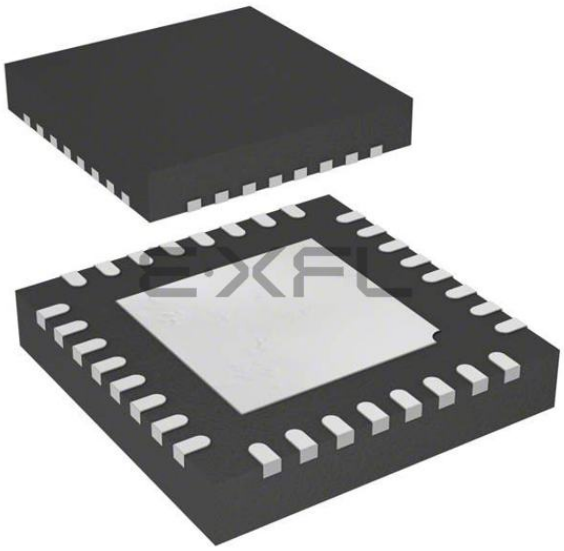


Welcome to E-XFL.COM

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-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	64KB (32K x 16)
Program Memory Type	FLASH
EEPROM Size	2K x 8
RAM Size	4K 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 ~ 150°C (TA)
Mounting Type	Surface Mount
Package / Case	32-VQFN Exposed Pad
Supplier Device Package	32-QFN (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atmega64m1-15md

1.1 Pin Descriptions

Table 1-1. Pin Out Description

QFN32 Pin Number	Mnemonic	Type	Name, Function and Alternate Function
5	GND	Power	**Ground:** 0V reference
20	AGND	Power	**Analog Ground:** 0V reference for analog part
4	VCC	Power	Power Supply
19	AVCC	Power	**Analog Power Supply:** This is the power supply voltage for analog part For a normal use this pin must be connected.
21	AREF	Power	**Analog Reference:** reference for analog converter. This is the reference voltage of the A/D converter. As output, can be used by external analog ISRC (Current Source Output)
8	PB0	I/O	MISO (SPI Master In Slave Out) PSCOUT2A (PSC Module 2 Output A) PCINT0 (Pin Change Interrupt 0)
9	PB1	I/O	MOSI (SPI Master Out Slave In) PSCOUT2B (PSC Module 2 Output B) PCINT1 (Pin Change Interrupt 1)
16	PB2	I/O	ADC5 (Analog Input Channel 5) INT1 (External Interrupt 1 Input) ACMPN0 (analog comparator 0 Negative Input) PCINT2 (Pin Change Interrupt 2)
23	PB3	I/O	AMP0- (Analog Differential Amplifier 0 Negative Input) PCINT3 (Pin Change Interrupt 3)
24	PB4	I/O	AMP0+ (Analog Differential Amplifier 0 Positive Input) PCINT4 (Pin Change Interrupt 4)
26	PB5	I/O	ADC6 (Analog Input Channel 6) INT2 (External Interrupt 2 Input) ACMPN1 (analog comparator 1 Negative Input) AMP2- (Analog Differential Amplifier 2 Negative Input) PCINT5 (Pin Change Interrupt 5)
27	PB6	I/O	ADC7 (Analog Input Channel 7) PSCOUT1B (PSC Module 1 Output A) PCINT6 (Pin Change Interrupt 6)
28	PB7	I/O	ADC4 (Analog Input Channel 4) PSCOUT0B (PSC Module 0 Output B) SCK (SPI Clock) PCINT7 (Pin Change Interrupt 7)
30	PC0	I/O	PSCOUT1A (PSC Module 1 Output A) INT3 (External Interrupt 3 Input) PCINT8 (Pin Change Interrupt 8)

Note: 1. On the first 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)

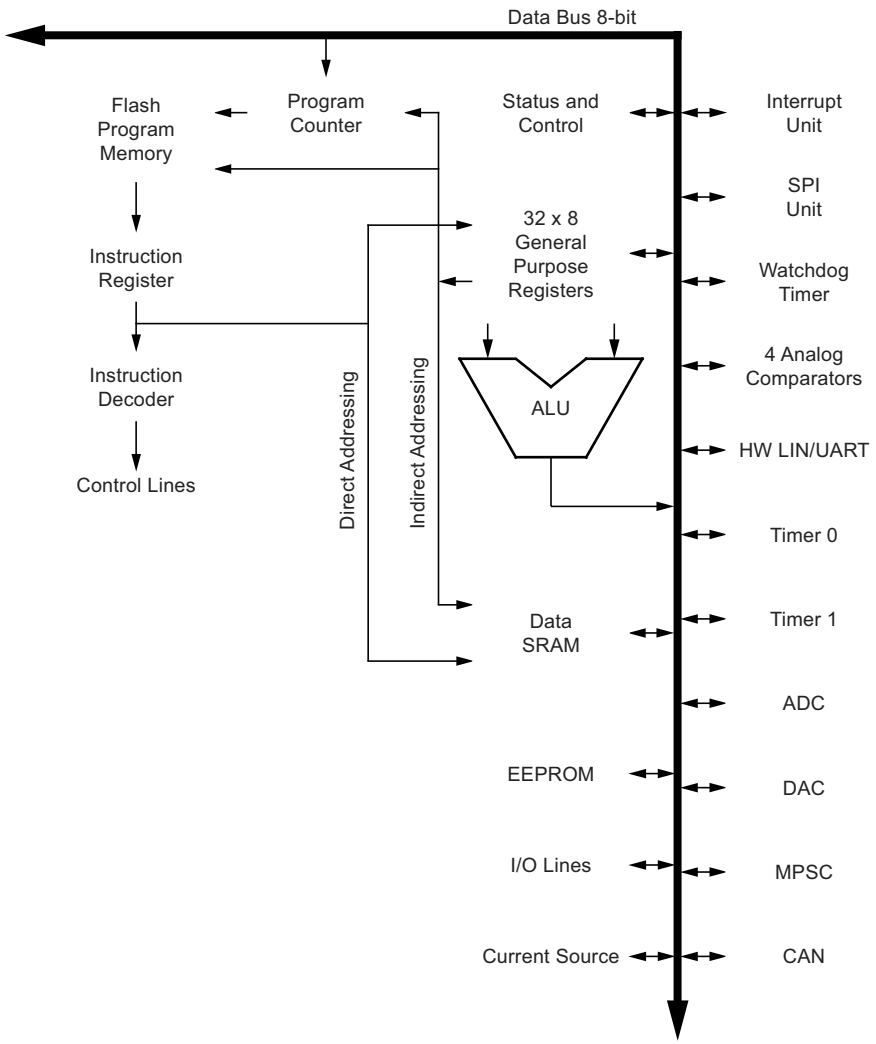
Note: 1. On the first engineering samples (Parts marked AT90PWM324), the ACMPN3 alternate function is not located on PC4. It is located on PE2.

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.

5.10.1 Clock Prescaler Register – CLKPR

Bit	7	6	5	4	3	2	1	0	
	CLKPCE	–	–	–	CLKPS3	CLKPS2	CLKPS1	CLKPS0	CLKPR
Read/Write	R/W	R	R	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	See Bit Description				

- **Bit 7 – CLKPCE: Clock Prescaler Change Enable**

The CLKPCE bit must be written to logic one to enable change of the CLKPS bits. The CLKPCE bit is only updated when the other bits in CLKPR are simultaneously written to zero. CLKPCE is cleared by hardware four cycles after it is written or when CLKPS bits are written. Rewriting the CLKPCE bit within this time-out period does neither extend the time-out period, nor clear the CLKPCE bit.

- **Bits 3..0 – CLKPS3..0: Clock Prescaler Select Bits 3 - 0**

These bits define the division factor between the selected clock source and the internal system clock. These bits can be written run-time to vary the clock frequency to suit the application requirements. As the divider divides the master clock input to the MCU, the speed of all synchronous peripherals is reduced when a division factor is used. The division factors are given in Table 5-10.

The CKDIV8 fuse determines the initial value of the CLKPS bits. If CKDIV8 is unprogrammed, the CLKPS bits will be reset to “0000”. If CKDIV8 is programmed, CLKPS bits are reset to “0011”, giving a division factor of 8 at start up. This feature should be used if the selected clock source has a higher frequency than the maximum frequency of the device at the present operating conditions. Note that any value can be written to the CLKPS bits regardless of the CKDIV8 fuse setting. The application software must ensure that a sufficient division factor is chosen if the selected clock source has a higher frequency than the maximum frequency of the device at the present operating conditions. The device is shipped with the CKDIV8 fuse programmed.

Table 5-10. Clock Prescaler Select

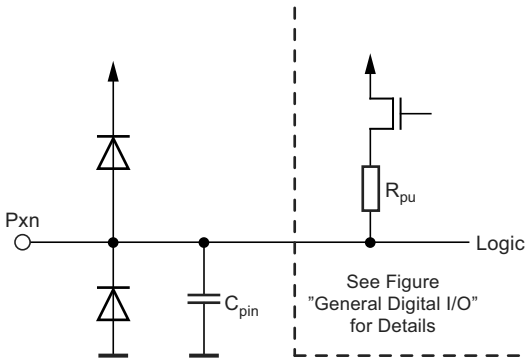
CLKPS3	CLKPS2	CLKPS1	CLKPS0	Clock Division Factor
0	0	0	0	1
0	0	0	1	2
0	0	1	0	4
0	0	1	1	8
0	1	0	0	16
0	1	0	1	32
0	1	1	0	64
0	1	1	1	128
1	0	0	0	256
1	0	0	1	Reserved
1	0	1	0	Reserved
1	0	1	1	Reserved
1	1	0	0	Reserved
1	1	0	1	Reserved
1	1	1	0	Reserved
1	1	1	1	Reserved

9. I/O-Ports

9.1 Introduction

All AVR® ports have true read-modify-write functionality when used as general digital I/O ports. This means that the direction of one port pin can be changed without unintentionally changing the direction of any other pin with the SBI and CBI instructions. The same applies when changing drive value (if configured as output) or enabling/disabling of pull-up resistors (if configured as input). Each output buffer has symmetrical drive characteristics with both high sink and source capability. All port pins have individually selectable pull-up resistors with a supply-voltage invariant resistance. All I/O pins have protection diodes to both V_{CC} and ground as indicated in Figure 9-1. Refer to Section 26. “Electrical Characteristics” on page 273 for a complete list of parameters.

Figure 9-1. I/O Pin Equivalent Schematic



All registers and bit references in this section are written in general form. A lower case “x” represents the numbering letter for the port, and a lower case “n” represents the bit number. However, when using the register or bit defines in a program, the precise form must be used. For example, PORTB3 for bit no. 3 in Port B, here documented generally as PORTxn. The physical I/O registers and bit locations are listed in “Register Description for I/O-Ports”.

Three I/O memory address locations are allocated for each port, one each for the data register – PORTx, data direction register – DDRx, and the port input pins – PINx. The port input pins I/O location is read only, while the data register and the data direction register are read/write. However, writing a logic one to a bit in the PINx register, will result in a toggle in the corresponding bit in the data register. In addition, the pull-up disable – PUD bit in MCUCR disables the pull-up function for all pins in all ports when set.

Using the I/O port as general digital I/O is described in “Ports as General Digital I/O”. Most port pins are multiplexed with alternate functions for the peripheral features on the device. How each alternate function interferes with the port pin is described in Section 9.3 “Alternate Port Functions” on page 55. Refer to the individual module sections for a full description of the alternate functions.

Note that enabling the alternate function of some of the port pins does not affect the use of the other pins in the port as general digital I/O.

Table 9-2 summarizes the function of the overriding signals. The pin and port indexes from Figure 9-5 on page 56 are not shown in the succeeding tables. The overriding signals are generated internally in the modules having the alternate function.

Table 9-2. Generic Description of Overriding Signals for Alternate Functions

Signal Name	Full Name	Description
PUOE	Pull-up override enable	If this signal is set, the pull-up enable is controlled by the PUOV signal. If this signal is cleared, the pull-up is enabled when {DDxn, PORTxn, PUD} = 0b010.
PUOV	Pull-up override value	If PUOE is set, the pull-up is enabled/disabled when PUOV is set/cleared, regardless of the setting of the DDxn, PORTxn, and PUD register bits.
DDOE	Data direction override enable	If this signal is set, the output driver enable is controlled by the DDOV signal. If this signal is cleared, the output driver is enabled by the DDxn register bit.
DDOV	Data direction override value	If DDOE is set, the output driver is enabled/disabled when DDOV is set/cleared, regardless of the setting of the DDxn register bit.
PVOE	Port value override enable	If this signal is set and the output driver is enabled, the port value is controlled by the PVOV signal. If PVOE is cleared, and the output driver is enabled, the port value is controlled by the PORTxn register bit.
PVOV	Port value override value	If PVOE is set, the port value is set to PVOV, regardless of the setting of the PORTxn register bit.
PTOE	Port toggle override enable	If PTOE is set, the PORTxn register bit is inverted.
DIEOE	Digital input enable override enable	If this bit is set, the digital input enable is controlled by the DIEOV signal. If this signal is cleared, the digital input enable is determined by MCU state (normal mode, sleep mode).
DIEOV	Digital input enable override value	If DIEOE is set, the digital Input is enabled/disabled when DIEOV is set/cleared, regardless of the MCU state (normal mode, sleep mode).
DI	Digital Input	This is the digital input to alternate functions. In the figure, the signal is connected to the output of the schmitt trigger but before the synchronizer. Unless the digital input is used as a clock source, the module with the alternate function will use its own synchronizer.
AIO	Analog input/output	This is the analog input/output to/from alternate functions. The signal is connected directly to the pad, and can be used bi-directionally.

The following subsections shortly describe the alternate functions for each port, and relate the overriding signals to the alternate function. Refer to the alternate function description for further details.

9.3.1 MCU Control Register – MCUCR

Bit	7	6	5	4	3	2	1	0	
	SPIPS	**–**	**–**	**PUD**	**–**	**–**	**IVSEL**	**IVCE**	**MCUCR**
Read/Write	R/W	R	R	R/W	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 4 – PUD: Pull-up Disable**

When this bit is written to one, the pull-ups in the I/O ports are disabled even if the DDxn and PORTxn registers are configured to enable the pull-ups ({DDxn, PORTxn} = 0b01).

9.3.2 Alternate Functions of Port B

The Port B pins with alternate functions are shown in Table 9-3.

Table 9-3. Port B Pins Alternate Functions

Port Pin	Alternate Functions
PB7	PSCOUT0B (PSC output 0B) ADC4 (Analog Input Channel 4) SCK (SPI Bus Serial Clock) PCINT7 (Pin Change Interrupt 7)
PB6	ADC7 (Analog Input Channel 7) PSCOUT1B (PSC output 1B) PCINT6 (Pin Change Interrupt 6)
PB5	ADC6 (Analog Input Channel 6) INT2 (External Interrupt 2) ACMPN1 (analog comparator 1 Negative Input) AMP2- (Analog Differential Amplicator 2 Negative Input) PCINT5 (Pin Change Interrupt 5)
PB4	AMP0+ (Analog Differential Amplifier 0 Positive Input) PCINT4 (Pin Change Interrupt 4)
PB3	AMP0- (Analog Differential Amplifier 0 Negative Input) PCINT3 (Pin Change Interrupt 3)
PB2	ADC5 (Analog Input Channel5) INT1 (External Interrupt 1) ACMPN0 (analog comparator 0 Negative Input) PCINT2 (Pin Change Interrupt 2)
PB1	MOSI (SPI Master Out Slave In) PSCOUT2B (PSC output 2B) PCINT1 (Pin Change Interrupt 1)
PB0	MISO (SPI Master In Slave Out) PSCOUT2A (PSC output 2A) PCINT0 (Pin Change Interrupt 0)

The alternate pin configuration is as follows:

- **ADC4/PSCOUT0B/SCK/PCINT7 – Bit 7**

PSCOUT0B, output 0B of PSC.

ADC4, analog to digital converter, input channel 4.

SCK, master clock output, slave clock input pin for SPI channel. When the SPI is enabled as a slave, this pin is configured as an input regardless of the setting of DDB7. When the SPI is enabled as a master, the data direction of this pin is controlled by DDB7. When the pin is forced to be an input, the pull-up can still be controlled by the PORTB7 bit.

PCINT7, pin change interrupt 7.

- **ADC7/PSCOUT1B/PCINT6 – Bit 6**

ADC7, analog to digital converter, input channel 7.

PSCOUT1B, output 1B of PSC.

PCINT6, pin change interrupt 6.

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)

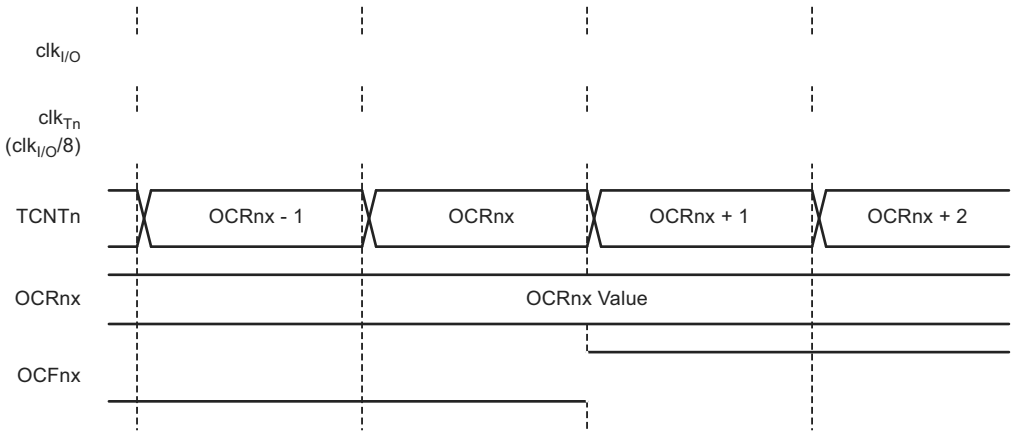
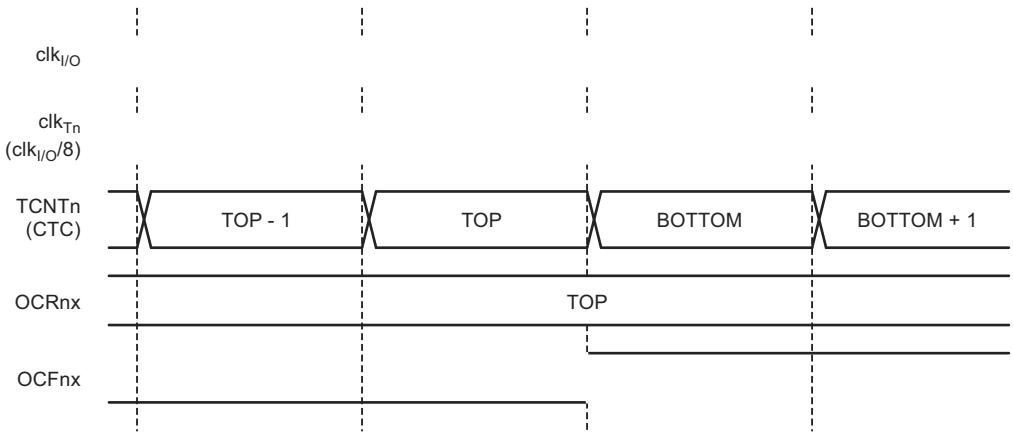


Figure 12-11 shows the setting of OCF0A and the clearing of TCNT0 in CTC mode and fast PWM mode where OCR0A is TOP.

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).

12.8.2 Timer/Counter Control Register B – TCCR0B

Bit	7	6	5	4	3	2	1	0	
	FOC0A	FOC0B	–	–	WGM02	CS02	CS01	CS00	TCCR0B
Read/Write	W	W	R	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – FOC0A: Force Output Compare A**

The FOC0A bit is only active when the WGM bits specify a non-PWM mode.

However, for ensuring compatibility with future devices, this bit must be set to zero when TCCR0B is written when operating in PWM mode. When writing a logical one to the FOC0A bit, an immediate compare match is forced on the waveform generation unit. The OC0A output is changed according to its COM0A1:0 bits setting. Note that the FOC0A bit is implemented as a strobe. Therefore it is the value present in the COM0A1:0 bits that determines the effect of the forced compare.

A FOC0A strobe will not generate any interrupt, nor will it clear the timer in CTC mode using OCR0A as TOP.

The FOC0A bit is always read as zero.

- **Bit 6 – FOC0B: Force Output Compare B**

The FOC0B bit is only active when the WGM bits specify a non-PWM mode.

However, for ensuring compatibility with future devices, this bit must be set to zero when TCCR0B is written when operating in PWM mode. When writing a logical one to the FOC0B bit, an immediate compare match is forced on the waveform generation unit. The OC0B output is changed according to its COM0B1:0 bits setting. Note that the FOC0B bit is implemented as a strobe. Therefore it is the value present in the COM0B1:0 bits that determines the effect of the forced compare.

A FOC0B strobe will not generate any interrupt, nor will it clear the timer in CTC mode using OCR0B as TOP.

The FOC0B bit is always read as zero.

- **Bits 5:4 – Res: Reserved Bits**

These bits are reserved bits in the ATmega16/32/64/M1/C1 and will always read as zero.

- **Bit 3 – WGM02: Waveform Generation Mode**

See the description in Section 12.8.1 “Timer/Counter Control Register A – TCCR0A” on page 86.

- **Bits 2:0 – CS02:0: Clock Select**

The three clock select bits select the clock source to be used by the Timer/Counter.

Table 12-9. Clock Select Bit Description

CS02	CS01	CS00	Description
0	0	0	No clock source (Timer/Counter stopped)
0	0	1	$clk_{I/O}$/(no prescaling)
0	1	0	$clk_{I/O}$/8 (from prescaler)
0	1	1	$clk_{I/O}$/64 (from prescaler)
1	0	0	$clk_{I/O}$/256 (from prescaler)
1	0	1	$clk_{I/O}$/1024 (from prescaler)
1	1	0	External clock source on T0 pin. Clock on falling edge.
1	1	1	External clock source on T0 pin. Clock on rising edge.

If external pin modes are used for the Timer/Counter0, transitions on the T0 pin will clock the counter even if the pin is configured as an output. This feature allows software control of the counting.

- **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)

14.16.7 PSC Configuration Register – PCNF

Bit	7	6	5	4	3	2	1	0	
	-	-	PULOCK	PMODE	POPB	POPA	-	-	PCNF
Read/Write	R	R	R/W	R/W	R/W	R/W	R	R	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7:6 - not use**

not use

- **Bit 5 – PULOCK: PSC Update Lock**

When this bit is set, the output compare registers POCRnRA, POCRnSA, POCRnSB, POCR_RB and the PSC output configuration registers POC can be written without disturbing the PSC cycles. The update of the PSC internal registers will be done if the PULOCK bit is released to zero.

- **Bit 4 – PMODE PSC Mode**

Select the mode of PSC.

Table 14-10. PSC Mode Selection

PMODE	Description
0	One ramp mode (edge aligned)
1	Center aligned mode

- **Bit 3 – POPB: PSC B Output Polarity**

If this bit is cleared, the PSC outputs B are active low.

If this bit is set, the PSC outputs B are active high.

- **Bit 2 – POPA: PSC A Output Polarity**

If this bit is cleared, the PSC outputs A are active low.

If this bit is set, the PSC outputs A are active high.

- **Bit 1:0 – not use**

not use

14.16.8 PSC Control Register – PCTL

Bit	7	6	5	4	3	2	1	0	
	PPRE1	PPRE0	PCLKSEL	SWAP2	SWAP1	SWAP0	PCCYC	PRUN	PCTL
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 7:6 – PPRE1:0 : PSC Prescaler Select**

This two bits select the PSC input clock division factor. All generated waveform will be modified by this factor.

Table 14-11. PSC Prescaler Selection

PPRE1	PPRE0	Description
0	0	No divider on PSC input clock
0	1	Divide the PSC input clock by 4
1	0	Divide the PSC input clock by 32
1	1	Divide the PSC clock by 256

- **Bit 5 – PCLKSEL: PSC Input Clock Select**

This bit is used to select between CLK_{PLL} or CLK_{IO} clocks.

Set this bit to select the fast clock input (CLK_{PLL}). Clear this bit to select the slow clock input (CLK_{IO}).

The following code examples show how to initialize the SPI as a master and how to perform a simple transmission.

DDR_SPI in the examples must be replaced by the actual data direction register controlling the SPI pins. DD_MOSI, DD_MISO and DD_SCK must be replaced by the actual data direction bits for these pins. E.g. if MOSI is placed on pin PB2, replace DD_MOSI with DDB2 and DDR_SPI with DDRB.

Assembly Code Example[1]

```
SPI_MasterInit:
       ; Set MOSI and SCK output, all others input
       ldi                 r17,(1<<DD_MOSI)|(1<<DD_SCK)
       out                 DDR_SPI,r17
       ; Enable SPI, Master, set clock rate fck/16
       ldi                 r17,(1<<SPE)|(1<<MSTR)|(1<<SPR0)
       out                 SPCR,r17
       ret

SPI_MasterTransmit:
       ; Start transmission of data (r16)
       out                 SPDR,r16
Wait_Transmit:
       ; Wait for transmission complete
       sbis                SPSR,SPIF
       rjmp                Wait_Transmit
       ret
```

C Code Example[1]

```
void SPI_MasterInit(void)
{
       /* Set MOSI and SCK output, all others input */
       DDR_SPI = (1<<DD_MOSI)|(1<<DD_SCK);
       /* Enable SPI, Master, set clock rate fck/16 */
       SPCR = (1<<SPE)|(1<<MSTR)|(1<<SPR0);
}

void SPI_MasterTransmit(char cData)
{
       /* Start transmission */
       SPDR = cData;
       /* Wait for transmission complete */
       while(!(SPSR & (1<<SPIF)))
       ;
}
```

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

16.4 CAN Channel

16.4.1 Configuration

The CAN channel can be in:

- Enabled mode

  In this mode:

  - the CAN channel (internal TxCAN and RxCAN) is enabled,
  - the input clock is enabled.
- Standby mode

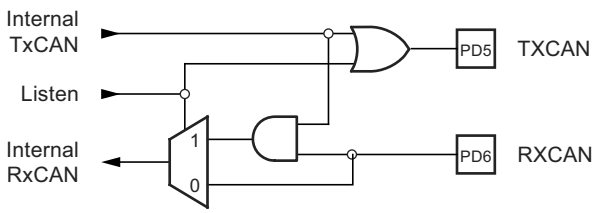
  In standby mode:

  - the transmitter constantly provides a recessive level (on internal TxCAN) and the receiver is disabled,
  - input clock is enabled,
  - the registers and pages remain accessible.
- Listening mode

  This mode is transparent for the CAN channel:

  - enables a hardware loop back, internal TxCAN on internal RxCAN
  - provides a recessive level on TXCAN output pin
  - does not disable RXCAN input pin
  - freezes TEC and REC error counters

Figure 16-6. Listening Mode



16.4.2 Bit Timing

FSM's (finite state machine) of the CAN channel need to be synchronous to the time quantum. So, the input clock for bit timing is the clock used into CAN channel FSM's.

Field and segment abbreviations:

- BRP: Baud rate prescaler.
- TQ: Time quantum (output of baud rate prescaler).
- SYNS: Synchronization segment is 1 TQ long.
- PRS: Propagation time segment is programmable to be 1, 2, ..., 8 TQ long.
- PHS1: Phase segment 1 is programmable to be 1, 2, ..., 8 TQ long.
- PHS2: Phase segment 2 is programmable to be ≤ PHS1 and ≥ INFORMATION PROCESSING TIME.
- INFORMATION PROCESSING TIME is 2 TQ.
- SJW: (Re) Synchronization jump width is programmable between 1 and min(4, PHS1).

The total number of TQ in a bit time has to be programmed at least from 8 to 25.

17.6.2 LIN Status and Interrupt Register - LINSIR

Bit	7	6	5	4	3	2	1	0	
	LIDST2	LIDST1	LIDST0	LBUSY	LERR	LIDOK	LTXOK	LRXOK	LINSIR
Read/Write	R	R	R	R	R/W_{one}	R/W_{one}	R/W_{one}	R/W_{one}	
Initial Value	0	0	0	0	0	0	0	0	

- **Bits 7:5 - LIDST[2:0]: Identifier Status**
  - 0xx = no specific identifier,
  - 100 = Identifier 60 (0x3C),
  - 101 = Identifier 61 (0x3D),
  - 110 = Identifier 62 (0x3E),
  - 111 = Identifier 63 (0x3F).

- **Bit 4 - LBUSY: Busy Signal**
  - 0 = Not busy,
  - 1 = Busy (receiving or transmitting).

- **Bit 3 - LERR: Error Interrupt**
  It is a logical OR of LINERR register bits. This bit generates an interrupt if its respective enable bit - LENERR - is set in LINENIR.
  - 0 = No error,
  - 1 = An error has occurred.

  The user clears this bit by writing 1 in order to reset this interrupt. Resetting LERR also resets all LINERR bits. In UART mode, this bit is also cleared by reading LINDAT.

- **Bit 2 - LIDOK: Identifier Interrupt**
  This bit generates an interrupt if its respective enable bit - LENIDOK - is set in LINENIR.
  - 0 = No identifier,
  - 1 = Slave task: Identifier present, master task: Tx header complete.

  The user clears this bit by writing 1, in order to reset this interrupt.

- **Bit 1 - LTXOK: Transmit Performed Interrupt**
  This bit generates an interrupt if its respective enable bit - LENTXOK - is set in LINENIR.
  - 0 = No Tx,
  - 1 = Tx Response complete.

  The user clears this bit by writing 1, in order to reset this interrupt.
  In UART mode, this bit is also cleared by writing LINDAT.

- **Bit 0 - LRXOK: Receive Performed Interrupt**
  This bit generates an interrupt if its respective enable bit - LENRXOK - is set in LINENIR.
  - 0 = No Rx
  - 1 = Rx Response complete.

  The user clears this bit by writing 1, in order to reset this interrupt.
  In UART mode, this bit is also cleared by reading LINDAT.

19.2.4 Threshold Reference for Internal analog comparator

An external resistor used in conjunction with the Current Source can be used as threshold reference for internal analog comparator (see Section 20. “Analog Comparator” on page 225). This can be connected to AIN0 (negative analog compare input pin) as well as AIN1 (positive analog compare input pin). Using a resistor in series with a lower tolerance than the current source accuracy (≤ 2%) is recommended. Table 19-2 gives an example of threshold references using standard values of resistors.

19.3 Control Register

19.3.1 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 6 – ISRCEN: Current Source Enable**

Set this bit to source a 100µA current to the AREF pin.
Clear this bit to disconnect.

- **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.

21.4.2.2 DALA = 1

Bit	7	6	5	4	3	2	1	0	
	DAC9	**DAC8**	**DAC7**	**DAC6**	**DAC5**	**DAC4**	**DAC3**	**DAC2**	**DACH**
	DAC1	**DAC0**	**-**	**-**	**-**	**-**	**-**	**-**	**DACL**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
	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	
	0	0	0	0	0	0	0	0	

To work with the 10-bit DAC, two registers have to be updated. In order to avoid intermediate value, the DAC input values which are really converted into analog signal are buffered into unreachable registers. In normal mode, the update of the shadow register is done when the register DACH is written.

In case DAATE bit is set, the DAC input values will be updated on the trigger event selected through DATS bits.
In order to avoid wrong DAC input values, the update can only be done after having written respectively DACL and DACH registers. It is possible to work on 8-bit configuration by only writing the DACH value. In this case, update is done each trigger event.

In case DAATE bit is cleared, the DAC is in an automatic update mode. Writing the DACH register automatically update the DAC input values with the DACH and DACL register values.

It means that whatever is the configuration of the DAATE bit, changing the DACL register has no effect on the DAC output until the DACH register has also been updated. So, to work with 10 bits, DACL must be written first before DACH. To work with 8-bit configuration, writing DACH allows the update of the DAC.

23.4 Software Break Points

debugWIRE supports program memory break points by the AVR® break instruction. Setting a break point in AVR Studio® will insert a BREAK instruction in the program memory. The instruction replaced by the BREAK instruction will be stored. When program execution is continued, the stored instruction will be executed before continuing from the program memory. A break can be inserted manually by putting the BREAK instruction in the program.

The flash must be re-programmed each time a break point is changed. This is automatically handled by AVR Studio through the debugWIRE interface. The use of break points will therefore reduce the flash data retention. Devices used for debugging purposes should not be shipped to end customers.

23.5 Limitations of debugWIRE

The debugWIRE communication pin (dW) is physically located on the same pin as external reset (RESET). An external reset source is therefore not supported when the debugWIRE is enabled.

The debugWIRE system accurately emulates all I/O functions when running at full speed, i.e., when the program in the CPU is running. When the CPU is stopped, care must be taken while accessing some of the I/O registers via the debugger (AVR Studio).

A programmed DWEN fuse enables some parts of the clock system to be running in all sleep modes. This will increase the power consumption while in sleep. Thus, the DWEN Fuse should be disabled when debugWire is not used.

23.6 debugWIRE Related Register in I/O Memory

The following section describes the registers used with the debugWire.

23.6.1 debugWire Data Register – DWDR

Bit	7	6	5	4	3	2	1	0	
	DWDR[7:0]								DWDR
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	

The DWDR register provides a communication channel from the running program in the MCU to the debugger. This register is only accessible by the debugWIRE and can therefore not be used as a general purpose register in the normal operations.

25.2 Fuse Bits

The ATmega16/32/64/M1/C1 has three Fuse bytes. Table 25-4 to Table 25-7 on page 257 describe briefly the functionality of all the fuses and how they are mapped into the Fuse bytes. Note that the fuses are read as logical zero, “0”, if they are programmed.

Table 25-4. Extended Fuse Byte

Extended Fuse Byte	Bit No	Description	Default Value
-	7	-	1 (unprogrammed)
-	6	-	1 (unprogrammed)
PSCRB	5	PSC reset behavior	1 (unprogrammed)
PSCRVA	4	PSCOUTnA reset value	1 (unprogrammed)
PSCRVB	3	PSCOUTnB reset value	1 (unprogrammed)
BODLEVEL2(1)	2	Brown-out detector trigger level	1 (unprogrammed)
BODLEVEL1(1)	1	Brown-out detector trigger level	1 (unprogrammed)
BODLEVEL0(1)	0	Brown-out detector trigger level	1 (unprogrammed)

Note: 1. See Table 7-2 on page 40 for BODLEVEL fuse decoding.

25.3 PSC Output Behavior during Reset

For external component safety reason, the state of PSC outputs during reset can be programmed by fuses PSCRB, PSCARV and PSCBRV. These fuses are located in the extended fuse byte (see Table 25-4 on page 256).

If PSCRB fuse equals 1 (unprogrammed), all PSC outputs keep a standard port behavior. If PSC0RB fuse equals 0 (programmed), all PSC outputs are forced at reset to low level or high level according to PSCARV and PSCBRV fuse bits. In this second case, the PSC outputs keep the forced state until POC register is written. Section 5.10.1 “Clock Prescaler Register – CLKPR” on page 33

PSCARV (PSCOUTnA reset value) gives the state low or high which will be forced on PSCOUT0A, PSCOUT1A and PSCOUT2A outputs when PSCRB is programmed. If PSCARV fuse equals 0 (programmed), the PSCOUT0A, PSCOUT1A and PSCOUT2A outputs will be forced to high state. If PSCRV fuse equals 1 (unprogrammed), the PSCOUT0A, PSCOUT1A and PSCOUT2A outputs will be forced to low state.

PSCBRV (PSCOUTnB Reset Value) gives the state low or high which will be forced on PSCOUT0B, PSCOUT1B and PSCOUT2B outputs when PSCRB is programmed. If PSCBRV fuse equals 0 (programmed), the PSCOUT0B, PSCOUT1B and PSCOUT2B outputs will be forced to high state. If PSCRV fuse equals 1 (unprogrammed), the PSCOUT0B, PSCOUT1B and PSCOUT2B outputs will be forced to low state.

Table 25-5. PSC Output Behavior during and after Reset until POC Register is Written

PSCRB	PSCARV	PSCBRV	PSCOUTnA	PSCOUTnB
Unprogrammed	X	X	Normal port	Normal port
Programmed	Unprogrammed	Unprogrammed	Forced low	Forced low
Programmed	Unprogrammed	Programmed	Forced low	Forced high
Programmed	Programmed	Unprogrammed	Forced high	Forced low
Programmed	Programmed	Programmed	Forced high	Forced high
BODLEVEL2(1)		2	Brown-out detector trigger level	1 (unprogrammed)
BODLEVEL1(1)		1	Brown-out detector trigger level	1 (unprogrammed)
BODLEVEL0(1)		0	Brown-out detector trigger level	1 (unprogrammed)