Microchip Technology - ATMEGA32M1-15AZ Datasheet

What is "Embedded - Microcontrollers"?

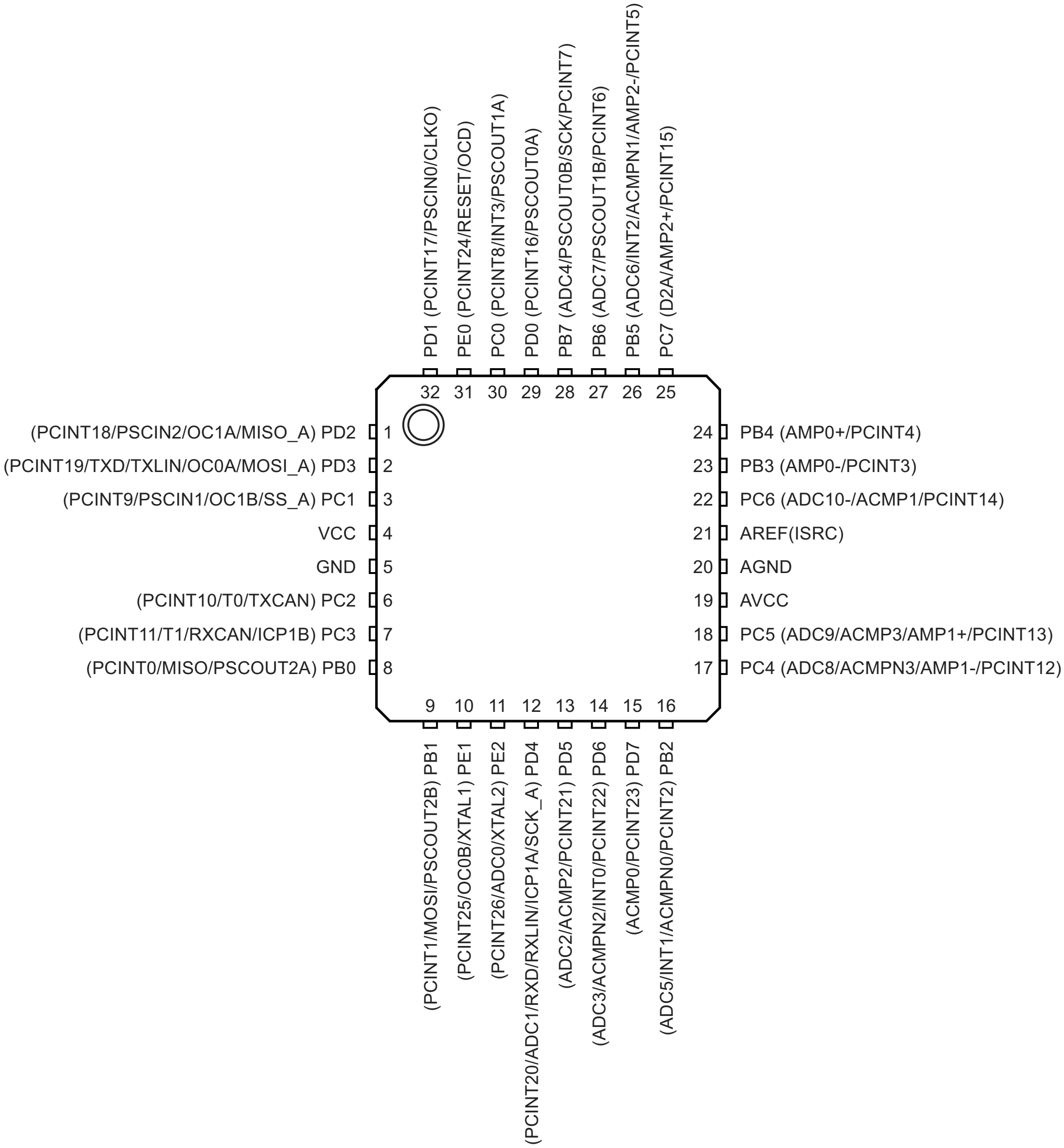
"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" 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-Bit
Speed	16MHz
Connectivity	CANbus, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, Temp Sensor, WDT
Number of I/O	-
Program Memory Size	32KB (16K x 16)
Program Memory Type	FLASH
EEPROM Size	1K x 8
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 5.5V
Data Converters	A/D 11x10b; D/A 1x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°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/atmega32m1-15az

1. Pin Configurations

Figure 1-1. ATmega16/32/64M1 TQFP32/QFN32 (7*7mm) Package


PD1 (PCINT17/PSCIN0/CLKO)
PE0 (PCINT24/RESET/OCD)
PC0 (PCINT8/INT3/PSCOUT1A)
PD0 (PCINT16/PSCOUT0A)
PB7 (ADC4/PSCOUT0B/SCK/PCINT7)
PB6 (ADC7/PSCOUT1B/PCINT6)
PB5 (ADC6/INT2/ACMPN1/AMP2-/PCINT5)
PC7 (D2A/AMP2+/PCINT15)
32
31
30
29
28
27
26
25
(PCINT18/PSCIN2/OC1A/MISO_A) PD2
1
(PCINT19/TXD/TXLIN/OC0A/MOSI_A) PD3
2
(PCINT9/PSCIN1/OC1B/SS_A) PC1
3
VCC
4
GND
5
(PCINT10/T0/TXCAN) PC2
6
(PCINT11/T1/RXCAN/ICP1B) PC3
7
(PCINT0/MISO/PSCOUT2A) PB0
8
24
PB4 (AMP0+/PCINT4)
23
PB3 (AMP0-/PCINT3)
22
PC6 (ADC10-/ACMP1/PCINT14)
21
AREF(ISRC)
20
AGND
19
AVCC
18
PC5 (ADC9/ACMP3/AMP1+/PCINT13)
17
PC4 (ADC8/ACMPN3/AMP1-/PCINT12)
9
10
11
12
13
14
15
16
(PCINT1/MOSI/PSCOUT2B) PB1
(PCINT25/OC0B/XTAL1) PE1
(PCINT26/ADC0/XTAL2) PE2
(PCINT20/ADC1/RXD/RXLIN/ICP1A/SCK_A) PD4
(ADC2/ACMP2/PCINT21) PD5
(ADC3/ACMPN2/INT0/PCINT22) PD6
(ACMP0/PCINT23) PD7
(ADC5/INT1/ACMPN0/PCINT2) PB2

Note: On the engineering samples (Parts marked AT90PWM324), the ACMPN3 alternate function is not located on PC4. It is located on PE2.

Table 1-1. Pin Out Description (Continued)

QFN32 Pin Number	Mnemonic	Type	Name, Function and Alternate Function
3	PC1	I/O	PSCIN1 (PSC Digital Input 1) OC1B (Timer 1 Output Compare B) SS_A (Alternate SPI Slave Select) PCINT9 (Pin Change Interrupt 9)
6	PC2	I/O	T0 (Timer 0 clock input) TXCAN (CAN Transmit Output) PCINT10 (Pin Change Interrupt 10)
7	PC3	I/O	T1 (Timer 1 clock input) RXCAN (CAN Receive Input) ICP1B (Timer 1 input capture alternate B input) PCINT11 (Pin Change Interrupt 11)
17	PC4	I/O	ADC8 (Analog Input Channel 8) AMP1- (Analog Differential Amplifier 1 Negative Input) ACMPN3 (analog comparator 3 Negative Input) PCINT12 (Pin Change Interrupt 12)
18	PC5	I/O	ADC9 (Analog Input Channel 9) AMP1+ (Analog Differential Amplifier 1 Positive Input) ACMP3 (analog comparator 3 Positive Input) PCINT13 (Pin Change Interrupt 13)
22	PC6	I/O	ADC10 (Analog Input Channel 10) ACMP1 (analog comparator 1 Positive Input) PCINT14 (Pin Change Interrupt 14)
25	PC7	I/O	D2A (DAC output) AMP2+ (Analog Differential Amplifier 2 Positive Input) PCINT15 (Pin Change Interrupt 15)
29	PD0	I/O	PSCOUT0A (PSC Module 0 Output A) PCINT16 (Pin Change Interrupt 16)
32	PD1	I/O	PSCIN0 (PSC Digital Input 0) CLKO (System Clock Output) PCINT17 (Pin Change Interrupt 17)
1	PD2	I/O	OC1A (Timer 1 Output Compare A) PSCIN2 (PSC Digital Input 2) MISO_A (Programming and alternate SPI Master In Slave Out) PCINT18 (Pin Change Interrupt 18)
2	PD3	I/O	TXD (UART Tx data) TXLIN (LIN Transmit Output) OC0A (Timer 0 Output Compare A) SS (SPI Slave Select) MOSI_A (Programming and alternate Master Out SPI Slave In) PCINT19 (Pin Change Interrupt 19)

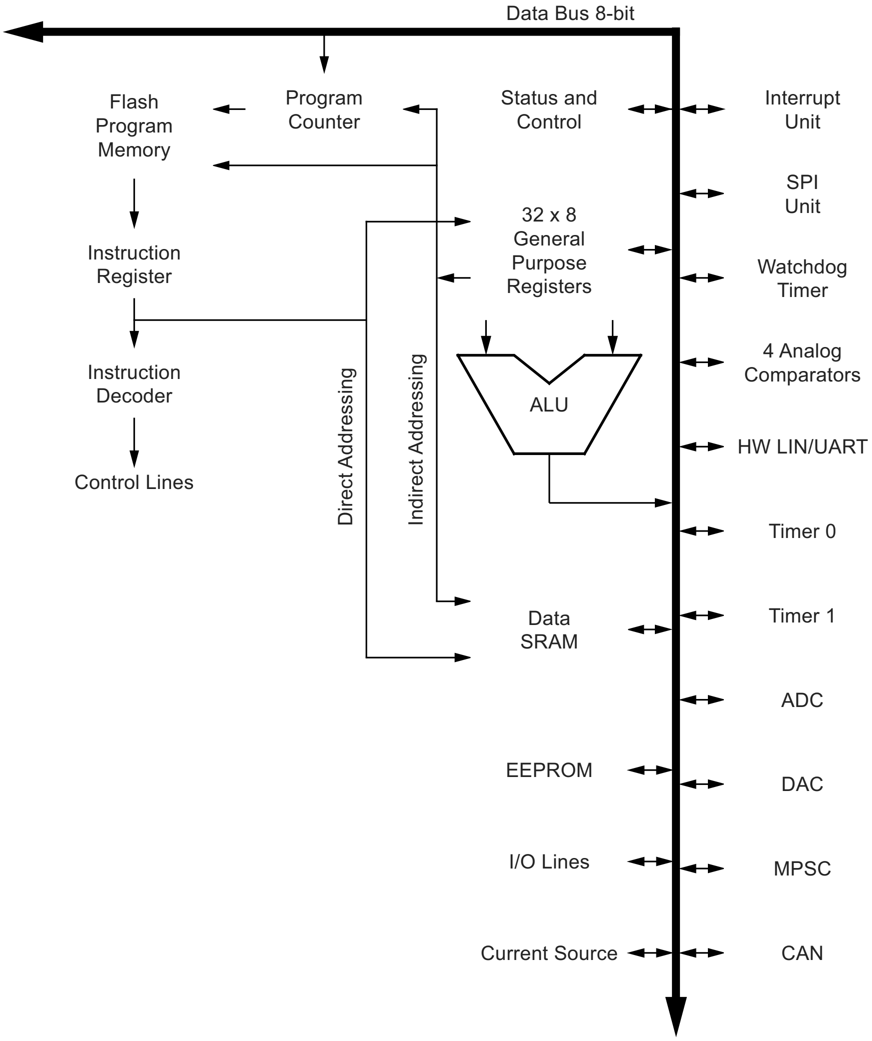
Note: 1. On the first engineering samples (Parts marked AT90PWM324), the ACMPN3 alternate function is not located on PC4. It is located on PE2.

2. Overview

The Atmel® ATmega16/32/64/M1/C1 is a low-power CMOS 8-bit microcontroller based on the AVR® enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the Atmel ATmega16/32/64/M1/C1 achieves throughputs approaching 1MIPS per MHz allowing the system designer to optimize power consumption versus processing speed.

2.1 Block Diagram

Figure 2-1. Block Diagram



The AVR core combines a rich instruction set with 32 general purpose working registers. All the 32 registers are directly connected to the Arithmetic Logic Unit (ALU), allowing two independent registers to be accessed in one single instruction executed in one clock cycle. The resulting architecture is more code efficient while achieving throughputs up to ten times faster than conventional CISC microcontrollers.

The Atmel ATmega16/32/64/M1/C1 provides the following features: 16K/32K/64K bytes of In-System Programmable Flash with Read-while-write capabilities, 512/1024/2048 bytes EEPROM, 1024/2048/4096 bytes SRAM, 27 general purpose I/O lines, 32 general purpose working registers, one Motor Power Stage Controller, two flexible Timer/Counters with compare modes and PWM, one UART with HW LIN, an 11-channel 10-bit ADC with two differential input stages with programmable gain, a 10-bit DAC, a programmable Watchdog Timer with Internal Individual Oscillator, an SPI serial port, an On-chip Debug system and four software selectable power saving modes.

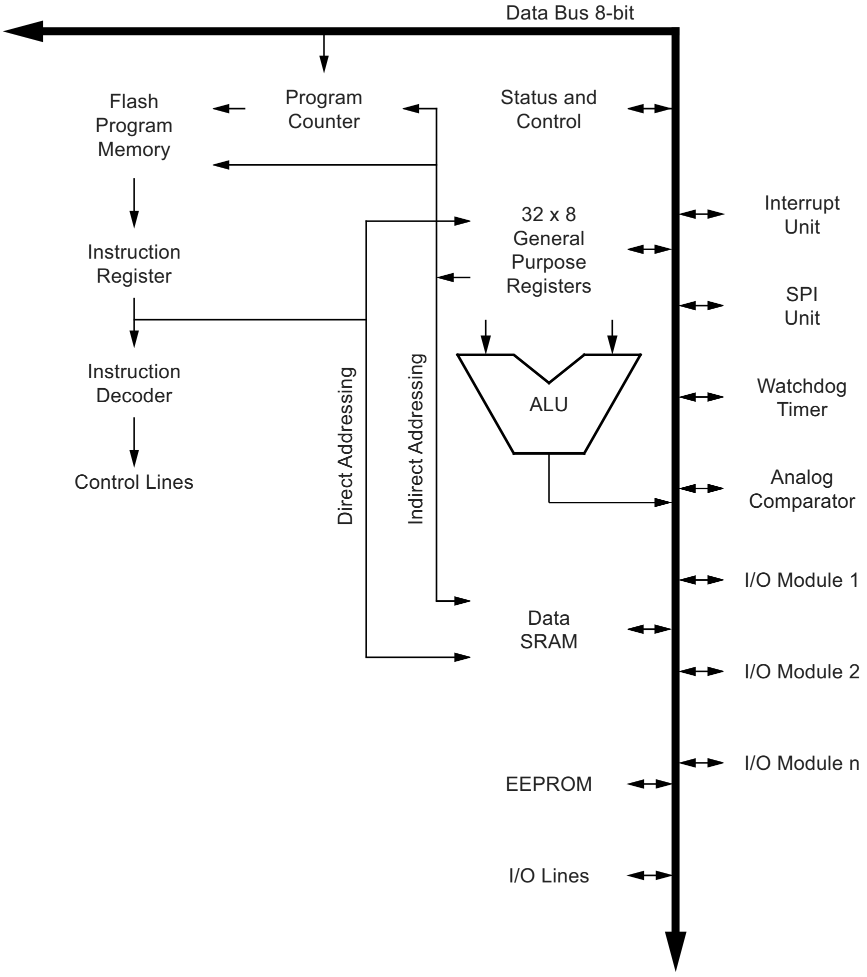
3. AVR CPU Core

3.1 Introduction

This section discusses the AVR® core architecture in general. The main function of the CPU core is to ensure correct program execution. The CPU must therefore be able to access memories, perform calculations, control peripherals, and handle interrupts.

3.2 Architectural Overview

Figure 3-1. Block Diagram of the AVR Architecture



In order to maximize performance and parallelism, the AVR uses a Harvard architecture – with separate memories and buses for program and data. Instructions in the program memory are executed with a single level pipelining. While one instruction is being executed, the next instruction is pre-fetched from the program memory. This concept enables instructions to be executed in every clock cycle. The program memory is in-system reprogrammable Flash memory.

The fast-access register file contains 32 x 8-bit general purpose working registers with a single clock cycle access time. This allows single-cycle arithmetic logic unit (ALU) operation. In a typical ALU operation, two operands are output from the register file, the operation is executed, and the result is stored back in the register file – in one clock cycle.

The following code example shows one assembly and one C function for changing the time-out value of the watchdog timer.

Assembly Code Example[1]

```
WDT_Prescaler_Change:
        ; Turn off global interrupt
        cli
        ; Reset Watchdog Timer
        wdr
        ; Start timed sequence
        lds r16, WDTCSR
        ori   r16, (1<<WDCE) | (1<<WDE)
        sts WDTCSR, r16
        ; -- Got four cycles to set the new values from here -
        ; Set new prescaler(time-out) value = 64K cycles (~0.5 s)
        ldi   r16, (1<<WDE) | (1<<WDP2) | (1<<WDP0)
        sts WDTCSR, r16
        ; -- Finished setting new values, used 2 cycles -
        ; Turn on global interrupt
        sei
        ret
```

C Code Example[1]

```
void WDT_Prescaler_Change(void)
{
        __disable_interrupt();
        __watchdog_reset();
        /* Start timed sequence */
        WDTCSR |= (1<<WDCE) | (1<<WDE);
        /* Set new prescaler(time-out) value = 64K cycles (~0.5 s) */
        WDTCSR = (1<<WDE) | (1<<WDP2) | (1<<WDP0);
        __enable_interrupt();
}
```

Notes: 1. The example code assumes that the part specific header file is included.

2. The watchdog timer should be reset before any change of the WDP bits, since a change in the WDP bits can result in a time-out when switching to a shorter time-out period;

7.4.1 Watchdog Timer Control Register - WDTCSR

Bit	7	6	5	4	3	2	1	0	
	WDIF	WDIE	WDP3	WDCE	WDE	WDP2	WDP1	WDP0	WDTCSR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	X	0	0	0	

- **Bit 7 - WDIF: Watchdog Interrupt Flag**

This bit is set when a time-out occurs in the watchdog timer and the watchdog timer is configured for interrupt. WDIF is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, WDIF is cleared by writing a logic one to the flag. When the I-bit in SREG and WDIE are set, the watchdog time-out interrupt is executed.

- **Bit 6 - WDIE: Watchdog Interrupt Enable**

When this bit is written to one and the I-bit in the status register is set, the watchdog interrupt is enabled. If WDE is cleared in combination with this setting, the watchdog timer is in interrupt mode, and the corresponding interrupt is executed if time-out in the watchdog timer occurs.

9.2.2 Toggling the Pin

Writing a logic one to PINxn toggles the value of PORTxn, independent on the value of DDRxn. Note that the SBI instruction can be used to toggle one single bit in a port.

9.2.3 Switching Between Input and Output

When switching between tri-state ({DDxn, PORTxn} = 0b00) and output high ({DDxn, PORTxn} = 0b11), an intermediate state with either pull-up enabled {DDxn, PORTxn} = 0b01) or output low ({DDxn, PORTxn} = 0b10) must occur. Normally, the pull-up enabled state is fully acceptable, as a high-impedant environment will not notice the difference between a strong high driver and a pull-up. If this is not the case, the PUD bit in the MCUCR Register can be set to disable all pull-ups in all ports.

Switching between input with pull-up and output low generates the same problem. The user must use either the tri-state ({DDxn, PORTxn} = 0b00) or the output high state ({DDxn, PORTxn} = 0b11) as an intermediate step.

Table 9-1 summarizes the control signals for the pin value.

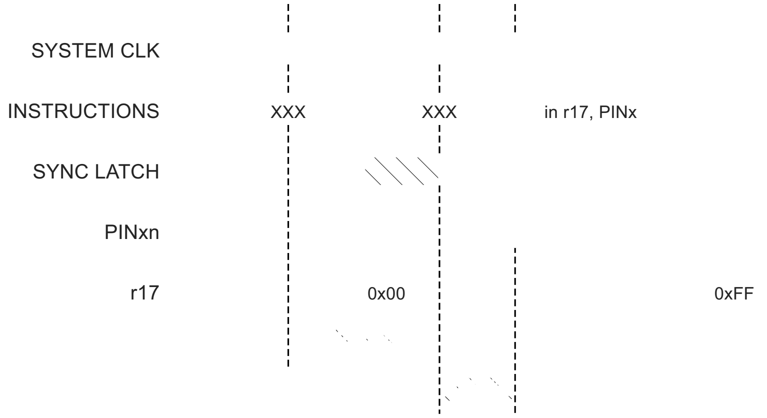
Table 9-1. Port Pin Configurations

DDxn	PORTxn	PUD (in MCUCR)	I/O	Pull-up	Comment
0	0	X	Input	No	Default configuration after reset. Tri-state (Hi-Z)
0	1	0	Input	Yes	Pxn will source current if ext. pulled low.
0	1	1	Input	No	Tri-state (Hi-Z)
1	0	X	Output	No	Output low (sink)
1	1	X	Output	No	Output high (source)

9.2.4 Reading the Pin Value

Independent of the setting of data direction bit DDxn, the port pin can be read through the PINxn register bit. As shown in Figure 9-2, the PINxn register bit and the preceding latch constitute a synchronizer. This is needed to avoid metastability if the physical pin changes value near the edge of the internal clock, but it also introduces a delay. Figure 9-3 shows a timing diagram of the synchronization when reading an externally applied pin value. The maximum and minimum propagation delays are denoted $t_{pd,max}$ and $t_{pd,min}$ respectively.

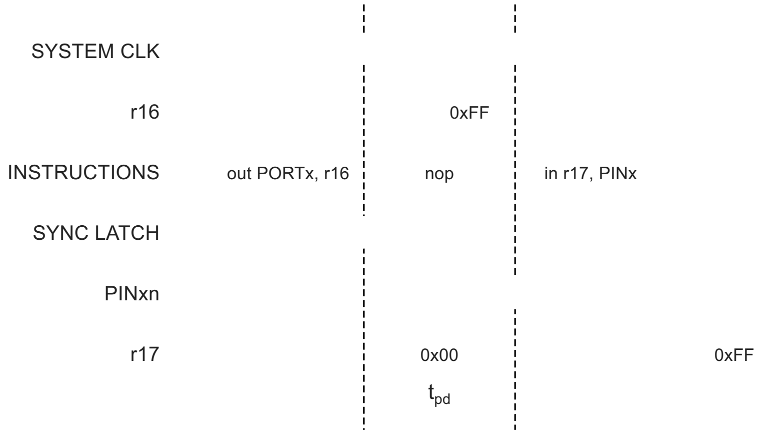
Figure 9-3. Synchronization when Reading an Externally Applied Pin Value



Consider the clock period starting shortly after the first falling edge of the system clock. The latch is closed when the clock is low, and goes transparent when the clock is high, as indicated by the shaded region of the “SYNC LATCH” signal. The signal value is latched when the system clock goes low. It is clocked into the PINxn register at the succeeding positive clock edge. As indicated by the two arrows $t_{pd,max}$ and $t_{pd,min}$, a single signal transition on the pin will be delayed between ½ and 1½ system clock period depending upon the time of assertion.

When reading back a software assigned pin value, a nop instruction must be inserted as indicated in Figure 9-4. The out instruction sets the "SYNC LATCH" signal at the positive edge of the clock. In this case, the delay t_{pd} through the synchronizer is 1 system clock period.

Figure 9-4. Synchronization when Reading a Software Assigned Pin Value



The following code example shows how to set port B pins 0 and 1 high, 2 and 3 low, and define the port pins from 4 to 7 as input with pull-ups assigned to port pins 6 and 7. The resulting pin values are read back again, but as previously discussed, a nop instruction is included to be able to read back the value recently assigned to some of the pins.

Assembly Code Example[1]

```
      ...
      ; Define pull-ups and set outputs high
      ; Define directions for port pins
      ldi                 r16, (1<<PB7)|(1<<PB6)|(1<<PB1)|(1<<PB0)
      ldi                 r17, (1<<DDB3)|(1<<DDB2)|(1<<DDB1)|(1<<DDB0)
      out                 PORTB, r16
      out                 DDRB, r17
      ; Insert nop for synchronization
      nop
      ; Read port pins
      in                  r16, PINB
      ...
```

C Code Example

```
unsigned char i;
      ...
      /* Define pull-ups and set outputs high */
      /* Define directions for port pins */
      PORTB = (1<<PB7)|(1<<PB6)|(1<<PB1)|(1<<PB0);
      DDRB = (1<<DDB3)|(1<<DDB2)|(1<<DDB1)|(1<<DDB0);
      /* Insert nop for synchronization*/
      _NOP();
      /* Read port pins */
      i = PINB;
      ...
```

Note: 1. For the assembly program, two temporary registers are used to minimize the time from pull-ups are set on pins 0, 1, 6, and 7, until the direction bits are correctly set, defining bit 2 and 3 as low and redefining bits 0 and 1 as strong high drivers.

9.3.5 Alternate Functions of Port E

The Port E pins with alternate functions are shown in Table 9-12.

Table 9-12. Port E Pins Alternate Functions

Port Pin	Alternate Function
PE2	XTAL2 (XTAL Output)
	ADC0 (Analog Input Channel 0)
	PCINT26 (Pin Change Interrupt 26)
PE1	XTAL1 (XTAL Input)
	OC0B (Timer 0 Output Compare B)
	PCINT25 (Pin Change Interrupt 25)
PE0	RESET# (Reset Input)
	OCD (On Chip Debug I/O)
	PCINT24 (Pin Change Interrupt 24)

Note: On the engineering samples (Parts marked AT90PWM324), the ACMPN3 alternate function is not located on PC4. It is located on PE2.

The alternate pin configuration is as follows:

- **PCINT26/XTAL2/ADC0 – Bit 2**

XTAL2: Chip clock oscillator pin 2. Used as clock pin for crystal oscillator or low-frequency crystal oscillator. When used as a clock pin, the pin can not be used as an I/O pin.

ADC0, analog to digital converter, input channel 0.

PCINT26, pin change interrupt 26.

- **PCINT25/XTAL1/OC0B – Bit 1**

XTAL1: Chip clock oscillator pin 1. Used for all chip clock sources except internal calibrated RC oscillator. When used as a clock pin, the pin can not be used as an I/O pin.

OC0B, output compare Match B output: This pin can serve as an external output for the Timer/Counter0 output compare B. The pin has to be configured as an output (DDE1 set “one”) to serve this function. This pin is also the output pin for the PWM mode timer function.

PCINT25, pin change interrupt 25.

- **PCINT24/$\overline{\text{RESET}}$/OCD – Bit 0**

$\overline{\text{RESET}}$, reset pin: When the RSTDISBL Fuse is programmed, this pin functions as a normal I/O pin, and the part will have to rely on power-on reset and brown-out reset as its reset sources. When the RSTDISBL Fuse is unprogrammed, the reset circuitry is connected to the pin, and the pin can not be used as an I/O pin.

If PE0 is used as a reset pin, DDE0, PORTE0 and PINE0 will all read 0.

PCINT24, pin change interrupt 24.

The OCR0x registers are double buffered when using any of the pulse width modulation (PWM) modes. For the normal and clear timer on compare (CTC) modes of operation, the double buffering is disabled. The double buffering synchronizes the update of the OCR0x compare registers to either top or bottom of the counting sequence. The synchronization prevents the occurrence of odd-length, non-symmetrical PWM pulses, thereby making the output glitch-free.

The OCR0x register access may seem complex, but this is not case. When the double buffering is enabled, the CPU has access to the OCR0x buffer register, and if double buffering is disabled the CPU will access the OCR0x directly.

12.4.1 Force Output Compare

In non-PWM waveform generation modes, the match output of the comparator can be forced by writing a one to the force output compare (FOC0x) bit. Forcing compare match will not set the OCF0x flag or reload/clear the timer, but the OC0x pin will be updated as if a real compare match had occurred (the COM0x1:0 bits settings define whether the OC0x pin is set, cleared or toggled).

12.4.2 Compare Match Blocking by TCNT0 Write

All CPU write operations to the TCNT0 register will block any compare match that occur in the next timer clock cycle, even when the timer is stopped. This feature allows OCR0x to be initialized to the same value as TCNT0 without triggering an interrupt when the Timer/Counter clock is enabled.

12.4.3 Using the Output Compare Unit

Since writing TCNT0 in any mode of operation will block all compare matches for one timer clock cycle, there are risks involved when changing TCNT0 when using the output compare unit, independently of whether the Timer/Counter is running or not. If the value written to TCNT0 equals the OCR0x value, the compare match will be missed, resulting in incorrect waveform generation. Similarly, do not write the TCNT0 value equal to BOTTOM when the counter is downcounting.

The setup of the OC0x should be performed before setting the data direction register for the port pin to output. The easiest way of setting the OC0x value is to use the force output compare (FOC0x) strobe bits in normal mode. The OC0x registers keep their values even when changing between waveform generation modes.

Be aware that the COM0x1:0 bits are not double buffered together with the compare value. Changing the COM0x1:0 bits will take effect immediately.

12.5 Compare Match Output Unit

The compare output mode (COM0x1:0) bits have two functions. The waveform generator uses the COM0x1:0 bits for defining the output compare (OC0x) state at the next compare match. Also, the COM0x1:0 bits control the OC0x pin output source. Figure 12-4 shows a simplified schematic of the logic affected by the COM0x1:0 bit setting. The I/O registers, I/O bits, and I/O pins in the figure are shown in bold. Only the parts of the general I/O port control registers (DDR and PORT) that are affected by the COM0x1:0 bits are shown. When referring to the OC0x state, the reference is for the internal OC0x register, not the OC0x pin. If a system reset occur, the OC0x register is reset to “0”.

Figure 12-10 shows the setting of OCF0B in all modes and OCF0A in all modes except CTC mode and PWM mode, where OCR0A is TOP.

Figure 12-10.Timer/Counter Timing Diagram, Setting of OCF0x, with Prescaler ($f_{clk\_I/O}$/8)

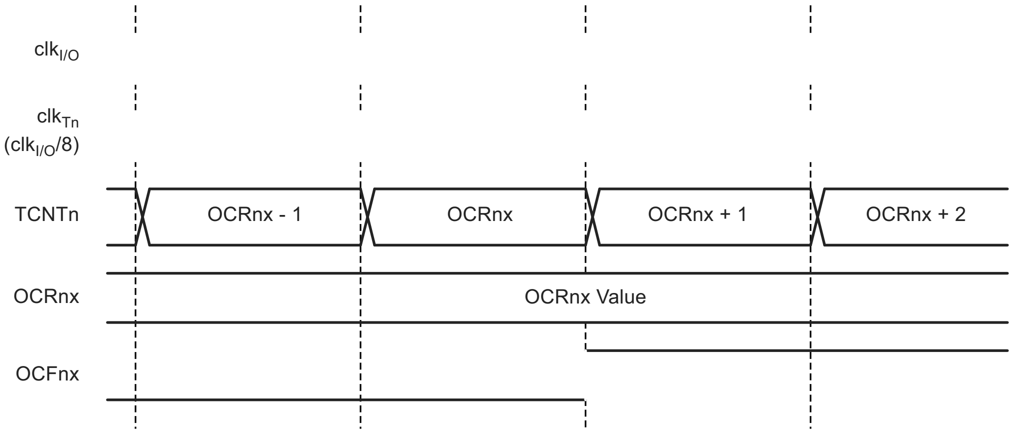
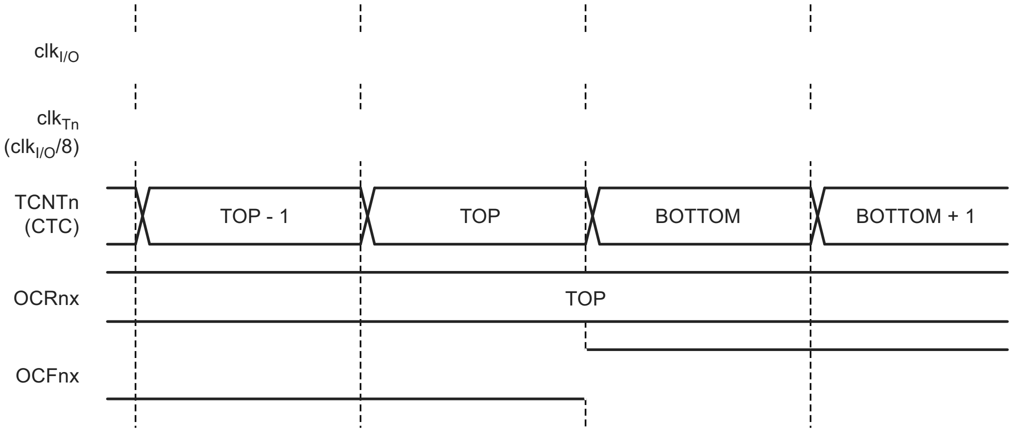


Figure 12-11 shows the setting of OCF0A and the clearing of TCNT0 in CTC mode and fast PWM mode where OCR0A is TOP.

Figure 12-11.Timer/Counter Timing Diagram, Clear Timer on Compare Match mode, with Prescaler ($f_{clk\_I/O}$/8)



12.8 8-bit Timer/Counter Register Description

12.8.1 Timer/Counter Control Register A – TCCR0A

Bit	7	6	5	4	3	2	1	0	
	COM0A1	**COM0A0**	**COM0B1**	**COM0B0**	**–**	**–**	**WGM01**	**WGM00**	**TCCR0A**
Read/Write	R/W	R/W	R/W	R/W	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bits 7:6 – COM0A1:0: Compare Match Output A Mode**

These bits control the Output Compare pin (OC0A) behavior. If one or both of the COM0A1:0 bits are set, the OC0A output overrides the normal port functionality of the I/O pin it is connected to. However, note that the data direction register (DDR) bit corresponding to the OC0A pin must be set in order to enable the output driver.

When OC0A is connected to the pin, the function of the COM0A1:0 bits depends on the WGM02:0 bit setting. Table 12-2 on page 87 shows the COM0A1:0 bit functionality when the WGM02:0 bits are set to a normal or CTC mode (non-PWM).

On-time A = (POCRnRAH/L - POCRnSAH/L) × 1/Fclkpsc

On-time B = (POCRnRBH/L - POCRnSBH/L) × 1/Fclkpsc

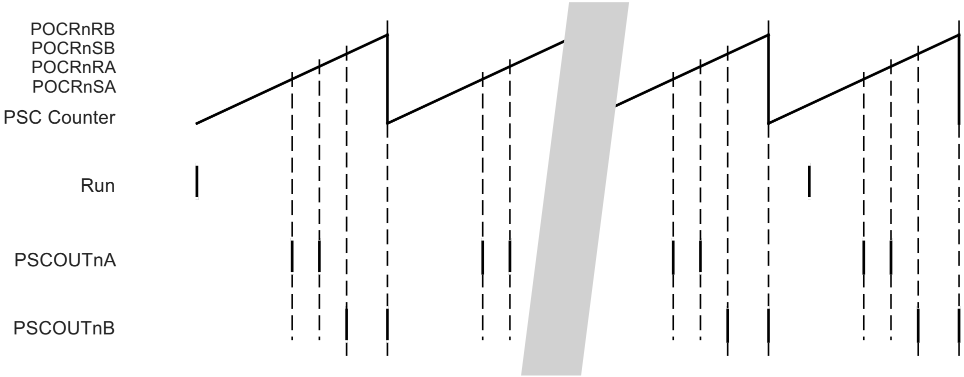
Dead-time A = (POCRnSAH/L + 1) × 1/Fclkpsc

Dead-time B = (POCRnSBH/L – POCRnRAH/L) × 1/Fclkpsc

Minimal value for dead-time A = 1/Fclkpsc

If the overlap protection is disabled, in one-ramp mode, PSCOUTnA and PSCOUTnB outputs can be configured to overlap each other, though in normal use this is not desirable.

Figure 14-5. Controlled Start and Stop Mechanism in One-Ramp Mode

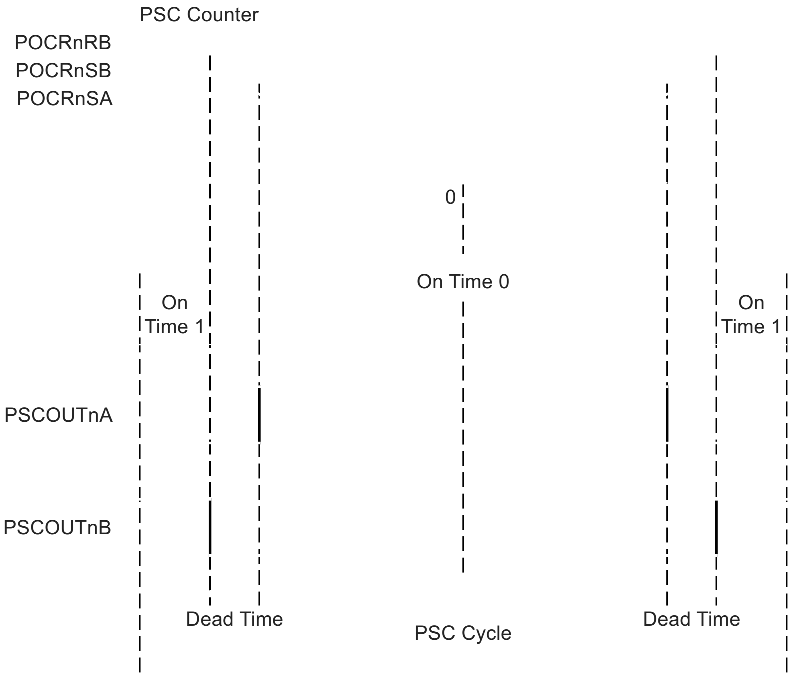


Note: See Section 14.16.8 “PSC Control Register – PCTL” on page 130 (PCCYC = 1)

14.5.3.2 Center Aligned Mode

In center aligned mode, the center of PSCOUTnA and PSCOUTnB signals are centered.

Figure 14-6. PSCOUTnA and PSCOUTnB Basic Waveforms in Center Aligned Mode



- **Bit 4 – POEN2A: PSC Output 2A Enable**

When this bit is clear, I/O pin affected to PSCOUT2A acts as a standard port.

When this bit is set, I/O pin affected to PSCOUT2A is connected to the PSC module 2 waveform generator A output and is set and clear according to the PSC operation.

- **Bit 3 – POEN1B: PSC Output 1B Enable**

When this bit is clear, I/O pin affected to PSCOUT1B acts as a standard port.

When this bit is set, I/O pin affected to PSCOUT1B is connected to the PSC module 1 waveform generator B output and is set and clear according to the PSC operation.

- **Bit 2 – POEN1A: PSC Output 1A Enable**

When this bit is clear, I/O pin affected to PSCOUT1A acts as a standard port.

When this bit is set, I/O pin affected to PSCOUT1A is connected to the PSC module 1 waveform generator A output and is set and clear according to the PSC operation.

- **Bit 1 – POEN0B: PSC Output 0B Enable**

When this bit is clear, I/O pin affected to PSCOUT0B acts as a standard port.

When this bit is set, I/O pin affected to PSCOUT0B is connected to the PSC module 0 waveform generator B output and is set and clear according to the PSC operation.

- **Bit 0 – POEN0A: PSC Output 0A Enable**

When this bit is clear, I/O pin affected to PSCOUT0A acts as a standard port.

When this bit is set, I/O pin affected to PSCOUT0A is connected to the PSC module 0 waveform generator A output and is set and clear according to the PSC operation.

14.16.2 PSC Synchro Configuration – PSYNC

Bit	7	6	5	4	3	2	1	0	
	-	**-**	**PSYNC21**	**PSYNC20**	**PSYNC11**	**PSYNC10**	**PSYNC01**	**PSYNC00**	**PSYNC**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – not use**

not use

- **Bit 6 – not use**

not use

- **Bit 5:4 – PSYNC21:0: Synchronization Out for ADC Selection**

Select the polarity and signal source for generating a signal which will be sent from module 2 to the ADC for synchronization

- **Bit 3:2 – PSYNC11:0: Synchronization Out for ADC Selection**

Select the polarity and signal source for generating a signal which will be sent from module 1 to the ADC for synchronization

- **Bit 1:0 – PSYNC01:0: Synchronization Out for ADC Selection**

Select the polarity and signal source for generating a signal which will be sent from module 0 to the ADC for synchronization.

Table 14-8. Synchronization Source Description in One Ramp Mode

PSYNCn1	PSYNCn0	Description
0	0	Send signal on leading edge of PSCOUTnA(match with OCRnSA)
0	1	Send signal on trailing edge of PSCOUTnA(match with OCRnRA or fault/retrigger on part A)
1	0	Send signal on leading edge of PSCOUTnB (match with OCRnSB)
1	1	Send signal on trailing edge of PSCOUTnB (match with OCRnRB or fault/retrigger on part B)

16.6.4 Stamping Message

The capture of the timer value is done in the MOb which receives or sends the frame. All managed MOb are stamped, the stamping of a received (sent) frame occurs on RxOk (TXOK).

16.7 Error Management

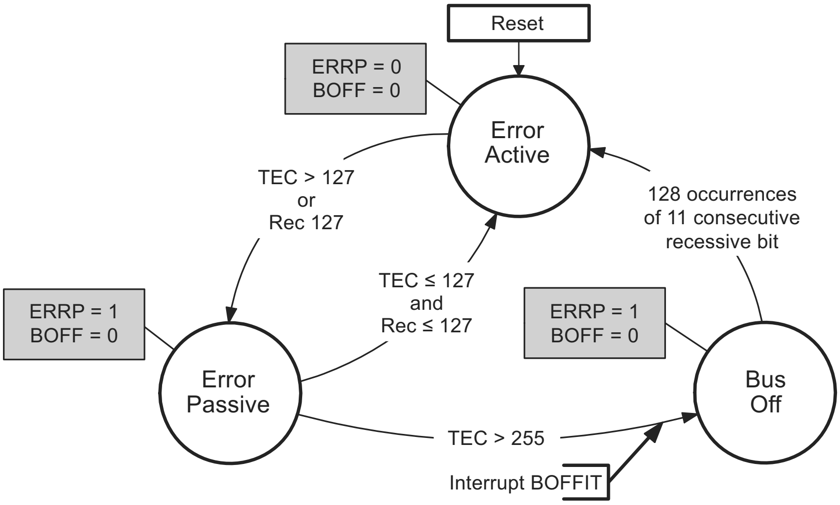
16.7.1 Fault Confinement

The CAN channel may be in one of the three following states:

- Error active (default):
  The CAN channel takes part in bus communication and can send an active error frame when the CAN macro detects an error.
- Error passive:
  The CAN channel cannot send an active error frame. It takes part in bus communication, but when an error is detected, a passive error frame is sent. Also, after a transmission, an error passive unit will wait before initiating further transmission.
- Bus off:
  The CAN channel is not allowed to have any influence on the bus.

For fault confinement, a transmit error counter (TEC) and a receive error counter (REC) are implemented. BOFF and ERRP bits give the information of the state of the CAN channel. Setting BOFF to one may generate an interrupt.

Figure 16-12. Line Error Mode



Note: More than one REC/TEC change may apply during a given message transfer.

16.10.7 CAN Status Interrupt MOb Registers - CANSIT2 and CANSIT1

Bit	7	6	5	4	3	2	1	0	
	-	**-**	**SIT5**	**SIT4**	**SIT3**	**SIT2**	**SIT1**	**SIT0**	**CANSIT2**
	-	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**CANSIT1**
Bit	15	14	13	12	11	10	9	8	
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	

- **Bits 5:0 - SIT5:0: Status of Interrupt by MOb**
  - 0 - no interrupt.
  - 1- MOb interrupt.

Note: Example: CANSIT2 = 0010 0001_b: MOb 0 and 5 interrupts.

- **Bit 15:6 – Reserved Bits**

These bits are reserved for future use.

16.10.8 CAN Bit Timing Register 1 - CANBT1

Bit	7	6	5	4	3	2	1	0	
	-	**BRP5**	**BRP4**	**BRP3**	**BRP2**	**BRP1**	**BRP0**	**-**	**CANBT1**
Read/Write	-	R/W	R/W	R/W	R/W	R/W	R/W	-	
Initial Value	-	0	0	0	0	0	0	-	

- **Bit 7– Reserved Bit**

This bit is reserved for future use. For compatibility with future devices, it must be written to zero when CANBT1 is written.

- **Bit 6:1 – BRP5:0: Baud Rate Prescaler**

The period of the CAN controller system clock Tscl is programmable and determines the individual bit timing.

$$\mathrm{Tscl} = \frac{\mathrm{BRP[5:0]} + 1}{\mathrm{clk_{IO}frequency}}$$

If ‘BRP[5..0]=0’, see Section 16.4.3 “Baud Rate” on page 148 and Section • “Bit 0 – SMP: Sample Point(s)” on page 164.

- **Bit 0 – Reserved Bit**

This bit is reserved for future use. For compatibility with future devices, it must be written to zero when CANBT1 is written.

16.10.9 CAN Bit Timing Register 2 - CANBT2

Bit	7	6	5	4	3	2	1	0	
	-	**SJW1**	**SJW0**	**-**	**PRS2**	**PRS1**	**PRS0**	**-**	**CANBT2**
Read/Write	-	R/W	R/W	-	R/W	R/W	R/W	-	
Initial Value	-	0	0	-	0	0	0	-	

- **Bit 7– Reserved Bit**

This bit is reserved for future use. For compatibility with future devices, it must be written to zero when CANBT2 is written.

- **Bit 4– ADIF: ADC Interrupt Flag**

Set by hardware as soon as a conversion is complete and the data register are updated with the conversion result.
Cleared by hardware when executing the corresponding interrupt handling vector.
Alternatively, ADIF can be cleared by writing it to logical one.

- **Bit 3– ADIE: ADC Interrupt Enable Bit**

Set this bit to activate the ADC end of conversion interrupt.
Clear it to disable the ADC end of conversion interrupt.

- **Bit 2, 1, 0– ADPS2, ADPS1, ADPS0: ADC Prescaler Selection Bits**

These 3 bits determine the division factor between the system clock frequency and input clock of the ADC.
The different setting are shown in Table 18-6.

.

Table 18-6. ADC Prescaler Selection

ADPS2	ADPS1	ADPS0	Division Factor
0	0	0	2
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

18.9.3 ADC control and status register B– ADCSRB

Bit	7	6	5	4	3	2	1	0	
	ADHSM	ISRCEN	AREFEN	-	ADTS3	ADTS2	ADTS1	ADTS0	ADCSRB
Read/Write	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – ADHSM: ADC High-speed Mode**

Writing this bit to one enables the ADC high-speed mode. Set this bit if you wish to convert with an ADC clock frequency higher than 200KHz.

Clear this bit to reduce the power consumption of the ADC when the ADC clock frequency is lower than 200KHz.

- **Bit 6 – ISRCEN: Current Source Enable**

Set this bit to source a 100µA current to the AREF pin.
Clear this bit to use AREF pin as analog reference pin.

- **Bit 5 – AREFEN: Analog Reference pin Enable**

Set this bit to connect the internal AREF circuit to the AREF pin.
Clear this bit to disconnect the internal AREF circuit from the AREF pin.

- **Bit 4 – Res: Reserved Bit**

This bit is unused bit in the ATmega16/32/64/M1/C1, and will always read as zero.

- **Bit 3, 2, 1, 0– ADTS3:ADTS0: ADC Auto Trigger Source Selection Bits**

These bits are only necessary in case the ADC works in auto trigger mode. It means if ADATE bit in ADCSRA register is set.

In accordance with Table 18-6 on page 212, these 3 bits select the interrupt event which will generate the trigger of the start of conversion. The start of conversion will be generated by the rising edge of the selected interrupt flag whether the interrupt is enabled or not. In case of trig on PSCnASY event, there is no flag. So in this case a conversion will start each time the trig event appears and the previous conversion is completed.

20.3 Use of ADC Amplifiers

Thanks to AMPCMP0 configuration bit, comparator 0 positive input can be connected to amplifier O output. In that case, the clock of comparator 0 is twice the amplifier 0 clock. See Section 18.11.1 “Amplifier 0 control and status register – AMP0CSR” on page 218.

Thanks to AMPCMP1 configuration bit, comparator 1 positive input can be connected to amplifier 1 output. In that case, the clock of comparator 1 is twice the amplifier 1 clock. See Section 18.11.2 “Amplifier 1 Control and Status Register – AMP1CSR” on page 219.

Thanks to AMPCMP2 configuration bit, comparator 2 positive input can be connected to amplifier 2 output. In that case, the clock of comparator 2 is twice the amplifier 2 clock. See Section 18.11.2 “Amplifier 1 Control and Status Register – AMP1CSR” on page 219.

20.4 Analog Comparator Register Description

Each analog comparator has its own control register.

A dedicated register has been designed to consign the outputs and the flags of the 4 analog comparators.

20.4.1 Analog Comparator 0 Control Register – AC0CON

Bit	7	6	5	4	3	2	1	0	
	AC0EN	AC0IE	AC0IS1	AC0IS0	ACCKSEL	AC0M2	AC0M1	AC0M0	AC0CON
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7– AC0EN: analog comparator 0 Enable Bit**

Set this bit to enable the analog comparator 0.
Clear this bit to disable the analog comparator 0.

- **Bit 6– AC0IE: analog comparator 0 Interrupt Enable bit**

Set this bit to enable the analog comparator 0 interrupt.
Clear this bit to disable the analog comparator 0 interrupt.

- **Bit 5, 4– AC0IS1, AC0IS0: analog comparator 0 Interrupt Select bit**

These 2 bits determine the sensitivity of the interrupt trigger.
The different setting are shown in Table 18-7.

Table 20-1. Interrupt Sensitivity Selection

AC0IS1	AC0IS0	Description
0	0	Comparator interrupt on output toggle
0	1	Reserved
1	0	Comparator interrupt on output falling edge
1	1	Comparator interrupt on output rising edge

- **Bit 3 – ACCKSEL: Analog Comparator Clock Select**

Set this bit to use the 16MHz PLL output as comparator clock. Clear this bit to use the CLK_{IO} as comparator clock.

- **Bit 2, 1, 0– AC0M2, AC0M1, AC0M0: Analog Comparator 0 Multiplexer Register**

These 3 bits determine the input of the negative input of the analog comparator.
The different setting are shown in Table 20-2 on page 228.

- **Bit 1– AC1O: Analog Comparator 1 Output Bit**

AC1O bit is directly the output of the analog comparator 1.
Set when the output of the comparator is high.
Cleared when the output comparator is low.

- **Bit 0– AC0O: Analog Comparator 0 Output Bit**

AC0O bit is directly the output of the analog comparator 0.
Set when the output of the comparator is high.
Cleared when the output comparator is low.

20.4.6 Digital Input Disable Register 0 – DIDR0

Bit	7	6	5	4	3	2	1	0	
	ADC7D	**ADC6D ACMPN1D AMP2ND**	**ADC5D ACMPN0D**	**ADC4D**	**ADC3D ACMPN2D**	**ADC2D ACMP2D**	**ADC1D**	**ADC0D ACMPN3D**	**DIDR0**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 6, 5, 3, 2, 0 – ACMPN1D, ACMPN0D, ACMPN2D, ACMP2D and ACMPN3D: ACMPN1, ACMPN0, ACMPN2, ACMP2 and ACMPN3 Digital Input Disable**

When this bit is written logic one, the digital input buffer on the corresponding Analog pin is disabled. The corresponding PIN Register bit will always read as zero when this bit is set. When an analog signal is applied to one of these pins and the digital input from this pin is not needed, this bit should be written logic one to reduce power consumption in the digital input buffer.

20.4.7 Digital Input Disable Register 1– DIDR1

Bit	7	6	5	4	3	2	1	0	
	-	**AMP2PD**	**ACMP0D**	**AMP0PD**	**AMP0ND**	**ADC10D ACMP1D**	**ADC9D AMP1PD** ACMP3D	**ADC8D AMP1ND**	**DIDR1**
Read/Write	-	-	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 5, 2, 1: ACMP0D, ACMP1PD, ACMP3PD: ACMP0, ACMP1P, ACMP3P Digital Input Disable**

When this bit is written logic one, the digital input buffer on the corresponding analog pin is disabled. The corresponding PIN Register bit will always read as zero when this bit is set. When an analog signal is applied to one of these pins and the digital input from this pin is not needed, this bit should be written logic one to reduce power consumption in the digital input buffer.

Table 24-11. Read-while-write Limit

Section	Pages	Address
Read-while-write section (RWW)	224	0x0000 - 0x37FF
No Read-while-write section (NRWW)	32	0x3800 - 0x3FFF

For details about these two section, see Section 24.3.2 “NRWW – No Read-while-write Section” on page 242 and Section 24.3.1 “RWW – Read-while-write Section” on page 242.

Table 24-12. Explanation of Different Variables used in Figure 24-3 and the Mapping to the Z-pointer

Variable		Corresponding Z-value[1]	Description
PCMSB	13		Most significant bit in the program counter (the program counter is 14 bits PC[13:0])
PAGEMSB	5		Most significant bit which is used to address the words within one page (64 words in a page requires 6 bits PC [5:0]).
ZPCMSB		Z14	Bit in Z-register that is mapped to PCMSB. Because Z0 is not used, the ZPCMSB equals PCMSB + 1.
ZPAGEMSB		Z6	Bit in Z-register that is mapped to PAGEMSB. Because Z0 is not used, the ZPAGEMSB equals PAGEMSB + 1.
PCPAGE	PC[13:6]	Z14:Z7	Program counter page address: Page select, for page erase and page write
PCWORD	PC[5:0]	Z6:Z1	Program counter word address: Word select, for filling temporary buffer (must be zero during page write operation)

Note: 1. Z15:Z13: always ignored
Z0: should be zero for all SPM commands, byte select for the LPM instruction.
See Section 24.6 “Addressing the Flash during Self-Programming” on page 246 for details about the use of Z-pointer during self-programming.

24.7.16 ATmega16/32/64/M1/C1 - 64K - Flash Boot Loader Parameters

In Table 24-13 through Table 24-15 on page 254, the parameters used in the description of the self programming are given.

Table 24-13. Boot Size Configuration, ATmega16/32/64/M1/C1 (64K Product)

BOOTSZ1	BOOTSZ0	Boot Size[2]	Pages	Application Flash Section	Boot Loader Flash Section	End Application Section	Boot Reset Address (Start Boot Loader Section)
1	1	512 words	4	0x0000 - 0x7DFF	0x7E00 - 0x7FFF	0x7DFF	0x7E00
1	0	1024 words	8	0x0000 - 0x7BFF	0x7C00 - 0x7FFF	0x7BFF	0x7C00
0	1	2048 words	16	0x0000 - 0x77FF	0x7800 - 0x7FFF	0x77FF	0x7800
0	0	4096 words	32	0x0000 - 0x6FFF	0x7000 - 0x7FFF	0x6FFF	0x7000

Note: 1. The different BOOTSZ Fuse configurations are shown in Figure 24-2 on page 243.
2. 1 word equals 2 bytes.

26.9 ADC Characteristics

Table 26-6. ADC Characteristics in Single Ended Mode - T_A = –40°C to +125°C, V_{CC} = 2.7V to 5.5V (unless otherwise noted)

Parameter	Condition	Symbol	Min	Typ	Max	Unit
Resolution	Single Ended Conversion			10		Bits
Absolute accuracy	V_{CC} = 5V, V_{REF} = 2.56V ADC clock = 1MHz	TUE		3.2	5.0	LSB
	V_{CC} = 5V, V_{REF} = 2.56V ADC clock = 2MHz	TUE		3.2	5.0	LSB
Integral Non-linearity	V_{CC} = 5V, V_{REF} = 2.56V ADC clock = 1MHz	INL		0.7	1.5	LSB
	V_{CC} = 5V, V_{REF} = 2.56V ADC clock = 2MHz	INL		0.8	2.0	LSB
Differential Non-linearity	V_{CC} = 5V, V_{REF} = 2.56V ADC clock = 1MHz	DNL		0.5	0.8	LSB
	V_{CC} = 5V, V_{REF} = 2.56V ADC clock = 2MHz	DNL		0.6	1.4	LSB
Gain error	V_{CC} = 5V, V_{REF} = 2.56V ADC clock = 1MHz		–9.0	-5.0	0.0	LSB
	V_{CC} = 5V, V_{REF} = 2.56V ADC clock = 2MHz		–9.0	-5.0	0.0	LSB
Offset error	V_{CC} = 5V, V_{REF} = 2.56V ADC clock = 1MHz		–2.0	+2.5	+5.0	LSB
	V_{CC} = 5V, V_{REF} = 2.56V ADC clock = 2MHz		–2.0	+2.5	+5.0	LSB
Ref voltage		V_{REF}	2.56		AVCC	V

Figure 27-24. XTAL1 Input Threshold Voltage versus V_{CC} (XTAL1 Pin Read As '1')

Figure 27-25. XTAL1 Input Threshold Voltage versus V_{CC} (XTAL1 Pin Read As '0')

27.7 BOD Thresholds and Analog Comparator Hysteresis

Figure 27-26. BOD Thresholds versus Temperature (BODLEVEL Is 4.3V)