in the address (FPAGE) gives the page number and the least significant address bits (FWORD) gives the word in the page.

Table 7-2. Number of words and Pages in the Flash.

Devices	Flash Size	Page Size (words)	FWORD	FPAGE	Application		Boot	
					Size	No of Pages	Size	No of Pages
ATxmega64A3	64 KB + 4 KB	128	Z[7:1]	Z[16:8]	64K	256	4 KB	16
ATxmega128A3	128 KB + 8 KB	256	Z[8:1]	Z[17:9]	128K	256	8 KB	16
ATxmega192A3	192 KB + 8 KB	256	Z[8:1]	Z[18:9]	192K	384	8 KB	16
ATxmega256A3	256 KB + 8 KB	256	Z[8:1]	Z[18:9]	256K	512	8 KB	16

Table 7-3 on page 14 shows EEPROM memory organization for the XMEGA A3 devices. EEPROM write and erase operations can be performed one page or one byte at a time, while reading the EEPROM is done one byte at a time. For EEPROM access the NVM Address Register (ADDR[m:n]) is used for addressing. The most significant bits in the address (E2PAGE) gives the page number and the least significant address bits (E2BYTE) gives the byte in the page.

Table 7-3. Number of bytes and Pages in the EEPROM.

Devices	EEPROM Size	Page Size (Bytes)	E2BYTE	E2PAGE	No of Pages
ATxmega64A3	2 KB	32	ADDR[4:0]	ADDR[10:5]	64
ATxmega128A3	2 KB	32	ADDR[4:0]	ADDR[10:5]	64
ATxmega192A3	2 KB	32	ADDR[4:0]	ADDR[10:5]	64
ATxmega256A3	4 KB	32	ADDR[4:0]	ADDR[11:5]	128

8. DMAC - Direct Memory Access Controller

8.1 Features

- **Allows High-speed data transfer**
  - **From memory to peripheral**
  - **From memory to memory**
  - **From peripheral to memory**
  - **From peripheral to peripheral**
- **4 Channels**
- **From 1 byte and up to 16 M bytes transfers in a single transaction**
- **Multiple addressing modes for source and destination address**
  - **Increment**
  - **Decrement**
  - **Static**
- **1, 2, 4, or 8 bytes Burst Transfers**
- **Programmable priority between channels**

8.2 Overview

The XMEGA A3 has a Direct Memory Access (DMA) Controller to move data between memories and peripherals in the data space. The DMA controller uses the same data bus as the CPU to transfer data.

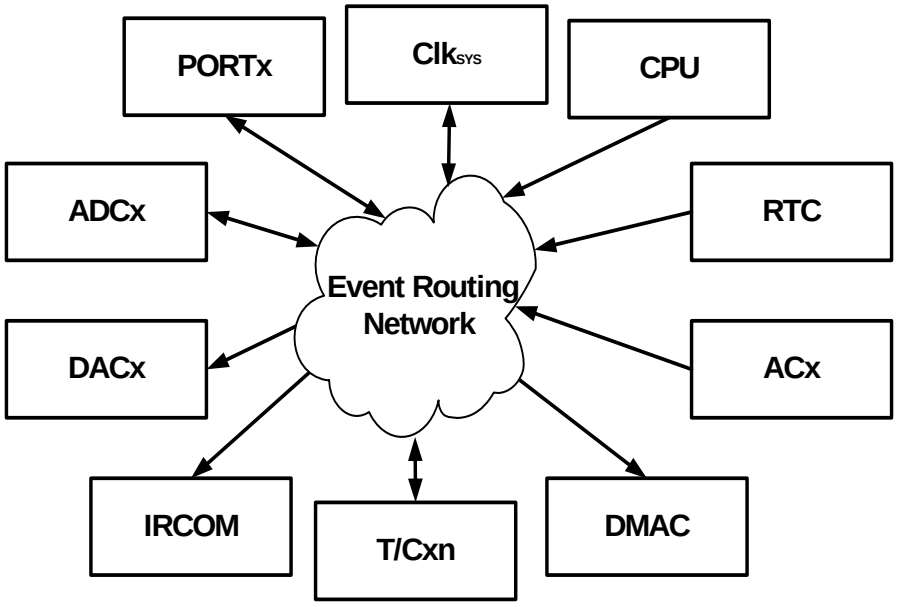
It has 4 channels that can be configured independently. Each DMA channel can perform data transfers in blocks of configurable size from 1 to 64K bytes. A repeat counter can be used to repeat each block transfer for single transactions up to 16M bytes. Each DMA channel can be configured to access the source and destination memory address with incrementing, decrementing or static addressing. The addressing is independent for source and destination address. When the transaction is complete the original source and destination address can automatically be reloaded to be ready for the next transaction.

The DMAC can access all the peripherals through their I/O memory registers, and the DMA may be used for automatic transfer of data to/from communication modules, as well as automatic data retrieval from ADC conversions, data transfer to DAC conversions, or data transfer to or from port pins. A wide range of transfer triggers is available from the peripherals, Event System and software. Each DMA channel has different transfer triggers.

To allow for continuous transfers, two channels can be interlinked so that the second takes over the transfer when the first is finished and vice versa.

The DMA controller can read from memory mapped EEPROM, but it cannot write to the EEPROM or access the Flash.

Figure 9-1. Event system block diagram.



The Event Routing Network can directly connect together ADCs, DACs, Analog Comparators (ACx), I/O ports (PORTx), the Real-time Counter (RTC), Timer/Counters (T/C) and the IR Communication Module (IRCOM). Events can also be generated from software (CPU).

All events from all peripherals are always routed into the Event Routing Network. This consist of eight multiplexers where each can be configured in software to select which event to be routed into that event channel. All eight event channels are connected to the peripherals that can use events, and each of these peripherals can be configured to use events from one or more event channels to automatically trigger a software selectable action.

17. AWEX - Advanced Waveform Extension

17.1 Features

- **Output with complementary output from each Capture channel**
- **Four Dead Time Insertion (DTI) Units, one for each Capture channel**
- **8-bit DTI Resolution**
- **Separate High and Low Side Dead-Time Setting**
- **Double Buffered Dead-Time**
- **Event Controlled Fault Protection**
- **Single Channel Multiple Output Operation (for BLDC motor control)**
- **Double Buffered Pattern Generation**

17.2 Overview

The Advanced Waveform Extension (AWEX) provides extra features to the Timer/Counter in Waveform Generation (WG) modes. The AWEX enables easy and safe implementation of for example, advanced motor control (AC, BLDC, SR, and Stepper) and power control applications.

Any WG output from a Timer/Counter 0 is split into a complimentary pair of outputs when any AWEX feature is enabled. These output pairs go through a Dead-Time Insertion (DTI) unit that enables generation of the non-inverted Low Side (LS) and inverted High Side (HS) of the WG output with dead time insertion between LS and HS switching. The DTI output will override the normal port value according to the port override setting. Optionally the final output can be inverted by using the invert I/O setting for the port pin.

The Pattern Generation unit can be used to generate a synchronized bit pattern on the port it is connected to. In addition, the waveform generator output from Compare Channel A can be distributed to, and override all port pins. When the Pattern Generator unit is enabled, the DTI unit is bypassed.

The Fault Protection unit is connected to the Event System. This enables any event to trigger a fault condition that will disable the AWEX output. Several event channels can be used to trigger fault on several different conditions.

The AWEX is available for TCC0. The notation of this is AWEXC.

18. Hi-Res - High Resolution Extension

18.1 Features

- **Increases Waveform Generator resolution by 2-bits (4x)**
- **Supports Frequency, single- and dual-slope PWM operation**
- **Supports the AWEX when this is enabled and used for the same Timer/Counter**

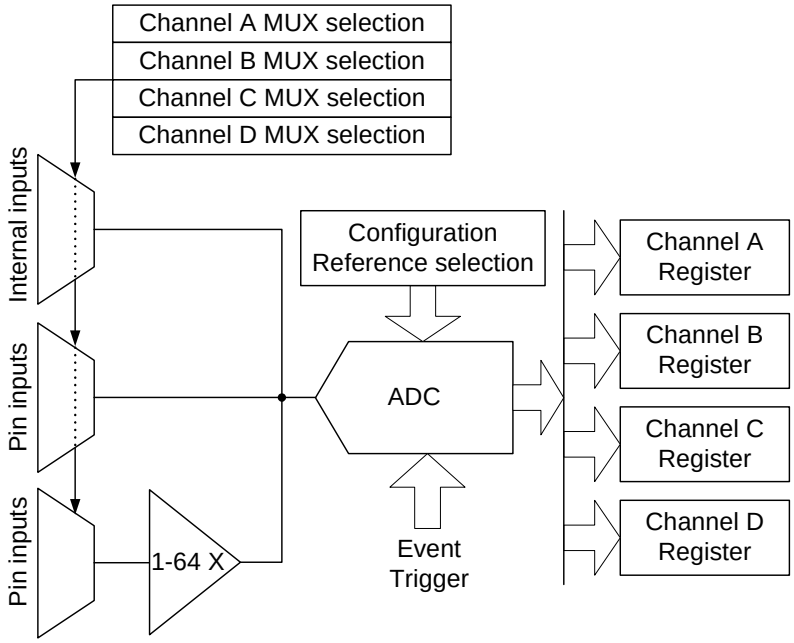
18.2 Overview

The Hi-Resolution (Hi-Res) Extension is able to increase the resolution of the waveform generation output by a factor of 4. When enabled for a Timer/Counter, the Fast Peripheral clock running at four times the CPU clock speed will be as input to the Timer/Counter.

The High Resolution Extension can also be used when an AWEX is enabled and used with a Timer/Counter.

XMEGA A3 devices have four Hi-Res Extensions that each can be enabled for each Timer/Counters pair on PORTC, PORTD, PORTE and PORTF. The notation of these are HIRESC, HIRESD, HIRESE and HIRESF, respectively.

Figure 25-1. ADC overview



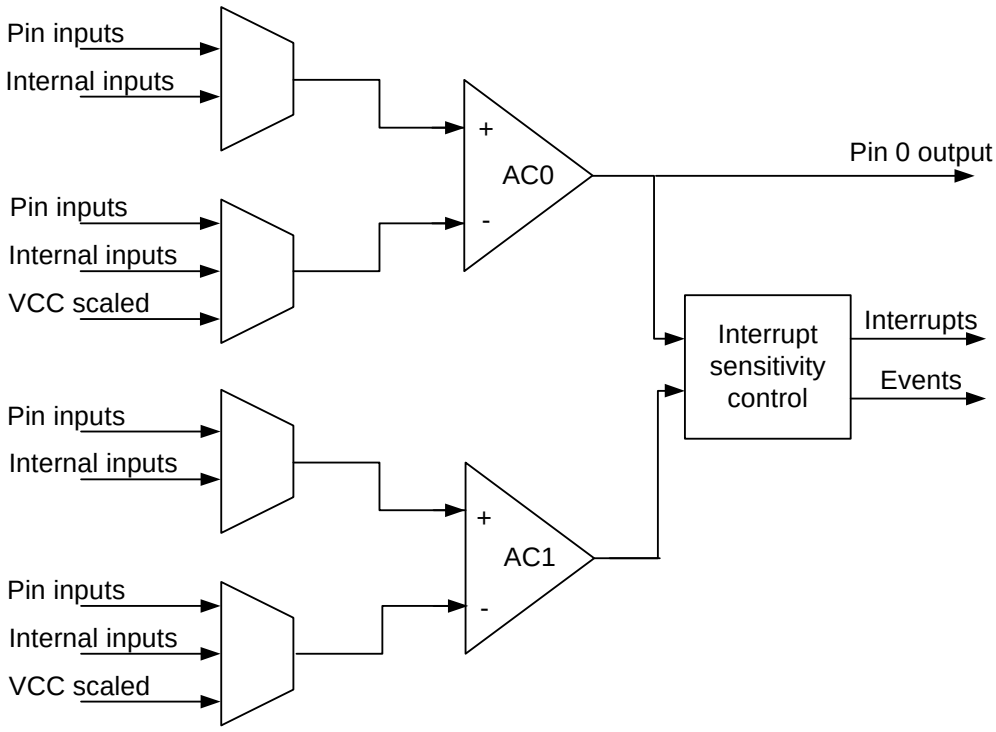
Each ADC has four MUX selection registers with a corresponding result register. This means that four channels can be sampled within 1.5 µs without any intervention by the application other than starting the conversion. The results will be available in the result registers.

The ADC may be configured for 8- or 12-bit result, reducing the minimum conversion time (propagation delay) from 3.5 µs for 12-bit to 2.5 µs for 8-bit result.

ADC conversion results are provided left- or right adjusted with optional ‘1’ or ‘0’ padding. This eases calculation when the result is represented as a signed integer (signed 16-bit number).

PORTA and PORTB each has one ADC. Notation of these peripherals are ADCA and ADCB, respectively.

Figure 27-1. Analog comparator overview



27.3 Input Selection

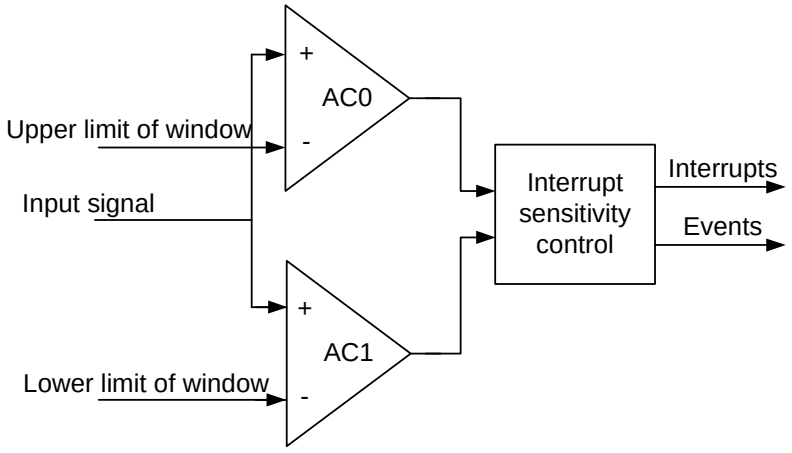
The Analog comparators have a very flexible input selection and the two comparators grouped in a pair may be used to realize a window function. One pair of analog comparators is shown in Figure 27-1 on page 45.

- **Input selection from pin**
  - **Pin 0, 1, 2, 3, 4, 5, 6 selectable to positive input of analog comparator**
  - **Pin 0, 1, 3, 5, 7 selectable to negative input of analog comparator**
- **Internal signals available on positive analog comparator inputs**
  - **Output from 12-bit DAC**
- **Internal signals available on negative analog comparator inputs**
  - **64-level scaler of the VCC, available on negative analog comparator input**
  - **Bandgap voltage reference**
- **Output from 12-bit DAC**

27.4 Window Function

The window function is realized by connecting the external inputs of the two analog comparators in a pair as shown in Figure 27-2.

Figure 27-2. Analog comparator window function



30. Pinout and Pin Functions

The pinout of XMEGA A3 is shown "" on page 2. In addition to general I/O functionality, each pin may have several function. This will depend on which peripheral is enabled and connected to the actual pin. Only one of the alternate pin functions can be used at time.

30.1 Alternate Pin Function Description

The tables below show the notation for all pin functions available and describe its function.

30.1.1 Operation/Power Supply

VCC	Digital supply voltage
AVCC	Analog supply voltage
GND	Ground

30.1.2 Port Interrupt functions

SYNC	Port pin with full synchronous and limited asynchronous interrupt function
ASYNC	Port pin with full synchronous and full asynchronous interrupt function

30.1.3 Analog functions

ACn	Analog Comparator input pin n
AC0OUT	Analog Comparator 0 Output
ADCn	Analog to Digital Converter input pin n
DACn	Digital to Analog Converter output pin n
AREF	Analog Reference input pin

30.1.4 Timer/Counter and AWEX functions

OCnx	Output Compare Channel x for Timer/Counter n
$\overline{\text{OCnx}}$	Inverted Output Compare Channel x for Timer/Counter n
OCnxLS	Output Compare Channel x Low Side for Timer/Counter n
OCnxHS	Output Compare Channel x High Side for Timer/Counter n

Table 30-3. Port C - Alternate functions

PORT C	PIN #	INTERRUPT	TCC0	AWEXC	TCC1	USARTC0	USARTC1	SPIC	TWIC	CLOCKOUT	EVENTOUT
PC0	16	SYNC	OC0A	OC0ALS					SDA		
PC1	17	SYNC	OC0B	OC0AHS		XCK0			SCL		
PC2	18	SYNC/ASYNC	OC0C	OC0BLS		RXD0					
PC3	19	SYNC	OC0D	OC0BHS		TXD0					
PC4	20	SYNC		OC0CLS	OC1A			$\overline{SS}$			
PC5	21	SYNC		OC0CHS	OC1B		XCK1	MOSI			
PC6	22	SYNC		OC0DLS			RXD1	MISO			
PC7	23	SYNC		OC0DHS			TXD1	SCK		CLKOUT	EVOUT
GND	24										
VCC	25										

Table 30-4. Port D - Alternate functions

PORT D	PIN #	INTERRUPT	TCD0	TCD1	USARTD0	USARTD1	SPID	CLOCKOUT	EVENTOUT
PD0	26	SYNC	OC0A						
PD1	27	SYNC	OC0B		XCK0				
PD2	28	SYNC/ASYNC	OC0C		RXD0				
PD3	29	SYNC	OC0D		TXD0				
PD4	30	SYNC		OC1A			$\overline{SS}$		
PD5	31	SYNC		OC1B		XCK1	MOSI		
PD6	32	SYNC				RXD1	MISO		
PD7	33	SYNC				TXD1	SCK	CLKOUT	EVOUT
GND	34								
VCC	35								

Table 30-5. Port E - Alternate functions

PORT E	PIN #	INTERRUPT	TCE0	TCE1	USARTE0	USARTE1	SPIE	TWIE	CLOCKOUT	EVENTOUT	TOSC
PE0	36	SYNC	OC0A					SDA			
PE1	37	SYNC	OC0B		XCK0			SCL			
PE2	38	SYNC/ASYNC	OC0C		RXD0						
PE3	39	SYNC	OC0D		TXD0						
PE4	40	SYNC		OC1A			$\overline{SS}$				
PE5	41	SYNC		OC1B		XCK1	MOSI				
PE6	42	SYNC				RXD1	MISO				TOSC2
PE7	43	SYNC				TXD1	SCK		CLKOUT	EVOUT	TOSC1
GND	44										
VCC	45										

Table 30-6. Port F - Alternate functions

PORT F	PIN #	INTERRUPT	TCF0	USARTF0
PF0	46	SYNC	OC0A	
PF1	47	SYNC	OC0B	XCK0
PF2	48	SYNC/ASYNC	OC0C	RXD0
PF3	49	SYNC	OC0D	TXD0
PF4	50	SYNC		
PF5	51	SYNC		
PF6	54	SYNC		
PF7	55	SYNC		
GND	52			
VCC	53			

Table 30-7. Port R- Alternate functions

PORT R	PIN #	INTERRUPT	PROGR	XTAL
PDI	56		PDI_DATA	
$\overline{\text{RESET}}$	57		PDI_CLOCK	
PRO	58	SYNC		XTAL2
PR1	59	SYNC		XTAL1

Table 30-8. ATxmega256/192/128/64A3 Boundary Scan Order

Bit Number	Signal Name	Module
149	PQ3.Bidir	PORT Q
148	PQ3.Control	
147	PQ2.Bidir	
146	PQ2.Control	
145	PQ1.Bidir	
144	PQ1.Control	
143	PQ0.Bidir	
142	PQ0.Control	
141	PK7.Bidir	PORT K
140	PK7.Control	
139	PK6.Bidir	
138	PK6.Control	
137	PK5.Bidir	
136	PK5.Control	
135	PK4.Bidir	
134	PK4.Control	
133	PK3.Bidir	
132	PK3.Control	
131	PK2.Bidir	
130	PK2.Control	
129	PK1.Bidir	
128	PK1.Control	
127	PK0.Bidir	
126	PK0.Control	

35.3 Power-down Supply Current

Figure 35-15. Power-down Supply Current vs. Temperature

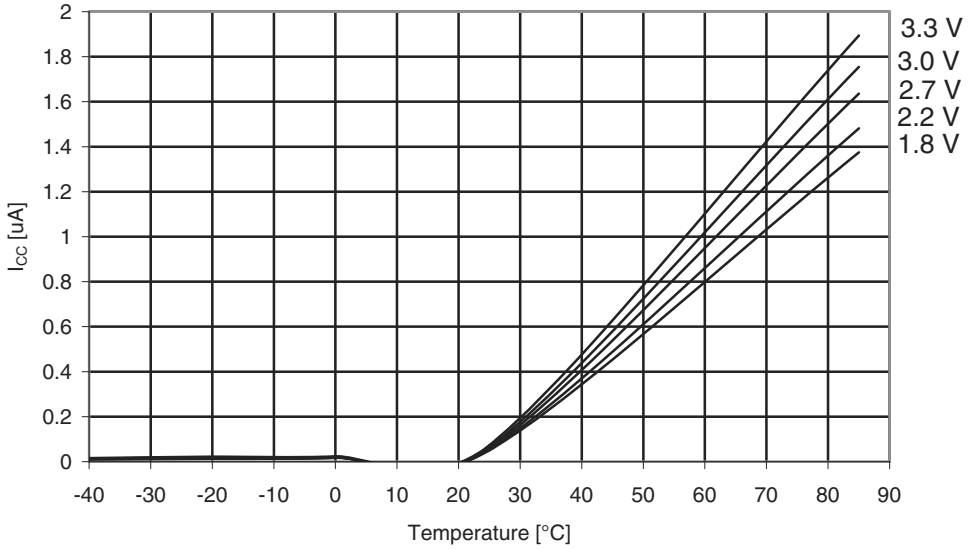
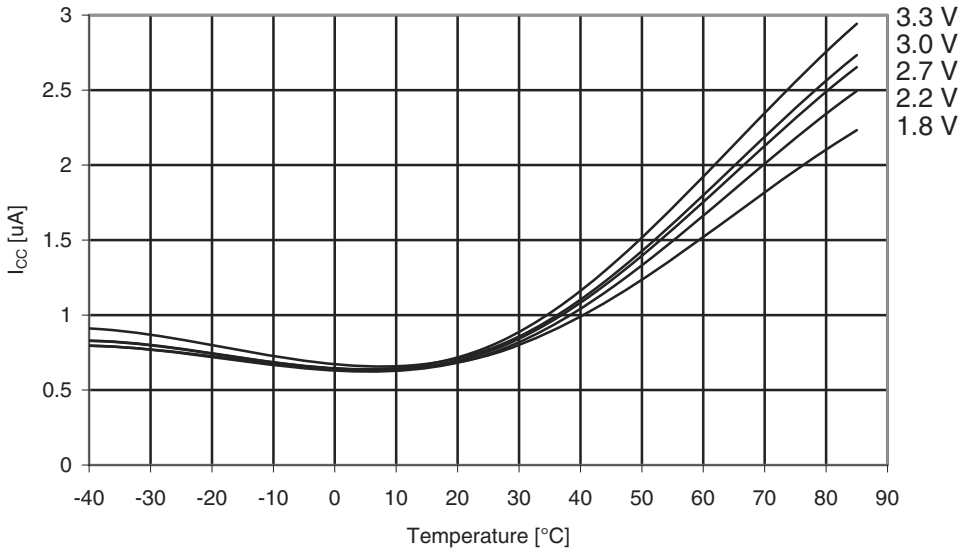


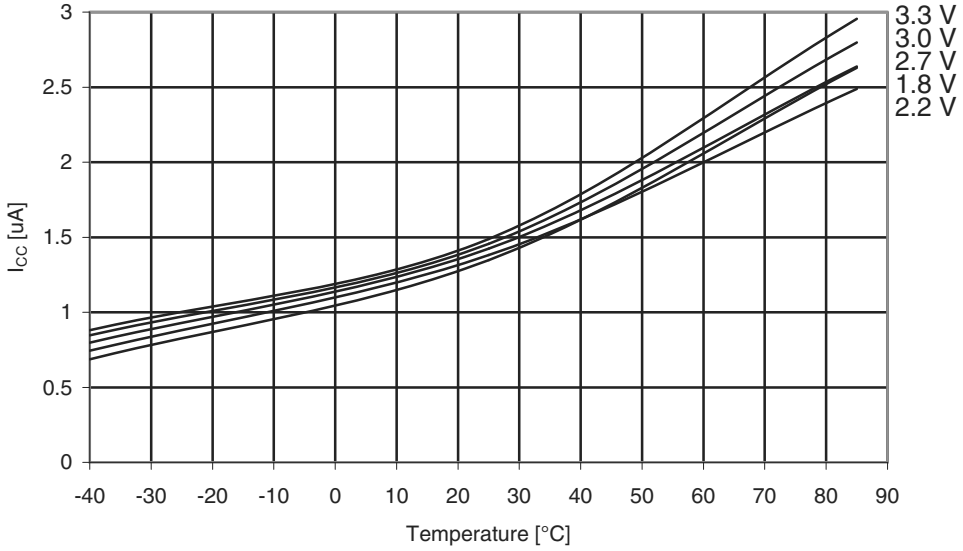
Figure 35-16. Power-down Supply Current vs. Temperature

With WDT and sampled BOD enabled.



35.4 Power-save Supply Current

Figure 35-17. Power-save Supply Current vs. Temperature
With WDT, sampled BOD and RTC from ULP enabled



35.5 Pin Pull-up

Figure 35-18. Reset Pull-up Resistor Current vs. Reset Pin Voltage
V_{CC} = 1.8V

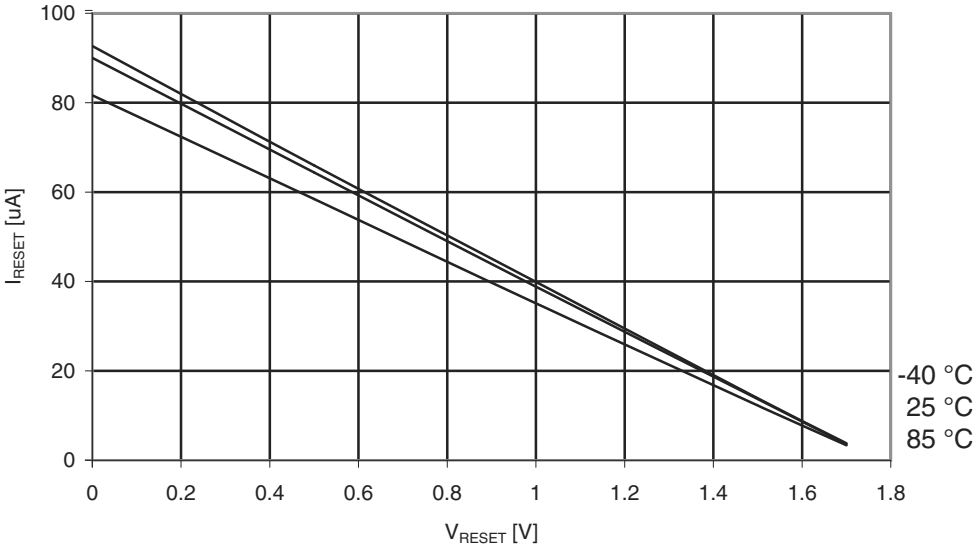
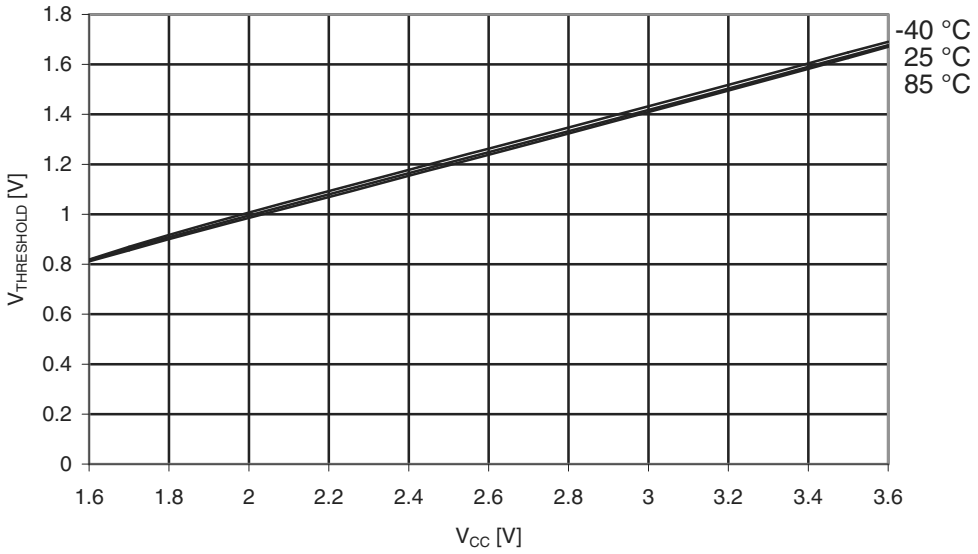


Figure 35-31. Reset Input Threshold Voltage vs. V_{CC}

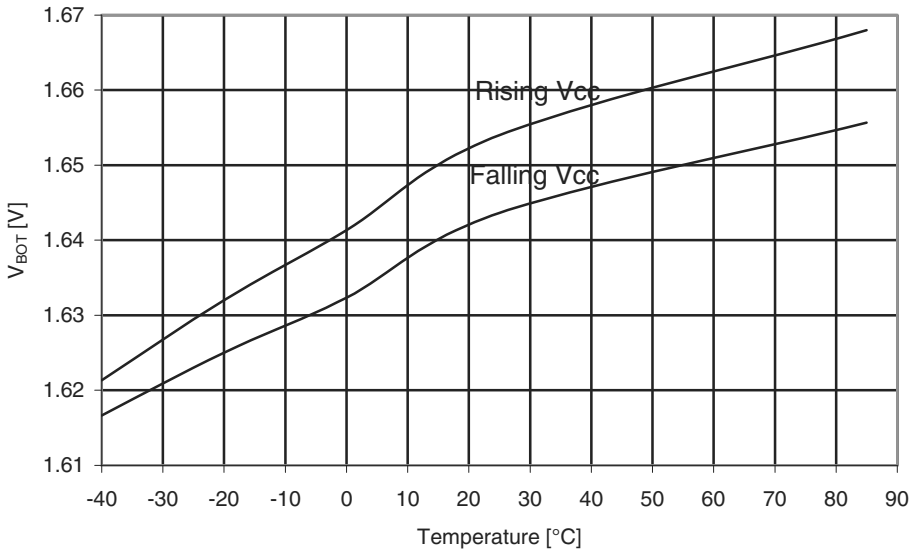
V_{IL} - I/O Pin Read as "0"



35.8 Bod Thresholds

Figure 35-32. BOD Thresholds vs. Temperature

BOD Level = 1.6V



35.10 Module current consumption

Figure 35-39. AC current consumption vs. Vcc

Low-power Mode

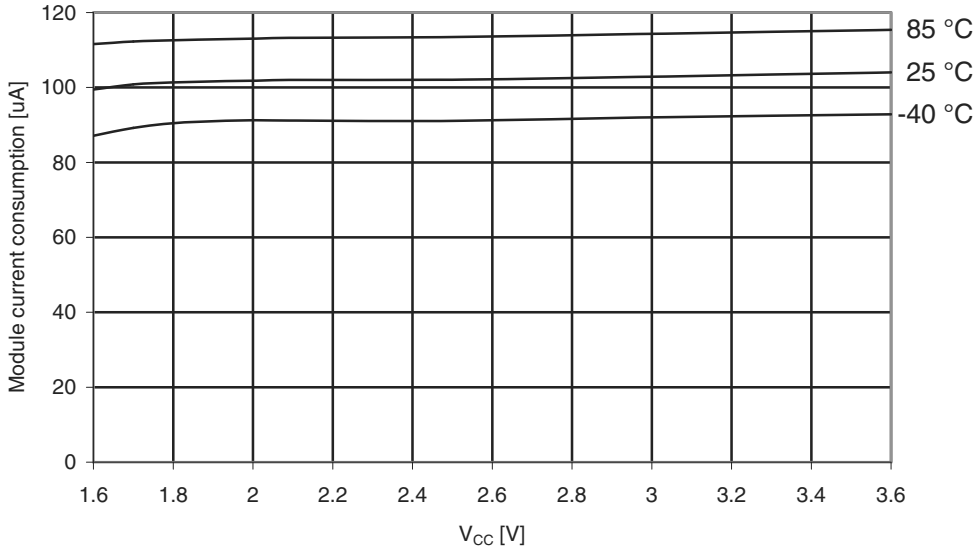
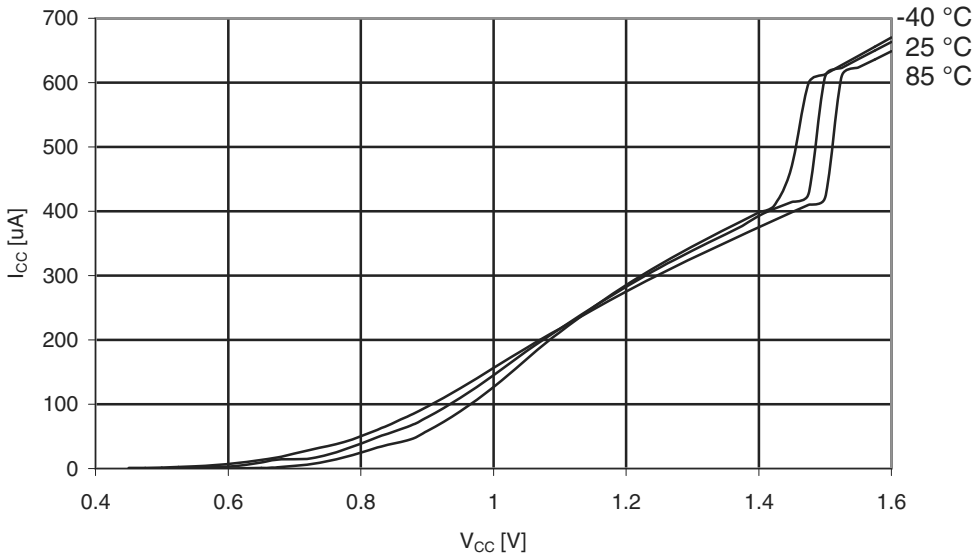


Figure 35-40. Power-up current consumption vs. Vcc



26. WDR instruction inside closed window will not issue reset

When a WDR instruction is execute within one ULP clock cycle after updating the window control register, the counter can be cleared without giving a system reset.

Problem fix/Workaround

Wait at least one ULP clock cycle before executing a WDR instruction.

Problem fix/Workaround

Wait at least one prescaled RTC clock cycle before reading the RTC CNT value.

22. Pending asynchronous RTC-interrupts will not wake up device

Asynchronous Interrupts from the Real-Time-Counter that is pending when the sleep instruction is executed, will be ignored until the device is woken from another source or the source triggers again.

Problem fix/Workaround

None.

23. TWI Transmit collision flag not cleared on repeated start

The TWI transmit collision flag should be automatically cleared on start and repeated start, but is only cleared on start.

Problem fix/Workaround

Clear the flag in software after address interrupt.

24. Clearing TWI Stop Interrupt Flag may lock the bus

If software clears the STOP Interrupt Flag (APIF) on the same Peripheral Clock cycle as the hardware sets this flag due to a new address received, CLKHOLD is not cleared and the SCL line is not released. This will lock the bus.

Problem fix/Workaround

Check if the bus state is IDLE. If this is the case, it is safe to clear APIF. If the bus state is not IDLE, wait for the SCL pin to be low before clearing APIF.

Code:

```
/* Only clear the interrupt flag if within a "safe zone". */
while ( /* Bus not IDLE: */
       ((COMMS_TWI.MASTER.STATUS & TWI_MASTER_BUSSTATE_gm) !=
         TWI_MASTER_BUSSTATE_IDLE_gc)) &&
         /* SCL not held by slave: */
         !(COMMS_TWI.SLAVE.STATUS & TWI_SLAVE_CLKHOLD_bm)
      )
{
    /* Ensure that the SCL line is low */
    if ( !(COMMS_PORT.IN & PIN1_bm) )
        if ( !(COMMS_PORT.IN & PIN1_bm) )
            break;
}
/* Check for an pending address match interrupt */
if ( !(COMMS_TWI.SLAVE.STATUS & TWI_SLAVE_CLKHOLD_bm) )
{
    /* Safely clear interrupt flag */
    COMMS_TWI.SLAVE.STATUS |= (uint8_t)TWI_SLAVE_APIF_bm;
}
```

25. TWI START condition at bus timeout will cause transaction to be dropped

If Bus Timeout is enabled and a timeout occurs on the same Peripheral Clock cycle as a START is detected, the transaction will be dropped.