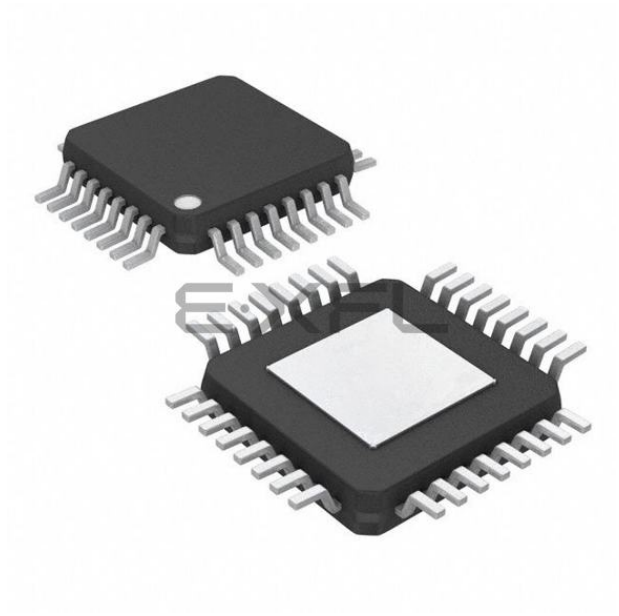


Welcome to E-XFL.COM

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	26
Program Memory Size	16KB (8K x 16)
Program Memory Type	FLASH
EEPROM Size	512 x 8
RAM Size	2K 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 ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	32-TQFP
Supplier Device Package	32-TQFP (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atxmega16e5-anr

7. CPU

7.1 Features

- 8/16-bit, high-performance Atmel AVR RISC CPU
- 142 instructions
- Hardware multiplier
- 32x8-bit registers directly connected to the ALU
- Stack in RAM
- Stack pointer accessible in I/O memory space
- Direct addressing of up to 16MB of program memory and 16MB of data memory
- True 16/24-bit access to 16/24-bit I/O registers
- Efficient support for 8-, 16-, and 32-bit arithmetic
- Configuration change protection of system-critical features

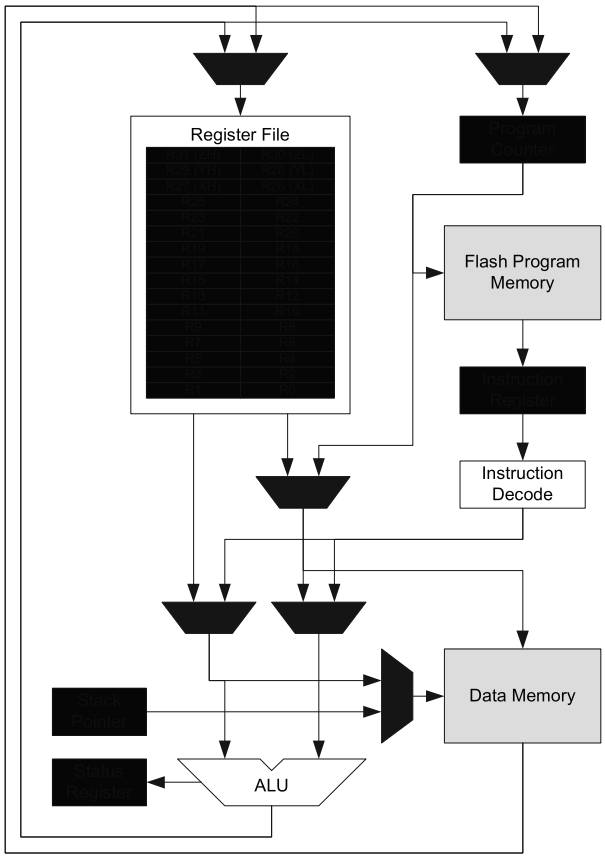
7.2 Overview

All AVR XMEGA devices use the 8/16-bit AVR CPU. The main function of the CPU is to execute the code and perform all calculations. The CPU is able to access memories, perform calculations, control peripherals, and execute the program in the flash memory. Interrupt handling is described in a separate section, refer to “Interrupts and Programmable Multilevel Interrupt Controller” on page 28.

7.3 Architectural Overview

In order to maximize performance and parallelism, the AVR CPU uses a Harvard architecture with separate memories and buses for program and data. Instructions in the program memory are executed with single-level pipelining. While one instruction is being executed, the next instruction is pre-fetched from the program memory. This enables instructions to be executed on every clock cycle. For details of all AVR instructions, refer to http://www.atmel.com/avr.

Figure 7-1. Block Diagram of the AVR CPU Architecture



The arithmetic logic unit (ALU) supports arithmetic and logic operations between registers or between a constant and a register. Single-register operations can also be executed in the ALU. After an arithmetic operation, the status register is updated to reflect information about the result of the operation.

The ALU is directly connected to the fast-access register file. The 32 x 8-bit general purpose working registers all have single clock cycle access time allowing single-cycle arithmetic logic unit (ALU) operation between registers or between a register and an immediate. Six of the 32 registers can be used as three 16-bit address pointers for program and data space addressing, enabling efficient address calculations.

The memory spaces are linear. The data memory space and the program memory space are two different memory spaces.

The data memory space is divided into I/O registers, SRAM, and memory mapped EEPROM.

All I/O status and control registers reside in the lowest 4KB addresses of the data memory. This is referred to as the I/O memory space. The lowest 64 addresses can be accessed directly, or as the data space locations from 0x00 to 0x3F. The rest is the extended I/O memory space, ranging from 0x0040 to 0x0FFF. I/O registers here must be accessed as data space locations using load (LD/LDS/LDD) and store (ST/STS/STD) instructions.

The SRAM holds data. Code execution from SRAM is not supported. It can easily be accessed through the five different addressing modes supported in the AVR architecture. The first SRAM address is 0x2000.

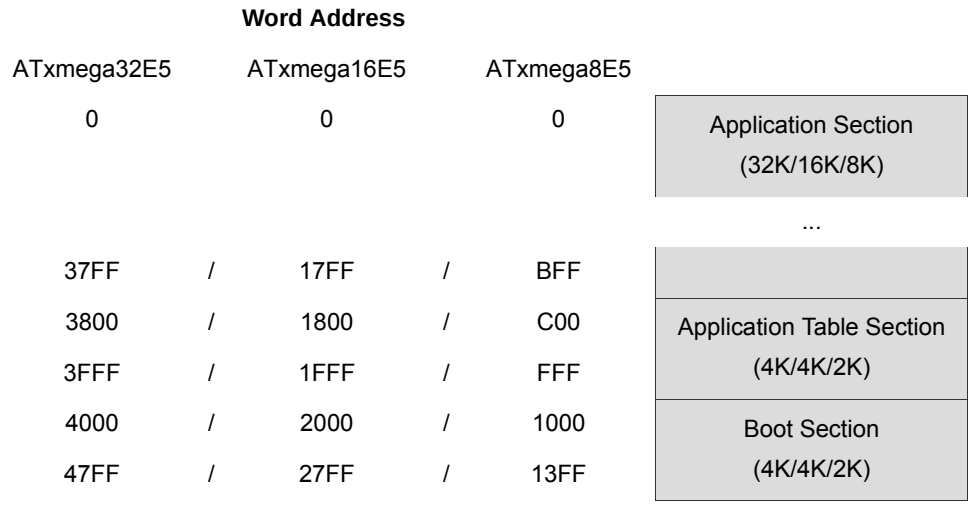
Data addresses 0x1000 to 0x1FFF are reserved for EEPROM.

The program memory is divided in two sections, the application program section and the boot program section. Both sections have dedicated lock bits for write and read/write protection. The SPM instruction that is used for self-programming of the application flash memory must reside in the boot program section. The application section contains an application table section with separate lock bits for write and read/write protection. The application table section can be used for save storing of nonvolatile data in the program memory.

All AVR CPU instructions are 16 or 32 bits wide, and each flash location is 16 bits wide. The flash memory is organized in two main sections, the application section and the boot loader section. The sizes of the different sections are fixed, but device-dependent. These two sections have separate lock bits, and can have different levels of protection. The store program memory (SPM) instruction, which is used to write to the flash from the application software, will only operate when executed from the boot loader section.

The application section contains an application table section with separate lock settings. This enables safe storage of nonvolatile data in the program memory.

Figure 8-1. Flash Program Memory (hexadecimal address)



8.3.1 Application Section

The Application section is the section of the flash that is used for storing the executable application code. The protection level for the application section can be selected by the boot lock bits for this section. The application section can not store any boot loader code since the SPM instruction cannot be executed from the application section.

8.3.2 Application Table Section

The application table section is a part of the application section of the flash memory that can be used for storing data. The size is identical to the boot loader section. The protection level for the application table section can be selected by the boot lock bits for this section. The possibilities for different protection levels on the application section and the application table section enable safe parameter storage in the program memory. If this section is not used for data, application code can reside here.

8.3.3 Boot Loader Section

While the application section is used for storing the application code, the boot loader software must be located in the boot loader section because the SPM instruction can only initiate programming when executing from this section. When programming, the CPU is halted, waiting for the flash operation to complete. The SPM instruction can access the entire flash, including the boot loader section itself. The protection level for the boot loader section can be selected by the boot loader lock bits. If this section is not used for boot loader software, application code can be stored here.

8.3.4 Production Signature Row

The production signature row is a separate memory section for factory programmed data. It contains calibration data for functions such as oscillators and analog modules. Some of the calibration values will be automatically loaded to the corresponding module or peripheral unit during reset. Other values must be loaded from the signature row and written to

12. Power Management and Sleep Modes

12.1 Features

- Power management for adjusting power consumption and functions
- Five sleep modes
  - Idle
  - Power down
  - Power save
  - Standby
  - Extended standby
- Power reduction register to disable clock and turn off unused peripherals in active and idle modes

12.2 Overview

Various sleep modes and clock gating are provided in order to tailor power consumption to application requirements. This enables the Atmel AVR XMEGA microcontroller to stop unused modules to save power.

All sleep modes are available and can be entered from active mode. In active mode, the CPU is executing application code. When the device enters sleep mode, program execution is stopped and interrupts or a reset is used to wake the device again. The application code decides which sleep mode to enter and when. Interrupts from enabled peripherals and all enabled reset sources can restore the microcontroller from sleep to active mode.

In addition, power reduction registers provide a method to stop the clock to individual peripherals from software. When this is done, the current state of the peripheral is frozen, and there is no power consumption from that peripheral. This reduces the power consumption in active mode and idle sleep modes and enables much more fine-tuned power management than sleep modes alone.

12.3 Sleep Modes

Sleep modes are used to shut down modules and clock domains in the microcontroller in order to save power. XMEGA microcontrollers have five different sleep modes tuned to match the typical functional stages during application execution. A dedicated sleep instruction (SLEEP) is available to enter sleep mode. Interrupts are used to wake the device from sleep, and the available interrupt wake-up sources are dependent on the configured sleep mode. When an enabled interrupt occurs, the device will wake up and execute the interrupt service routine before continuing normal program execution from the first instruction after the SLEEP instruction. If other, higher priority interrupts are pending when the wake-up occurs, their interrupt service routines will be executed according to their priority before the interrupt service routine for the wake-up interrupt is executed. After wake-up, the CPU is halted for four cycles before execution starts.

The content of the register file, SRAM and registers are kept during sleep. If a reset occurs during sleep, the device will reset, start up, and execute from the reset vector.

12.3.1 Idle Mode

In idle mode the CPU and nonvolatile memory are stopped (note that any ongoing programming will be completed), but all peripherals, including the interrupt controller, event system and EDMA controller are kept running. Any enabled interrupt will wake the device.

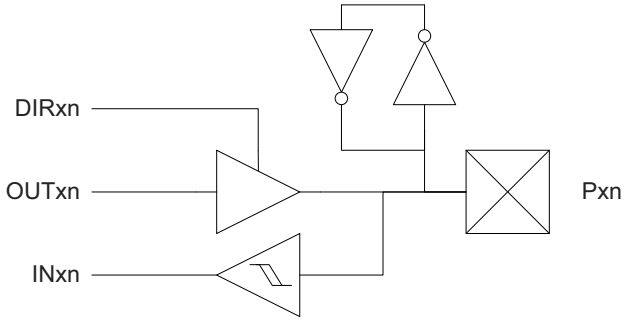
12.3.2 Power-down Mode

In power-down mode, all clocks, including the real-time counter clock source, are stopped. This allows operation only of asynchronous modules that do not require a running clock. The only interrupts that can wake up the MCU are the two-wire interface address match interrupt and asynchronous port interrupts.

16.3.4 Bus-keeper

The bus-keeper's weak output produces the same logical level as the last output level. It acts as a pull-up if the last level was '1', and pull-down if the last level was '0'.

Figure 16-4. I/O Configuration - Totem-pole with Bus-keeper



16.3.5 Others

Figure 16-5. Output Configuration - Wired-OR with Optional Pull-down

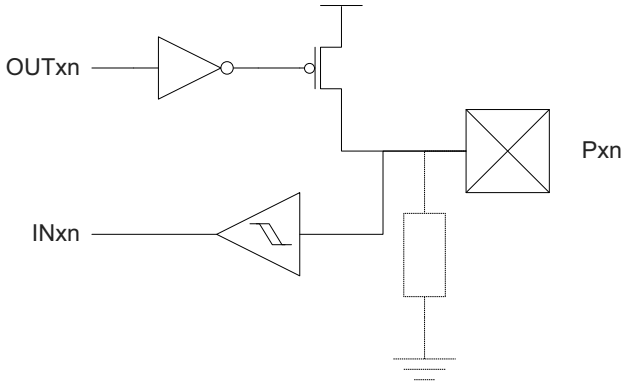
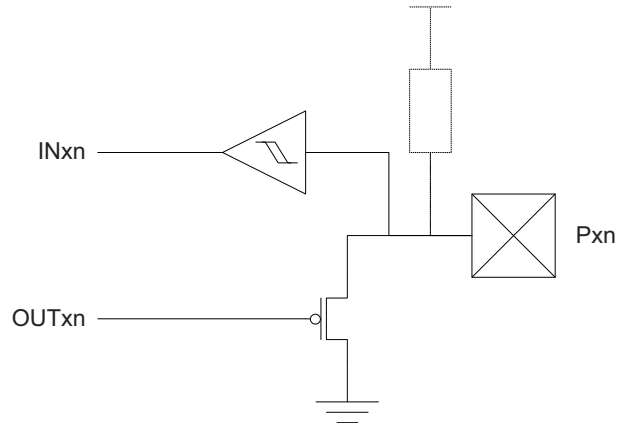


Figure 16-6. I/O Configuration - Wired-AND with Optional Pull-up

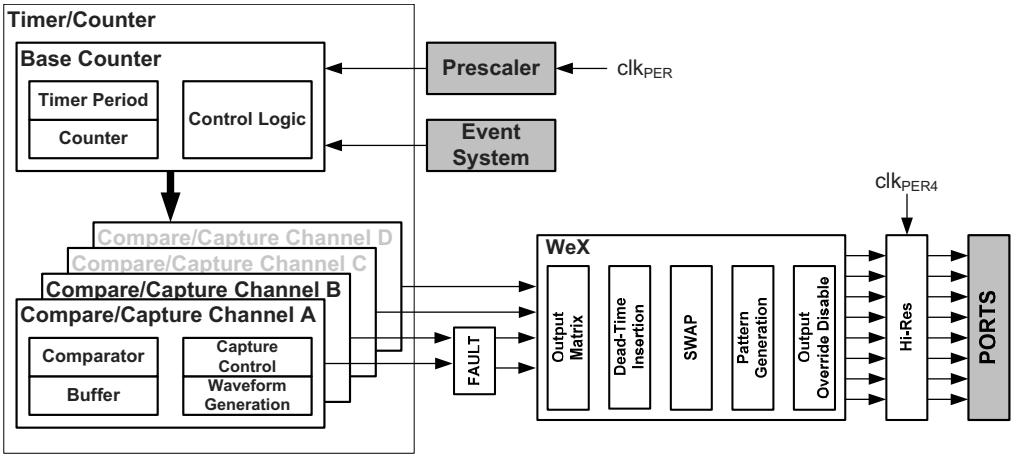


There are two differences between timer/counter type 4 and type 5. Timer/counter 4 has four CC channels, and timer/counter 5 has two CC channels. Both timer/counter 4 and 5 can be set in 8-bit mode, allowing the application to double the number of compare and capture channels that then get 8-bit resolution.

Some timer/counters have extensions that enable more specialized waveform generation. The waveform extension (WeX) is intended for motor control, ballast, LED, H-bridge, power converters, and other types of power control applications. It enables more customized waveform output distribution, and low- and high-side channel output with optional dead-time insertion. It can also generate a synchronized bit pattern across the port pins. The high-resolution (hi-res) extension can increase the waveform resolution by four or eight times by using an internal clock source four times faster than the peripheral clock. The fault extension (FAULT) enables fault protection for safe and deterministic handling, disabling and/or shut down of external drivers.

A block diagram of the 16-bit timer/counter with extensions and closely related peripheral modules (in grey) is shown in Figure 17-1.

Figure 17-1. 16-bit Timer/counter and Closely Related Peripherals



PORTC has one timer/counter 4 and one timer/counter 5. PORTD has one timer/counter 5. Notation of these are TCC4 (timer/counter C4), TCC5, and TCD5, respectively.

18. WeX – Waveform Extension

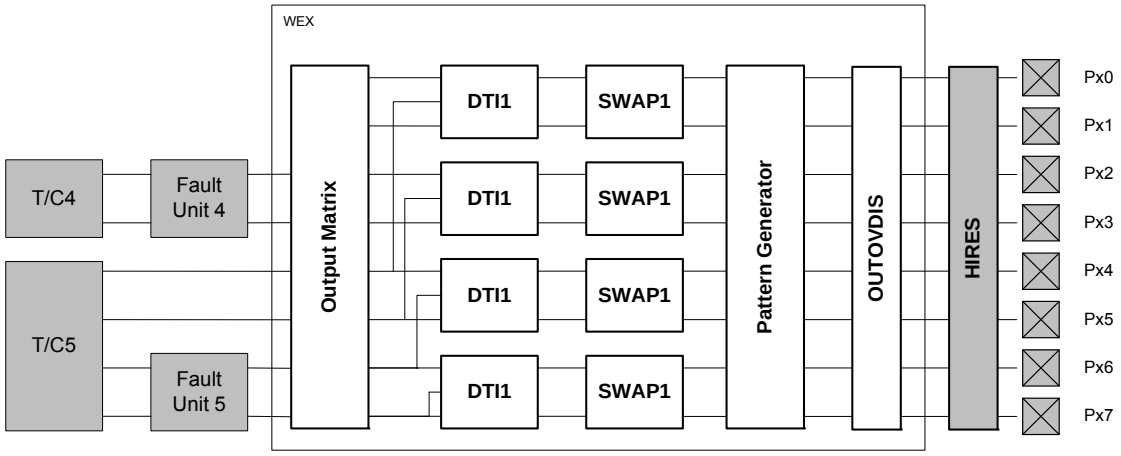
18.1 Features

- Module for more customized and advanced waveform generation
  - Optimized for various type of motor, ballast, and power stage control
- Output matrix for timer/counter waveform output distribution
  - Configurable distribution of compare channel output across port pins
  - Redistribution of dead-time insertion resource between TC4 and TC5
- Four dead-time insertion (DTI) units, each with
  - Complementary high and low side with non overlapping outputs
  - Separate dead-time setting for high and low side
  - 8-bit resolution
- Four swap (SWAP) units
  - Separate port pair or low high side drivers swap
  - Double buffered swap feature
- Pattern generation creating synchronized bit pattern across the port pins
  - Double buffered pattern generation

18.2 Overview

The waveform extension (WEX) provides extra functions to the timer/counter in waveform generation (WG) modes. It is primarily intended for motor control, ballast, LED, H-bridge, power converters, and other types of power control applications. The WEX consist of five independent and successive units, as shown in Figure 18-1.

Figure 18-1. Waveform Extension and Closely Related Peripherals



The output matrix (OTMX) can distribute and route out the waveform outputs from timer/counter 4 and 5 across the port pins in different configurations, each optimized for different application types. The dead time insertion (DTI) unit splits the four lower OTMX outputs into a two non-overlapping signals, the non-inverted low side (LS) and inverted high side (HS) of the waveform output with optional dead-time insertion between LS and HS switching.

The swap (SWAP) unit can swap the LS and HS pin position. This can be used for fast decay motor control. The pattern generation unit generates synchronized output waveform with constant logic level. This can be used for easy stepper motor and full bridge control.

The output override disable unit can disable the waveform output on selectable port pins to optimize the pins usage. This is to free the pins for other functional use, when the application does not need the waveform output spread across all the port pins as they can be selected by the OTMX configurations.

The waveform extension is available for TCC4 and TCC5. The notation of this is WEXC.

21. RTC – 16-bit Real-Time Counter

21.1 Features

- 16-bit resolution
- Selectable clock source
  - 32.768kHz external crystal
  - External clock
  - 32.768kHz internal oscillator
  - 32kHz internal ULP oscillator
- Programmable 10-bit clock prescaling
- One compare register
- One period register
- Clear counter on period overflow
- Optional interrupt/event on overflow and compare match
- Correction for external crystal oscillator frequency error down to ±0.5ppm accuracy

21.2 Overview

The 16-bit real-time counter (RTC) is a counter that typically runs continuously, including in low power sleep modes, to keep track of time. It can wake up the device from sleep modes and/or interrupt the device at regular intervals.

The reference clock is typically the 1.024kHz output from a high-accuracy crystal of 32.768kHz, and this is the configuration most optimized for low power consumption. The faster 32.768kHz output can be selected if the RTC needs a resolution higher than 1ms. The RTC can also be clocked from an external clock signal, the 32.768kHz internal oscillator or the 32kHz internal ULP oscillator.

The RTC includes a 10-bit programmable prescaler that can scale down the reference clock before it reaches the counter. A wide range of resolutions and time-out periods can be configured. With a 32.768kHz clock source, the maximum resolution is 30.5µs, and time-out periods can range up to 2000 seconds. With a resolution of 1s, the maximum timeout period is more than 18 hours (65536 seconds). The RTC can give a compare interrupt and/or event when the counter equals the compare register value, and an overflow interrupt and/or event when it equals the period register value.

25. IRCOM – IR Communication Module

25.1 Features

- Pulse modulation/demodulation for infrared communication
- IrDA compatible for baud rates up to 115.2Kbps
- Selectable pulse modulation scheme
  - 3/16 of the baud rate period
  - Fixed pulse period, 8-bit programmable
  - Pulse modulation disabled
- Built-in filtering
- Can be connected to and used by any USART

25.2 Overview

Atmel AVR XMEGA devices contain an infrared communication module (IRCOM) that is IrDA compatible for baud rates up to 115.2Kbps. It can be connected to any USART to enable infrared pulse encoding/decoding for that USART.

26. XCL – XMEGA Custom Logic Module

26.1 Features

- Two independent 8-bit timer/counter with:
  - Period and compare channel for each timer/counter
  - Input Capture for each timer
  - Serial peripheral data length control for each timer
  - Timeout support for each timer
  - Timer underflow interrupt/event
  - Compare match or input capture interrupt/event for each timer
- One 16-bit timer/counter by cascading two 8-bit timer/counters with:
  - Period and compare channel
  - Input capture
  - Timeout support
  - Timer underflow interrupt/event
  - Compare match or input capture interrupt/event
- Programmable lookup table supporting multiple configurations:
  - Two 2-input units
  - One 3-input unit
  - RS configuration
  - Duplicate input with selectable delay on one input or output
  - Connection to external I/O pins, event system or one selectable USART
- Combinatorial Logic Functions using programmable truth table:
  - AND, NAND, OR, NOR, XOR, XNOR, NOT, MUX
- Sequential Logic Functions:
  - D-Flip-Flop, D Latch, RS Latch
- Input sources:
  - From external pins or the event system
  - One input source includes selectable delay or synchronizing option
  - Can be shared with selectable USART pin locations
- Outputs:
  - Available on external pins or event system
  - Includes selectable delay or synchronizing option
  - Can override selectable USART pin locations
- Operates in active mode and all sleep modes

26.2 Overview

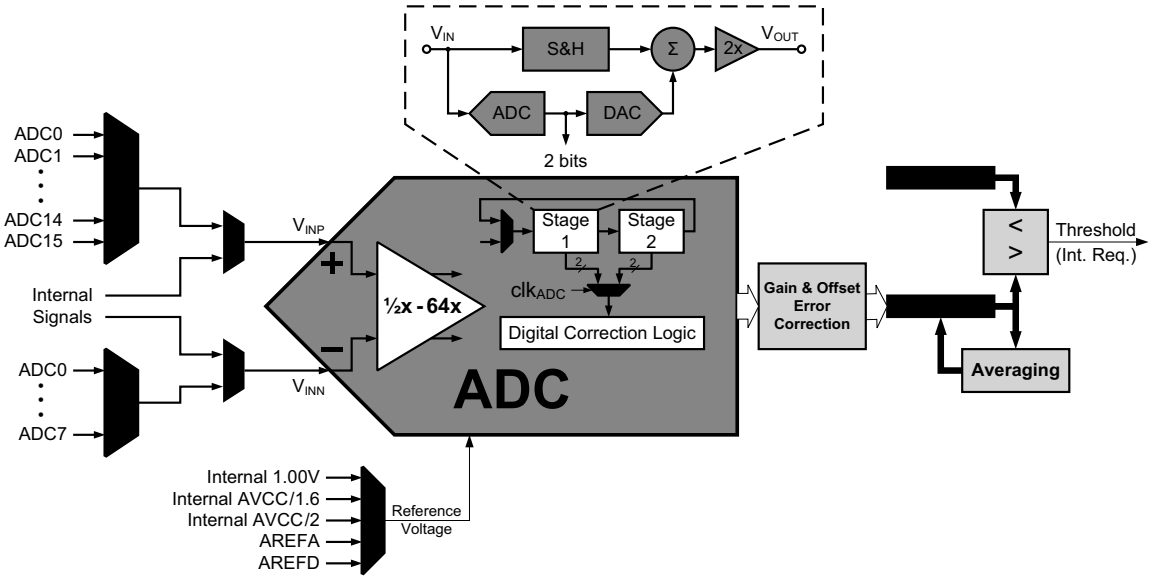
The XMEGA Custom Logic module (XCL) consists of two sub-units, each including 8-bit timer/counter with flexible settings, peripheral counter working with one software selectable USART module, delay elements, glue logic with programmable truth table and a global logic interconnect array.

The timer/counter configuration allows for two 8-bits timer/counters. Each timer/counter supports normal, compare and input capture operation, with common flexible clock selections and event channels for each timer. By cascading the two 8-bit timer/counters, the XCL can be used as a 16-bit timer/counter.

The peripheral counter (PEC) configuration, the XCL is connected to one software selectable USART. This USART controls the counter operation, and the PEC can optionally control the data length within the USART frame.

The glue logic configuration, the XCL implements two programmable lookup tables (LUTs). Each defines the truth table corresponding to the logical condition between two inputs. Any combinatorial function logic is possible. The LUT inputs can be connected to I/O pins or event system channels. If the LUT is connected to the USART0 pin locations, the data lines (TXD/RXD) data encoding/decoding will be possible. Connecting together the LUT units, RS Latch, or any combinatorial logic between two operands or three inputs can be enabled.

Figure 28-1. ADC Overview



The ADC may be configured for 8- or 12-bit result, reducing the propagation delay from 3.35µs for 12-bit to 2.3µs for 8-bit result. ADC conversion results are provided left- or right adjusted with eases calculation when the result is represented as a signed.

PORTA has one ADC. Notation of this peripheral is ADCA.

XCKn	Transfer Clock for USART n
RXDn	Receiver Data for USART n
TXDn	Transmitter Data for USART n
SS	Slave Select for SPI
MOSI	Master Out Slave In for SPI
MISO	Master In Slave Out for SPI
SCK	Serial Clock for SPI

32.1.6 Oscillators, Clock, and Event

TOSCn	Timer Oscillator pin n
XTALn	Input/Output for Oscillator pin n
CLKOUT	Peripheral Clock Output
EVOUT	Event Channel Output
RTCOUT	RTC Clock Source Output

32.1.7 Debug/System Functions

RESET	Reset pin
PDI_CLK	Program and Debug Interface Clock pin
PDI_DATA	Program and Debug Interface Data pin

Table 32-4. PORT R – Alternate Functions

PORT R	Pin #	XTAL	TOSC	EXTCLK	CLOCKOUT	EVENTOUT	RTCOUT	AC OUT
PR0	20	XTAL2	TOSC2		CLKOUT	EVOUT	RTCOUT	AC1 OUT
PR1	19	XTAL1	TOSC1	EXTCLK				AC0 OUT

Table 32-5. PORT D – Alternate Functions

PORT D	Pin #	ADCAPOS GAINPOS	TCD5	USART D0	TWID (Bridge)	XCL (LUT)	XCL (TC)	CLOCK OUT	EVENT OUT	RTCOUT	ACOUT	REFD
PD0	28	ADC8			SDA	IN1/ OUT0						AREF
PD1	27	ADC9		XCK0	SCL	IN2						
PD2	26	ADC10		RXD0		IN0	OC0					
PD3	25	ADC11		TXD0		IN3	OC1					
PD4	24	ADC12	OC5A			IN1/ OUT0		CLKOUT	EVOUT			
PD5	23	ADC13	OC5B	XCK0		IN2						
PD6	22	ADC14		RXD0		IN0				RTCOUT	AC1OUT	
PD7	21	ADC15		TXD0		IN3		CLKOUT	EVOUT		AC0OUT	

33. Peripheral Module Address Map

The address maps show the base address for each peripheral and module in XMEGA E5. For complete register description and summary for each peripheral module, refer to the XMEGA E Manual.

Table 33-1. Peripheral Module Address Map

Base Address	Name	Description
0x0000	GPIO	General Purpose IO Registers
0x0010	VPORT0	Virtual Port A
0x0014	VPORT1	Virtual Port C
0x0018	VPORT2	Virtual Port D
0x001C	VPORT3	Virtual Port R
0x0030	CPU	CPU
0x0040	CLK	Clock Control
0x0048	SLEEP	Sleep Controller
0x0050	OSC	Oscillator Control
0x0060	DFLLRC32M	DFLL for the 32MHz Internal Oscillator
0x0070	PR	Power Reduction
0x0078	RST	Reset Controller
0x0080	WDT	Watch-Dog Timer
0x0090	MCU	MCU Control
0x00A0	PMIC	Programmable Multilevel Interrupt Controller
0x00B0	PORTCFG	Port Configuration
0x00D0	CRC	CRC Module
0x0100	EDMA	Enhanced DMA Controller
0x0180	EVSYS	Event System
0x01C0	NVM	Non Volatile Memory (NVM) Controller
0x0200	ADCA	Analog to Digital Converter on port A
0x0300	DACA	Digital to Analog Converter on port A
0x0380	ACA	Analog Comparator pair on port A
0x0400	RTC	Real Time Counter
0x0460	XCL	XMEGA Custom Logic Module
0x0480	TWIC	Two-Wire Interface on port C
0x0600	PORTA	Port A
0x0640	PORTC	Port C
0x0660	PORTD	Port D

Base Address	Name	Description
0x07E0	PORTR	Port R
0x0800	TCC4	Timer/Counter 4 on port C
0x0840	TCC5	Timer/Counter 5 on port C
0x0880	FAULTC4	Fault Extension on TCC4
0x0890	FAULTC5	Fault Extensionon TCC5
0x08A0	WEXC	Waveform Extension on port C
0x08B0	HIRESC	High Resolution Extension on port C
0x08C0	USARTC0	USART 0 on port C
0x08E0	SPIC	Serial Peripheral Interface on port C
0x08F8	IRCOM	Infrared Communication Module
0x0940	TCD5	Timer/Counter 5 on port D
0x09C0	USARTD0	USART 0 on port D

Mnemonics	Operands	Description	Operation	Flags	#Clocks
LDS	Rd, k	Load Direct from data space	Rd ← (k)	None	2[1][2]
LD	Rd, X	Load Indirect	Rd ← (X)	None	1[1][2]
LD	Rd, X+	Load Indirect and Post-Increment	Rd ← (X) X ← X + 1	None	1[1][2]
LD	Rd, -X	Load Indirect and Pre-Decrement	X ← X - 1, ← X - 1 Rd ← (X) ← (X)	None	2[1][2]
LD	Rd, Y	Load Indirect	Rd ← (Y) ← (Y)	None	1[1][2]
LD	Rd, Y+	Load Indirect and Post-Increment	Rd ← (Y) Y ← Y + 1	None	1[1][2]
LD	Rd, -Y	Load Indirect and Pre-Decrement	Y ← Y - 1 Rd ← (Y)	None	2[1][2]
LDD	Rd, Y+q	Load Indirect with Displacement	Rd ← (Y + q)	None	2[1][2]
LD	Rd, Z	Load Indirect	Rd ← (Z)	None	1[1][2]
LD	Rd, Z+	Load Indirect and Post-Increment	Rd ← (Z), Z ← Z+1	None	1[1][2]
LD	Rd, -Z	Load Indirect and Pre-Decrement	Z ← Z - 1, Rd ← (Z)	None	2[1][2]
LDD	Rd, Z+q	Load Indirect with Displacement	Rd ← (Z + q)	None	2[1][2]
STS	k, Rr	Store Direct to Data Space	(k) ← Rd	None	2[1]
ST	X, Rr	Store Indirect	(X) ← Rr	None	1[1]
ST	X+, Rr	Store Indirect and Post-Increment	(X) ← Rr, X ← X + 1	None	1[1]
ST	-X, Rr	Store Indirect and Pre-Decrement	X ← X - 1, (X) ← Rr	None	2[1]
ST	Y, Rr	Store Indirect	(Y) ← Rr	None	1[1]
ST	Y+, Rr	Store Indirect and Post-Increment	(Y) ← Rr, Y ← Y + 1	None	1[1]
ST	-Y, Rr	Store Indirect and Pre-Decrement	Y ← Y - 1, (Y) ← Rr	None	2[1]
STD	Y+q, Rr	Store Indirect with Displacement	(Y + q) ← Rr	None	2[1]
ST	Z, Rr	Store Indirect	(Z) ← Rr	None	1[1]
ST	Z+, Rr	Store Indirect and Post-Increment	(Z) ← Rr Z ← Z + 1	None	1[1]
ST	-Z, Rr	Store Indirect and Pre-Decrement	Z ← Z - 1	None	2[1]
STD	Z+q,Rr	Store Indirect with Displacement	(Z + q) ← Rr	None	2[1]
LPM		Load Program Memory	R0 ← (Z)	None	3
LPM	Rd, Z	Load Program Memory	Rd ← (Z)	None	3
LPM	Rd, Z+	Load Program Memory and Post-Increment	Rd ← (Z), Z ← Z + 1	None	3
ELPM		Extended Load Program Memory	R0 ← (RAMPZ:Z)	None	3
ELPM	Rd, Z	Extended Load Program Memory	Rd ← (RAMPZ:Z)	None	3
ELPM	Rd, Z+	Extended Load Program Memory and Post-Increment	Rd ← (RAMPZ:Z), Z ← Z + 1	None	3
SPM		Store Program Memory	(RAMPZ:Z) ← R1:R0	None	-

Symbol	Parameter	Condition	Min.	Typ.	Max.	Units
C_{XTAL1}	Parasitic capacitance XTAL1 pin			5.4		pF
C_{XTAL2}	Parasitic capacitance XTAL2 pin			7.1		
C_{LOAD}	Parasitic capacitance load			3.07		

Note: 1. Numbers for negative impedance are not tested in production but guaranteed from design and characterization.

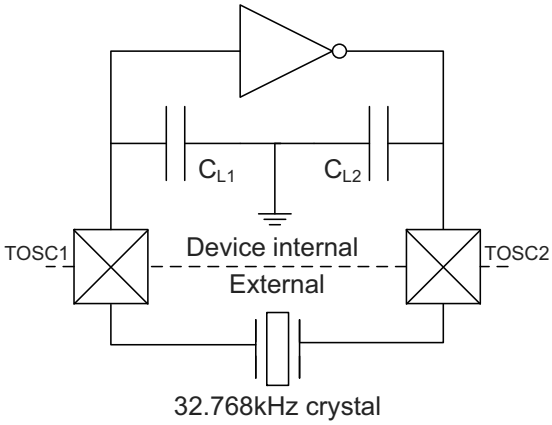
36.13.8 External 32.768kHz Crystal Oscillator and TOSC Characteristics

Table 36-28. External 32.768kHz Crystal Oscillator and TOSC Characteristics

Symbol	Parameter	Condition	Min.	Typ.	Max.	Units
ESR/R1	Recommended crystal equivalent series resistance (ESR)	Crystal load capacitance 6.5pF			60	kΩ
		Crystal load capacitance 9.0pF			35	
C_{TOSC1}	Parasitic capacitance TOSC1 pin			5.3		pF
C_{TOSC2}	Parasitic capacitance TOSC2 pin			7.4		
	Recommended safety factor	capacitance load matched to crystal specification	3.0			

Note: 1. See Figure 36-4 for definition.

Figure 36-4. TOSC Input Capacitance



The parasitic capacitance between the TOSC pins is C_{L1} + C_{L2} in series as seen from the crystal when oscillating without external capacitors.

36.15 Two-Wire Interface Characteristics

Table 36-6 on page 76 describes the requirements for devices connected to the two-wire interface (TWI) Bus. The Atmel AVR XMEGA TWI meets or exceeds these requirements under the noted conditions. Timing symbols refer to Figure 36-7.

Figure 36-7. Two-wire Interface Bus Timing

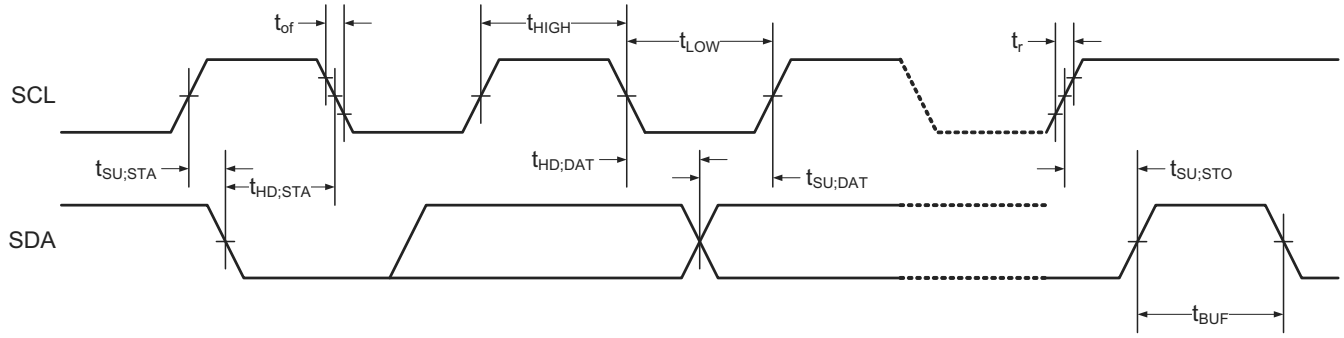


Table 36-30. Two-wire Interface Characteristics

Symbol	Parameter	Condition		Min.	Typ.	Max.	Units
V_{IH}	Input high voltage			$0.7V_{CC}$		V_{CC}+0.5	V
V_{IL}	Input low voltage			-0.5		$0.3V_{CC}$	
V_{hys}	Hysteresis of Schmitt trigger inputs			$0.05V_{CC}$(1)			
V_{OL}	Output low voltage	3mA, sink current		0		0.4	
I_{OL}	Low level output current	f_{SCL} ≤ 400kHz	V_{OL} = 0.4V	3			mA
		f_{SCL} ≤ 1MHz		20			
t_r	Rise time for both SDA and SCL	f_{SCL} ≤ 400kHz		$20+0.1C_b$(1)(2)		300	ns
		f_{SCL} ≤ 1MHz				120	
t_{of}	Output fall time from V_{IHmin} to V_{ILmax}	10pF< C_b<400pF(2)	f_{SCL} ≤ 400kHz	$20+0.1C_b$(1)(2)		250	
			f_{SCL} ≤ 1MHz			120	
t_{SP}	Spikes suppressed by Input filter			0		50	
I_I	Input current for each I/O Pin	0.1 V_{CC} <V_I <0.9 V_{CC}		-10		10	µA
C_I	Capacitance for each I/O Pin					10	pF
f_{SCL}	SCL clock frequency	f_{PER}(3) > max(10f_{SCL},250kHz)		0		1	MHz
R_P	Value of pull-up resistor	f_{SCL} ≤ 100kHz		(V_{CC}-0.4V)/I_{OL}		100ns/C_b	Ω
		f_{SCL} ≤ 400kHz				300ns/C_b	
		f_{SCL} ≤ 1MHz				550ns/C_b	
$t_{HD;STA}$	Hold time (repeated) START condition	f_{SCL} ≤ 100kHz		4			µs
		f_{SCL} ≤ 400kHz		0.6			
		f_{SCL} ≤ 1MHz		0.26			

Issue: TWI SM bus level one Master or slave remembering data

If a write is made to Data register, prior to Address register, the TWI design sends the data as soon as the write to Address register is made. But the send data will be always 0x00.

Workaround:

Since single interrupt line is shared by both timeout interrupt and other TWI interrupt sources, there is a possibility in software that data register will be written after timeout is detected but before timeout interrupt routine is executed. To avoid this, in software, before writing data register, always ensure that timeout status flag is not set.

Issue: Temperature sensor not calibrated

Temperature sensor factory calibration is not implemented on devices before date code 1324.

Workaround:

None.

Issue: Automatic port override on PORT C

When Waveform generation is enabled on PORT C Timers, Automatic port override of peripherals other than Tc may not work even though the pin is not used as waveform output pin.

Workaround:

No workaround.

Issue: Sext timer is not implemented in slave mode

In slave mode, only Ttout timer is implemented. Sext timer is needed in slave mode to release the SCL line and to allow the master to send a STOP condition. If only master implements Sext timer, slave continues to stretch the SCL line (up to the Ttout timeout in the worse case). Sext = Slave cumulative timeout.

Workaround:

No workaround.