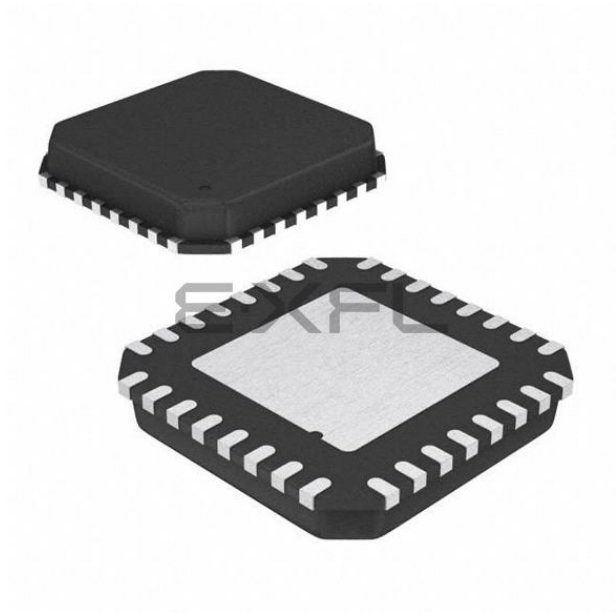


Welcome to E-XFL.COM

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	I²C, LINbus, SPI, UART/USART, USI
Peripherals	Brown-out Detect/Reset, POR, PWM, Temp Sensor, WDT
Number of I/O	16
Program Memory Size	8KB (4K x 16)
Program Memory Type	FLASH
EEPROM Size	512 x 8
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 5.5V
Data Converters	A/D 11x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	32-VFQFN Exposed Pad
Supplier Device Package	32-VQFN (5x5)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/attiny87-mu

1.4 Pin Configuration

Figure 1-2. Pinout ATtiny87/167 - SOIC20 & TSSOP20

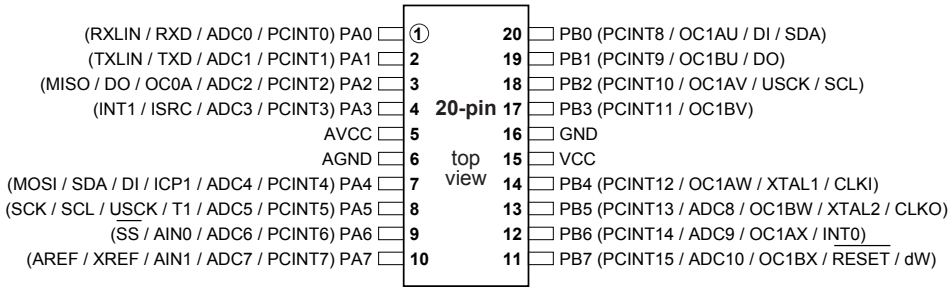
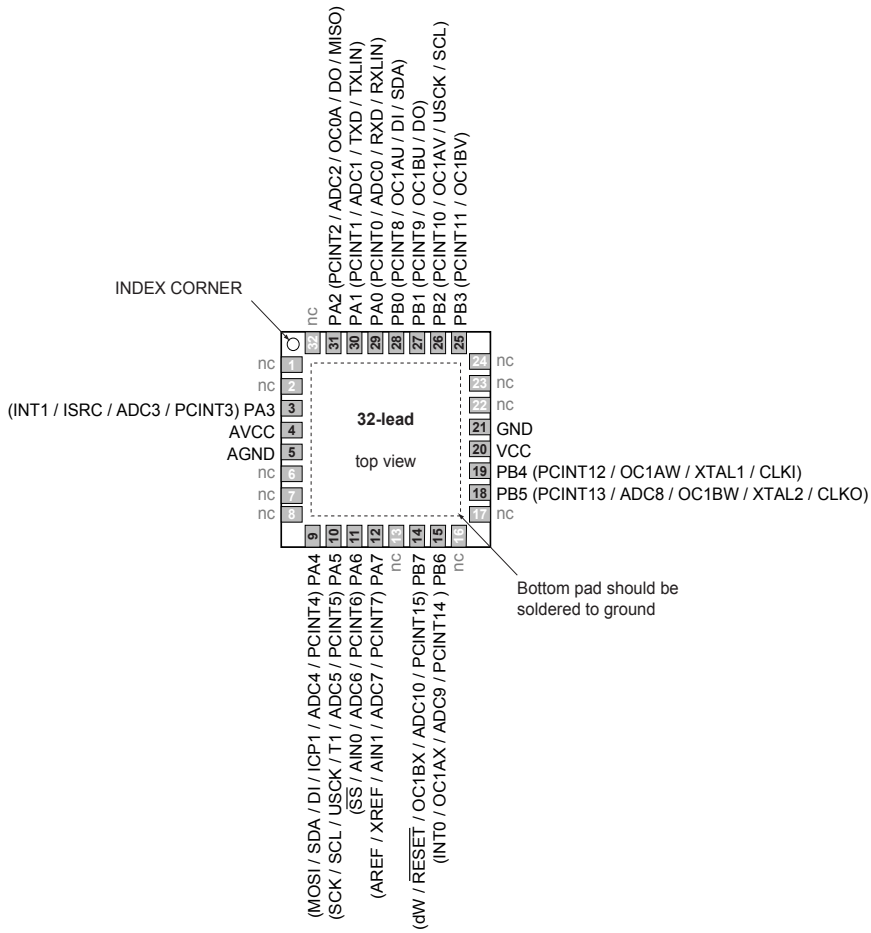


Figure 1-3. Pinout ATtiny87/167 - QFN32/MLF32



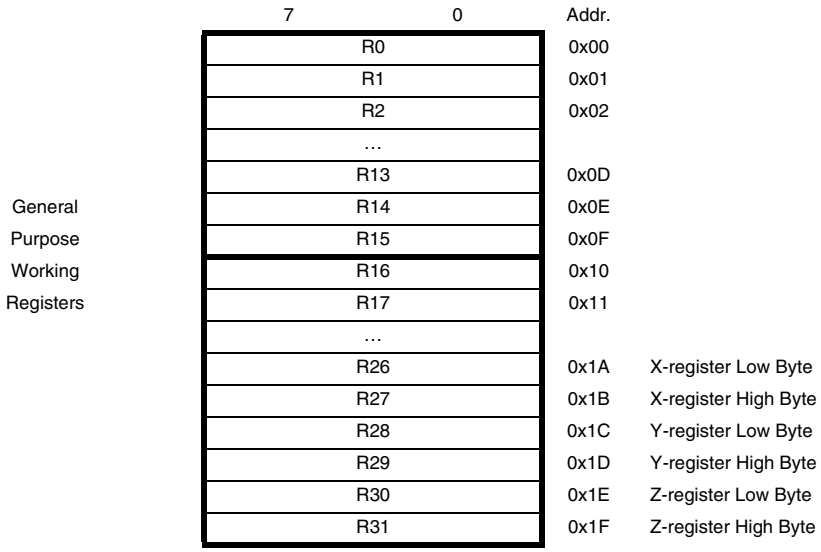
2.4 General Purpose Register File

The Register File is optimized for the AVR Enhanced RISC instruction set. In order to achieve the required performance and flexibility, the following input/output schemes are supported by the Register File:

- One 8-bit output operand and one 8-bit result input
- Two 8-bit output operands and one 8-bit result input
- Two 8-bit output operands and one 16-bit result input
- One 16-bit output operand and one 16-bit result input

Figure 2-2 shows the structure of the 32 general purpose working registers in the CPU.

Figure 2-2. AVR CPU General Purpose Working Registers



Most of the instructions operating on the Register File have direct access to all registers, and most of them are single cycle instructions.

As shown in Figure 2-2, each register is also assigned a Data memory address, mapping them directly into the first 32 locations of the user Data Space. Although not being physically implemented as SRAM locations, this memory organization provides great flexibility in access of the registers, as the X-, Y- and Z-pointer registers can be set to index any register in the file.

2.4.1 The X-register, Y-register, and Z-register

The registers R26:R31 have some added functions to their general purpose usage. These registers are 16-bit address pointers for indirect addressing of the data space. The three indirect address registers X, Y, and Z are defined as described in Figure 2-3 on page 11.

Here is a “light” C-code that describes such a sequence of commands.

C Code Example

```
void ClockSwiching (unsigned char clk_number, unsigned char sut) {

#define CLOCK_RECOVER   0x05
#define CLOCK_ENABLE    0x02
#define CLOCK_SWITCH    0x04
#define CLOCK_DISABLE   0x01

unsigned char previous_clk, temp;

  // Disable interrupts
    temp = SREG; asm ("cli");
  // Save the current system clock source
    CLKCSR = 1 << CLKCCE;
    CLKCSR = CLOCK_RECOVER;
    previous_clk = CLKSELR & 0x0F;
  // Enable the new clock source
    CLKSELR = ((sut << 4 ) & 0x30) | (clk_number & 0x0F);
    CLKCSR = 1 << CLKCCE;
    CLKCSR = CLOCK_ENABLE;
  // Wait for clock validity
    while ((CLKCSR & (1 << CLKRDY)) == 0);
  // Switch clock source
    CLKCSR = 1 << CLKCCE;
    CLKCSR = CLOCK_SWITCH;
  // Wait for effective switching
    while (1){
        CLKCSR = 1 << CLKCCE;
        CLKCSR = CLOCK_RECOVER;
        if ((CLKSELR & 0x0F) == (clk_number & 0x0F)) break;
    }
  // Shut down unneeded clock source
    if (previous_clk != (clk_number & 0x0F)) {
        CLKSELR = previous_clk;
        CLKCSR = 1 << CLKCCE;
        CLKCSR = CLOCK_DISABLE;
    }
  // Re-enable interrupts
    SREG = temp;
}
```

Warning:

In the ATtiny87/167, only one among the three external clock sources can be enabled at a given time. Moreover, the enables of the external clock and of the external low-frequency oscillator are shared with the asynchronous timer.

4.3.8 Clock Monitoring

A safe system needs to monitor its clock sources. Two domains need to be monitored:
- Clock sources for peripherals,
- Clocks sources for system clock generation.

In the first domain, the user (code) can easily check the validity of the clock(s) (See “COUT Command” on page 33.).

5.2 BOD Disable

When the Brown-out Detector (BOD) is enabled by BODLEVEL fuses, Table 21-3 on page 225, the BOD is actively monitoring the power supply voltage during a sleep period. To save power, it is possible to disable the BOD by software for some of the sleep modes, see Table 5-1. The sleep mode power consumption will then be at the same level as when BOD is globally disabled by fuses. If BOD is disabled in software, the BOD function is turned off immediately after entering the sleep mode. Upon wake-up from sleep, BOD is automatically enabled again. This ensures safe operation in case the V_{CC} level has dropped during the sleep period.

When the BOD has been disabled, the wake-up time from sleep mode will be approximately 60 µs to ensure that the BOD is working correctly before the MCU continues executing code.

BOD disable is controlled by BODS bit (BOD Sleep) in the control register MCUCR, see "MCUCR – MCU Control Register" on page 47. Setting it to one turns off the BOD in relevant sleep modes, while a zero in this bit keeps BOD active. Default setting keeps BOD active, i.e. BODS is cleared to zero.

Writing to the BODS bit is controlled by a timed sequence and an enable bit, see "MCUCR – MCU Control Register" on page 47.

5.3 Idle Mode

When the SM[1:0] bits are written to 00, the SLEEP instruction makes the MCU enter Idle mode, stopping the CPU but allowing the SPI, Analog Comparator, ADC, USI start condition, Asynchronous Timer/Counter, Watchdog, and the interrupt system to continue operating. This sleep mode basically halts clk_{CPU} and clk_{FLASH}, while allowing the other clocks to run.

Idle mode enables the MCU to wake up from external triggered interrupts as well as internal ones like the SPI interrupts. If wake-up from the Analog Comparator interrupt is not required, the Analog Comparator can be powered down by setting the ACD bit in the Analog Comparator Control and Status Register – ACSR. This will reduce power consumption in Idle mode. If the ADC is enabled, a conversion starts automatically when this mode is entered.

5.4 ADC Noise Reduction Mode

When the SM[1:0] bits are written to 01, the SLEEP instruction makes the MCU enter ADC Noise Reduction mode, stopping the CPU but allowing the ADC, the external interrupts, the USI start condition, the asynchronous Timer/Counter and the Watchdog to continue operating (if enabled). This sleep mode basically halts $clk_{I/O}$, clk_{CPU}, and clk_{FLASH}, while allowing the other clocks to run.

This improves the noise environment for the ADC, enabling higher resolution measurements. If the ADC is enabled, a conversion starts automatically when this mode is entered. Apart from the ADC Conversion Complete interrupt, only an External Reset, a Watchdog System Reset, a Watchdog Interrupt, a Brown-out Reset, a USI start condition interrupt, an asynchronous Timer/Counter interrupt, an SPM/EEPROM ready interrupt, an external level interrupt on INT0 or INT1 or a pin change interrupt can wake up the MCU from ADC Noise Reduction mode.

5.8.6 Watchdog Timer

If the Watchdog Timer is not needed in the application, the module should be turned off. If the Watchdog Timer is enabled, it will be enabled in all sleep modes and hence always consume power. In the deeper sleep modes, this will contribute significantly to the total current consumption. Refer to Section 6.3 “Watchdog Timer” on page 53 for details on how to configure the Watchdog Timer.

5.8.7 Port Pins

When entering a sleep mode, all port pins should be configured to use minimum power. The most important is then to ensure that no pins drive resistive loads. In sleep modes where both the I/O clock ($clk_{I/O}$) and the ADC clock (clk_{ADC}) are stopped, the input buffers of the device will be disabled. This ensures that no power is consumed by the input logic when not needed. In some cases, the input logic is needed for detecting wake-up conditions, and it will then be enabled. Refer to the section Section 9.2.6 “Digital Input Enable and Sleep Modes” on page 72 for details on which pins are enabled. If the input buffer is enabled and the input signal is left floating or have an analog signal level close to $V_{CC}/2$, the input buffer will use excessive power.

For analog input pins, the digital input buffer should be disabled at all times. An analog signal level close to $V_{CC}/2$ on an input pin can cause significant current even in active mode. Digital input buffers can be disabled by writing to the Digital Input Disable Registers (DIDR1 and DIDR0). Refer to Section 17.11.6 “DIDR1 – Digital Input Disable Register 1” on page 209 and Section 17.11.5 “DIDR0 – Digital Input Disable Register 0” on page 208 for details.

5.8.8 On-chip Debug System

If the On-chip debug system is enabled by the DWEN Fuse and the chip enters sleep mode, the main clock source is enabled and hence always consumes power. In the deeper sleep modes, this will contribute significantly to the total current consumption.

5.9 Register Description

5.9.1 SMCR – Sleep Mode Control Register

The Sleep Mode Control Register contains control bits for power management.

Bit	7	6	5	4	3	2	1	0	
0x33 (0x53)	–	–	–	–	–	SM1	SM0	SE	SMCR
Read/Write	R	R	R	R	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bits 7:3 – Res: Reserved Bits**

These bits are reserved bits in the ATtiny87/167 and will always read as zero.

- **Bits 2:1 – SM[1:0]: Sleep Mode Select Bits 1 and 0**

These bits select between the four available sleep modes as shown in Table 5-2.

- **Bit 4 – PRSPI: Power Reduction Serial Peripheral Interface**

If using debugWIRE On-chip Debug System, this bit should not be written to one.

Writing a logic one to this bit shuts down the Serial Peripheral Interface by stopping the clock to the module. When waking up the SPI again, the SPI should be re initialized to ensure proper operation.

- **Bit 3 – PRTIM1: Power Reduction Timer/Counter1**

Writing a logic one to this bit shuts down the Timer/Counter1 module. When the Timer/Counter1 is enabled, operation will continue like before the shutdown.

- **Bit 2 – PRTIM0: Power Reduction Timer/Counter0**

Writing a logic one to this bit shuts down the Timer/Counter0 module in synchronous mode (AS0 is 0). When the Timer/Counter0 is enabled, operation will continue like before the shutdown.

- **Bit 1 – PRUSI: Power Reduction USI**

Writing a logic one to this bit shuts down the USI by stopping the clock to the module. When waking up the USI again, the USI should be re-initialized to ensure proper operation.

- **Bit 0 – PRADC: Power Reduction ADC**

Writing a logic one to this bit shuts down the ADC. The ADC must be disabled before shut down. The analog comparator cannot use the ADC input MUX when the ADC is shut down.

- **Bit 7:0 – PCINT[7:0]: Pin Change Enable Mask 7:0**

Each PCINT[7:0] bit selects whether pin change interrupt is enabled on the corresponding I/O pin. If PCINT[7:0] is set and the PCIE0 bit in EIMSK is set, pin change interrupt is enabled on the corresponding I/O pin. If PCINT[7:0] is cleared, pin change interrupt on the corresponding I/O pin is disabled.

- $\overline{\text{RESET}}$: Reset input 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 PB7 is used as a reset pin, DDB7, PORTB7 and PINB7 will all read 0.
- dW: When the debugWIRE Enable (DWEN) Fuse is programmed and Lock bits are unprogrammed, the RESET port pin is configured as a wire-AND (open-drain) bi-directional I/O pin with pull-up enabled and becomes the communication gateway between target and emulator.

- **Port B, Bit 6 – PCINT14/ADC9/OC1AX/INT0**
  - PCINT14: Pin Change Interrupt, source 14.
  - ADC9: Analog to Digital Converter, channel 9.
  - OC1AX: Output Compare and PWM Output A-X for Timer/Counter1. The PB6 pin has to be configured as an output (DDB6 set (one)) to serve this function. The OC1AX pin is also the output pin for the PWM mode timer function (c.f. OC1AX bit of TCCR1D register).
  - INT0: External Interrupt0 Input. The PB6 pin can serve as an external interrupt source.

- **Port B, Bit 5 – PCINT13/ADC8/OC1BW/XTAL2/CLKO**
  - PCINT13: Pin Change Interrupt, source 13.
  - ADC8: Analog to Digital Converter, channel 8.
  - OC1BW: Output Compare and PWM Output B-W for Timer/Counter1. The PB5 pin has to be configured as an output (DDB5 set (one)) to serve this function. The OC1BW pin is also the output pin for the PWM mode timer function (c.f. OC1BW bit of TCCR1D register).
  - 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.
  - CLKO: Divided system clock output. The divided system clock can be output on the PB5 pin. The divided system clock will be output if the CKOUT Fuse is programmed, regardless of the PORTB5 and DDB5 settings. It will also be output during reset.

- **Port B, Bit 4 – PCINT12/OC1AW/XTAL1/CLKI**
  - PCINT12: Pin Change Interrupt, source 12.
  - OC1AW: Output Compare and PWM Output A-W for Timer/Counter1. The PB4 pin has to be configured as an output (DDB4 set (one)) to serve this function. The OC1AW pin is also the output pin for the PWM mode timer function (c.f. OC1AW bit of TCCR1D register).
  - 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.
  - CLKI: External clock input. When used as a clock pin, the pin can not be used as an I/O pin.

  Note: If PB4 is used as a clock pin (XTAL1 or CLKI), DDB4, PORTB4 and PINB4 will all read 0.

- **Port B, Bit 3 – PCINT11/OC1BV**
  - PCINT11: Pin Change Interrupt, source 11.
  - OC1BV: Output Compare and PWM Output B-V for Timer/Counter1. The PB3 pin has to be configured as an output (DDB3 set (one)) to serve this function. The OC1BV pin is also the output pin for the PWM mode timer function (c.f. OC1BV bit of TCCR1D register).

- **Port B, Bit 2 – PCINT10/OC1AV/USCK/SCL**

10. 8-bit Timer/Counter0 and Asynchronous Operation

Timer/Counter0 is a general purpose, single channel, 8-bit Timer/Counter module. The main features are:

10.1 Features

- **Single Channel Counter**
- **Clear Timer on Compare Match (Auto Reload)**
- **Glitch-free, Phase Correct Pulse Width Modulator (PWM)**
- **Frequency Generator**
- **10-bit Clock Prescaler**
- **Overflow and Compare Match Interrupt Sources (TOV0 and OCF0A)**
- **Allows Clocking from External Crystal (i.e. 32 kHz Watch Crystal) Independent of the I/O Clock**

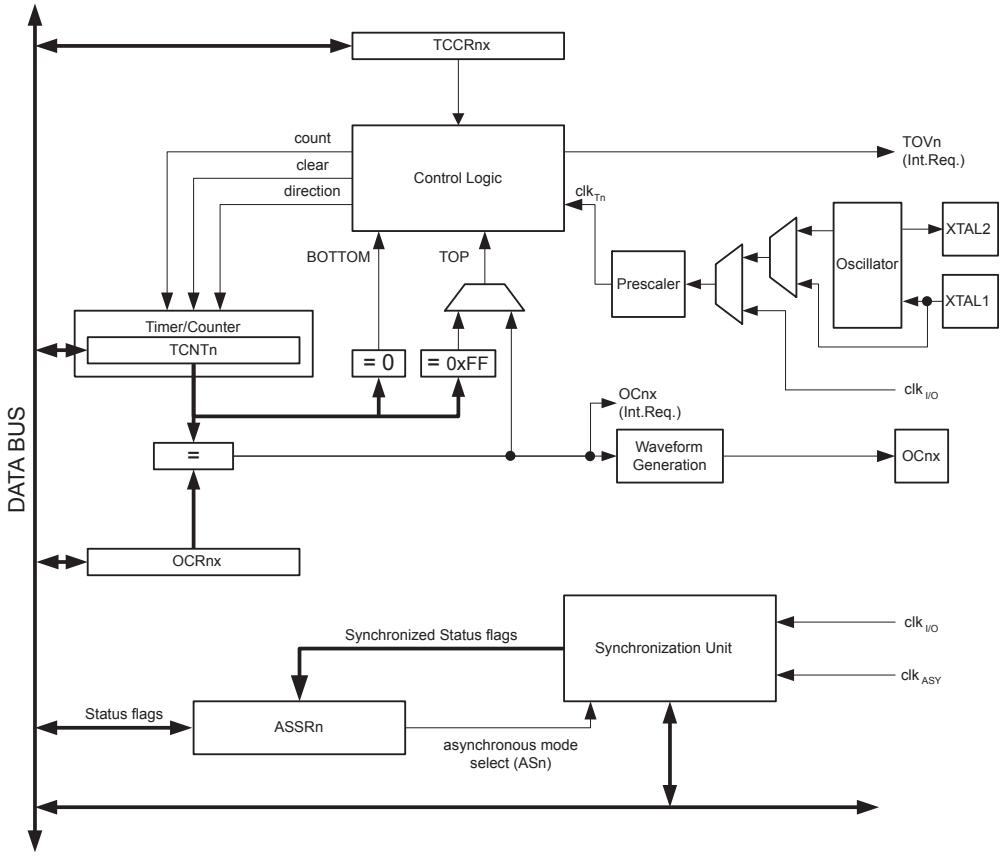
10.2 Overview

Many register and bit references in this section are written in general form.

- A lower case “n” replaces the Timer/Counter number, in this case 0. However, when using the register or bit defines in a program, the precise form must be used, i.e., TCNT0 for accessing Timer/Counter0 counter value and so on.
- A lower case “x” replaces the Output Compare unit channel, in this case A. However, when using the register or bit defines in a program, the precise form must be used, i.e., OCR0A for accessing Timer/Counter0 output compare channel A value and so on.

A simplified block diagram of the 8-bit Timer/Counter is shown in Figure 10-1. For the actual placement of I/O pins, refer to “Pin Configuration” on page 4. CPU accessible I/O Registers, including I/O bits and I/O pins, are shown in bold. The device-specific I/O Register and bit locations are listed in the “Register Description” on page 99.

Figure 10-1. 8-bit Timer/Counter0 Block Diagram



The Timer/Counter (TCNT0) and Output Compare Register (OCR0A) are 8-bit registers. Interrupt request (shorten as Int.Req.) signals are all visible in the Timer Interrupt Flag Register (TIFR0). All interrupts are individually masked with the Timer Interrupt Mask Register (TIMSK0). TIFR0 and TIMSK0 are not shown in the figure.

The Timer/Counter can be clocked internally, via the prescaler, or asynchronously clocked from the XTAL1/2 pins, as detailed later in this section. The asynchronous operation is controlled by the Asynchronous Status Register (ASSR). The Clock Select logic block controls which clock source the Timer/Counter uses to increment (or decrement) its value. The Timer/Counter is inactive when no clock source is selected. The output from the Clock Select logic is referred to as the timer clock (clk_T0).

The double buffered Output Compare Register (OCR0A) is compared with the Timer/Counter value at all times. The result of the compare can be used by the Waveform Generator to generate a PWM or variable frequency output on the Output Compare pin (OC0A). See ”Output Compare Unit” on page 89. for details. The compare match event will also set the compare flag (OCF0A) which can be used to generate an Output Compare interrupt request.

Depending on the mode of operation used, the counter is cleared, incremented, or decremented at each timer clock (clk_T0). clk_T0 can be generated from an external or internal clock source, selected by the Clock Select bits (CS0[2:0]). When no clock source is selected (CS0[2:0] = 0) the timer is stopped. However, the TCNT0 value can be accessed by the CPU, regardless of whether clk_T0 is present or not. A CPU write overrides (has priority over) all counter clear or count operations.

The counting sequence is determined by the setting of the WGM01 and WGM00 bits located in the Timer/Counter Control Register (TCCR0A). There are close connections between how the counter behaves (counts) and how waveforms are generated on the Output Compare output OC0A. For more details about advanced counting sequences and waveform generation, see “Modes of Operation” on page 91.

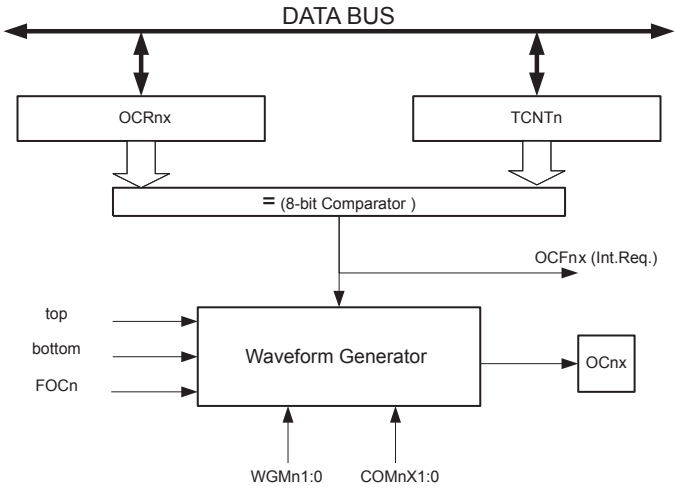
The Timer/Counter Overflow Flag (TOV0) is set according to the mode of operation selected by the WGM0[1:0] bits. TOV0 can be used for generating a CPU interrupt.

10.5 Output Compare Unit

The 8-bit comparator continuously compares TCNT0 with the Output Compare Register (OCR0A). Whenever TCNT0 equals OCR0A, the comparator signals a match. A match will set the Output Compare Flag (OCF0A) at the next timer clock cycle. If enabled (OCIE0A = 1), the Output Compare Flag generates an Output Compare interrupt. The OCF0A flag is automatically cleared when the interrupt is executed. Alternatively, the OCF0A flag can be cleared by software by writing a logical one to its I/O bit location. The Waveform Generator uses the match signal to generate an output according to operating mode set by the WGM0[1:0] bits and Compare Output mode (COM0A[1:0]) bits. The max and bottom signals are used by the Waveform Generator for handling the special cases of the extreme values in some modes of operation (“Modes of Operation” on page 91).

Figure 10-3 shows a block diagram of the Output Compare unit.

Figure 10-3. Output Compare Unit, Block Diagram



- **Bits 1:0 – WGM0[1:0]: Waveform Generation Mode**

These bits control the counting sequence of the counter, the source for the maximum (TOP) counter value, and what type of waveform generation to be used, see Table 10-4. Modes of operation supported by the Timer/Counter unit are: Normal mode (Counter), Clear Timer on Compare match (CTC) mode, and two types of Pulse Width Modulation (PWM) modes (See "Modes of Operation" on page 91.).

Table 10-4. Waveform Generation Mode Bit Description

Mode	WGM01 (CTC0)	WGM00 (PWM0)	Timer/Counter Mode of Operation	TOP	Update of OCR0A at	TOV0 Flag Set on(1)(2)
0	0	0	Normal	0xFF	Immediate	MAX
1	0	1	PWM, Phase Correct	0xFF	TOP	BOTTOM
2	1	0	CTC	OCR0A	Immediate	MAX
3	1	1	Fast PWM	0xFF	TOP	MAX

Notes: 1. MAX = 0xFF,
2. BOTTOM = 0x00.

10.11.2 TCCR0B – Timer/Counter0 Control Register B

Bit	7	6	5	4	3	2	1	0	
0x26 (0x46)	FOC0A	–	–	–	–	CS02	CS01	CS00	TCCR0B
Read/Write	W	R	R	R	R	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 COM0A[1:0] bits setting. Note that the FOC0A bit is implemented as a strobe. Therefore it is the value present in the COM0A[1:0] 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.

- **Bits 6:3 – Res: Reserved Bits**

These bits are reserved in the ATtiny87/167 and will always read as zero.

- **Bits 2:0 – CS0[2:0]: Clock Select**

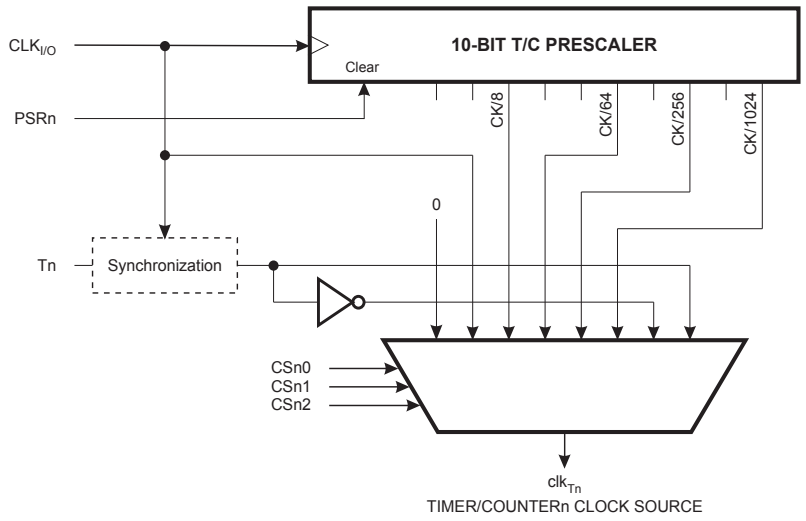
The synchronization and edge detector logic introduces a delay of 2.5 to 3.5 system clock cycles from an edge has been applied to the T1 pin to the counter is updated.

Enabling and disabling of the clock input must be done when T1 has been stable for at least one system clock cycle, otherwise it is a risk that a false Timer/Counter clock pulse is generated.

Each half period of the external clock applied must be longer than one system clock cycle to ensure correct sampling. The external clock must be guaranteed to have less than half the system clock frequency ($f_{ExtClk} < f_{clk\_I/O}/2$) given a 50/50 % duty cycle. Since the edge detector uses sampling, the maximum frequency of an external clock it can detect is half the sampling frequency (Nyquist sampling theorem). However, due to variation of the system clock frequency and duty cycle caused by Oscillator source (crystal, resonator, and capacitors) tolerances, it is recommended that maximum frequency of an external clock source is less than $f_{clk\_I/O}/2.5$.

An external clock source can not be prescaled.

Figure 11-2. Prescaler for Timer/Counter1 (1)



Note: 1. The synchronization logic on the input pin (T1) is shown in Figure 11-1.

The following code examples show how to do an atomic read of the TCNT1 Register contents. Reading any of the OCR1A/B or ICR1 Registers can be done by using the same principle.

Assembly Code Example[1]

```
TIM16_ReadTCNT1:
  ; Save global interrupt flag
  in     r18,SREG
  ; Disable interrupts
  cli
  ; Read TCNT1 into r17:r16
  lds    r16,TCNT1L
  lds    r17,TCNT1H
  ; Restore global interrupt flag
  out    SREG,r18
  ret
```

C Code Example[1]

```
unsigned int TIM16_ReadTCNT1(void)
{
  unsigned char sreg;
  unsigned int i;
  /* Save global interrupt flag */
  sreg = SREG;
  /* Disable interrupts */
  _CLI();
  /* Read TCNT1 into i */
  i = TCNT1;
  /* Restore global interrupt flag */
  SREG = sreg;
  return i;
}
```

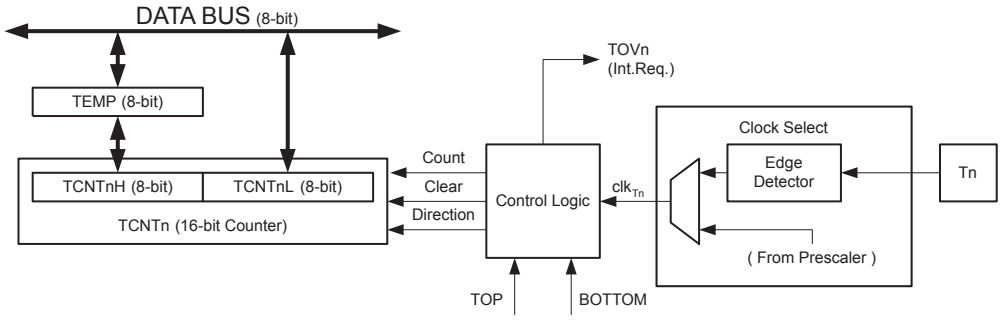
Note: 1. The example code assumes that the part specific header file is included.

The assembly code example returns the TCNT1 value in the r17:r16 register pair.

12.5 Counter Unit

The main part of the 16-bit Timer/Counter is the programmable 16-bit bi-directional counter unit. Figure 12-2 shows a block diagram of the counter and its surroundings.

Figure 12-2. Counter Unit Block Diagram



Signal description (internal signals):

Count	Increment or decrement TCNT1 by 1.
Direction	Select between increment and decrement.
Clear	Clear TCNT1 (set all bits to zero).
clk_T1	Timer/Counter clock.
TOP	Signalize that TCNT1 has reached maximum value.
BOTTOM	Signalize that TCNT1 has reached minimum value (zero).

The 16-bit counter is mapped into two 8-bit I/O memory locations: Counter High (TCNT1H) containing the upper eight bits of the counter, and Counter Low (TCNT1L) containing the lower eight bits. The TCNT1H Register can only be indirectly accessed by the CPU. When the CPU does an access to the TCNT1H I/O location, the CPU accesses the high byte temporary register (TEMP). The temporary register is updated with the TCNT1H value when the TCNT1L is read, and TCNT1H is updated with the temporary register value when TCNT1L is written. This allows the CPU to read or write the entire 16-bit counter value within one clock cycle via the 8-bit data bus. It is important to notice that there are special cases of writing to the TCNT1 Register when the counter is counting that will give unpredictable results. The special cases are described in the sections where they are of importance.

Depending on the mode of operation used, the counter is cleared, incremented, or decremented at each timer clock (clk_T1). The clk_T1 can be generated from an external or internal clock source, selected by the Clock Select bits (CS1[2:0]). When no clock source is selected (CS1[2:0] = 0) the timer is stopped. However, the TCNT1 value can be accessed by the CPU, independent of whether clk_T1 is present or not. A CPU write overrides (has priority over) all counter clear or count operations.

The counting sequence is determined by the setting of the Waveform Generation mode bits (WGM1[3:0]) located in the Timer/Counter Control Registers A and B (TCCR1A and TCCR1B). There are close connections between how the counter behaves (counts) and how waveforms are generated on the Output Compare outputs OC1A/B. For more details about advanced counting sequences and waveform generation, see “Modes of Operation” on page 121.

15.5.4 Configuration

Depending on the mode (LIN or UART), LCONF[1:0] bits of the LINCR register set the controller in the following configuration (Table 15-3):

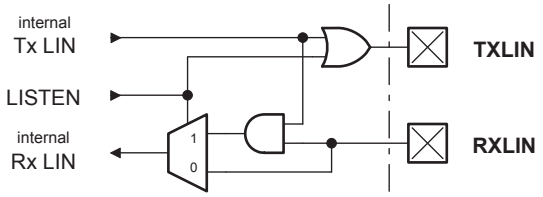
Table 15-3. Configuration Table versus Mode

Mode	LCONF[1:0]	Configuration
LIN	00_b	LIN standard configuration (default)
	01_b	No CRC field detection or transmission
	10_b	Frame_Time_Out disable
	11_b	Listening mode
UART	00_b	8-bit data, no parity & 1 stop-bit
	01_b	8-bit data, even parity & 1 stop-bit
	10_b	8-bit data, odd parity & 1 stop-bit
	11_b	Listening mode, 8-bit data, no parity & 1 stop-bit

The LIN configuration is independent of the programmed LIN protocol.

The listening mode connects the internal Tx LIN and the internal Rx LIN together. In this mode, the TXLIN output pin is disabled and the RXLIN input pin is always enabled. The same scheme is available in UART mode.

Figure 15-6. Listening Mode



15.5.5 Busy Signal

LBUSY bit flag in LINSIR register is the image of the BUSY signal. It is set and cleared by hardware. It signals that the controller is busy with LIN or UART communication.

15.5.5.1 Busy Signal in LIN Mode

Figure 15-7. Busy Signal in LIN Mode

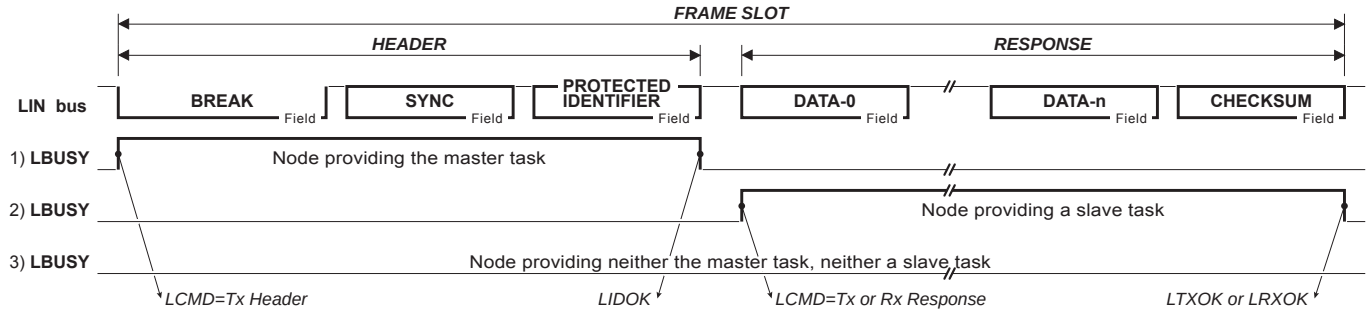


Figure 17-4. ADC Timing Diagram, First Conversion (Single Conversion Mode)

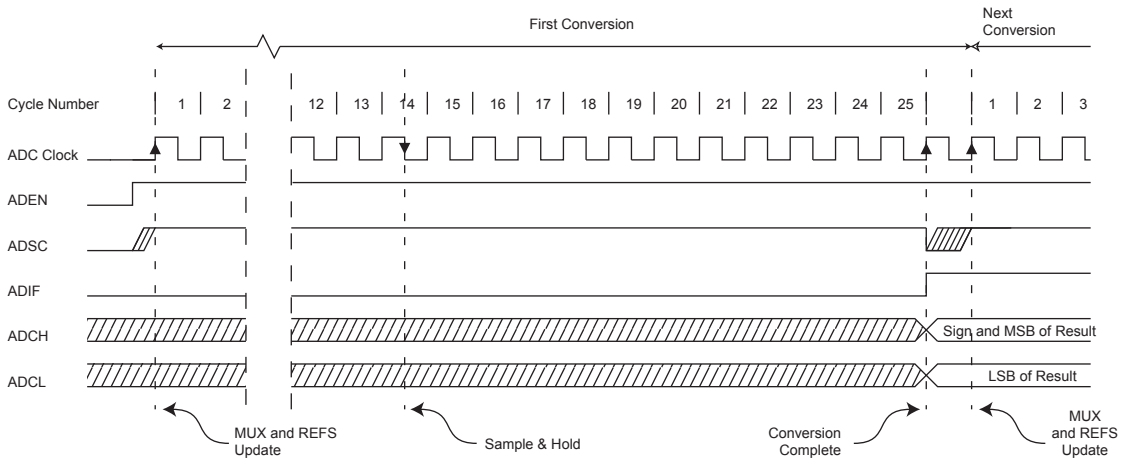


Figure 17-5. ADC Timing Diagram, Single Conversion

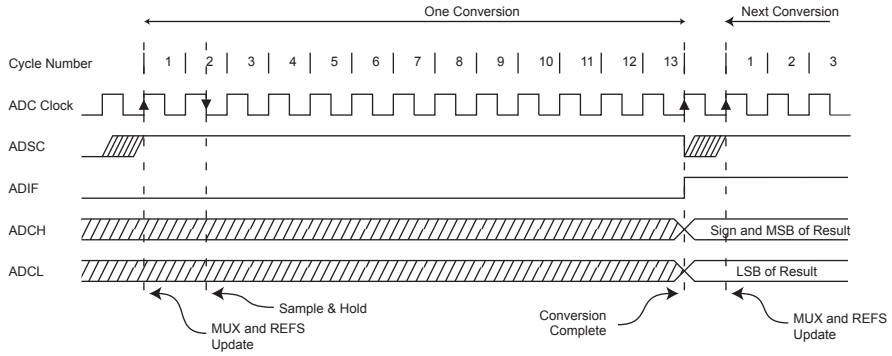


Figure 17-6. ADC Timing Diagram, Auto Triggered Conversion

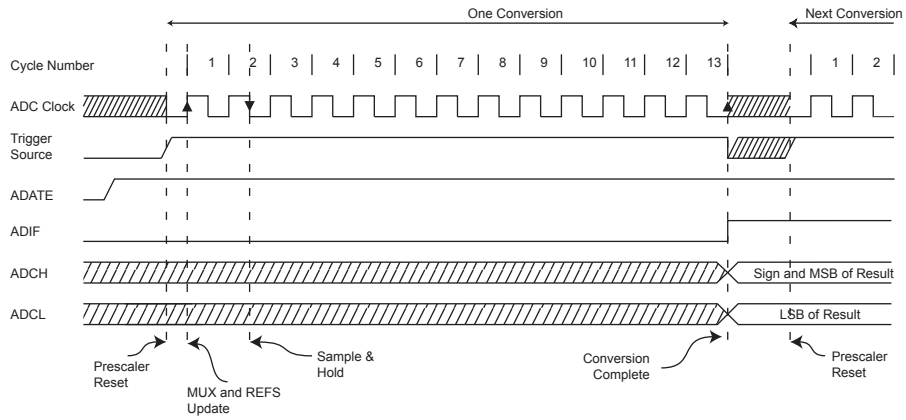


Figure 17-7. ADC Timing Diagram, Free Running Conversion

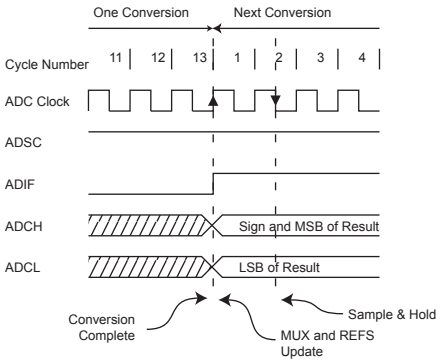


Table 17-1. ADC Conversion Time

Condition	Sample & Hold (Cycles from Start of Conversion)	Conversion Time (Cycles)
First conversion	13.5 cycles	25 cycles
Normal conversions	1.5 cycles	13 cycles
Auto Triggered conversions	2 cycles	13.5 cycles

17.6 Changing Channel or Reference Selection

The MUX[4:0] and REFS[1:0] bits in the ADMUX register are single buffered through a temporary register to which the CPU has random access. This ensures that the channels and reference selection only takes place at a safe point during the conversion. The channel and reference selection is continuously updated until a conversion is started. Once the conversion starts, the channel and reference selection is locked to ensure a sufficient sampling time for the ADC. Continuous updating resumes in the last ADC clock cycle before the conversion completes (ADIF in ADCSRA register is set). Note that the conversion starts on the following rising ADC clock edge after ADSC is written. The user is thus advised not to write new channel or reference selection values to ADMUX until one ADC clock cycle after ADSC is written.

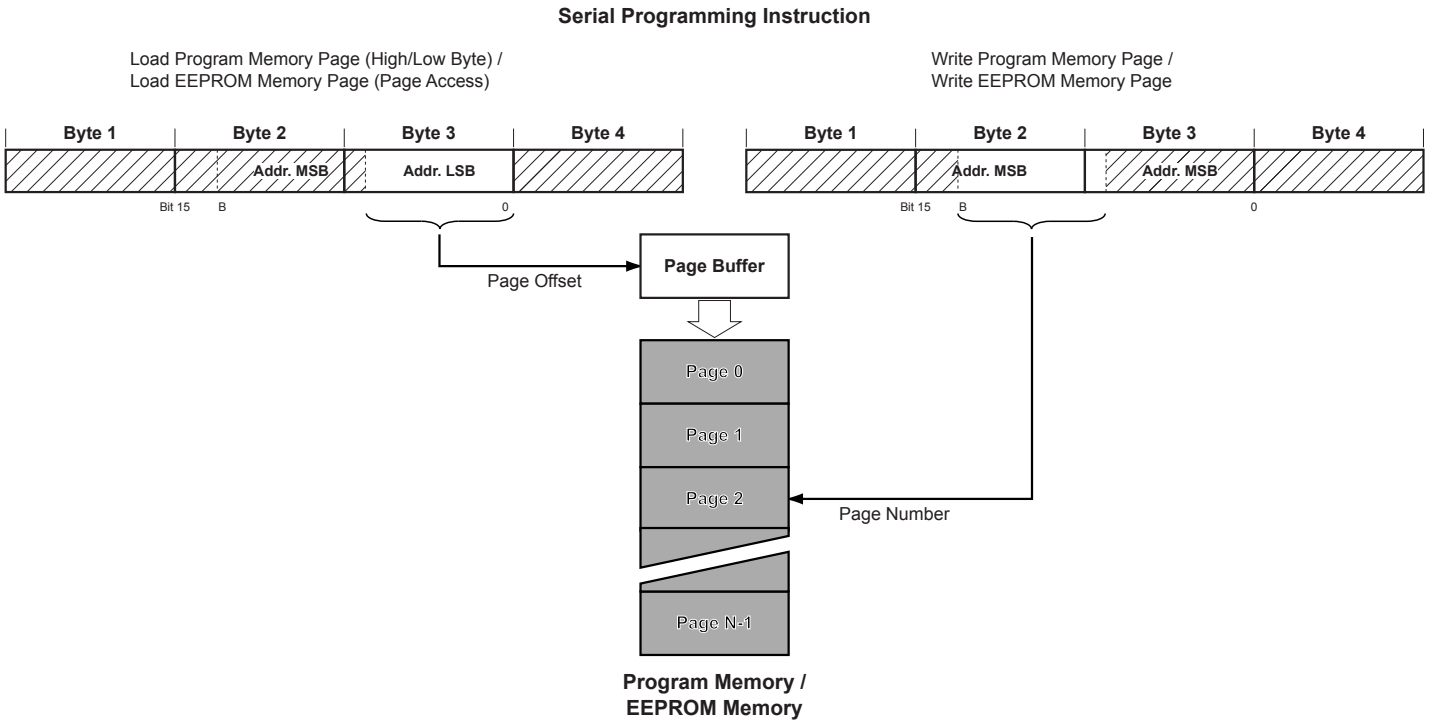
If Auto Triggering is used, the exact time of the triggering event can be indeterministic. Special care must be taken when updating the ADMUX register, in order to control which conversion will be affected by the new settings.

If both ADATE and ADEN is written to one, an interrupt event can occur at any time. If the ADMUX Register is changed in this period, the user cannot tell if the next conversion is based on the old or the new settings. ADMUX can be safely updated in the following ways:

a. When ADATE or ADEN is cleared.
b. During conversion, minimum one ADC clock cycle after the trigger event.
c. After a conversion, before the Interrupt Flag used as trigger source is cleared.

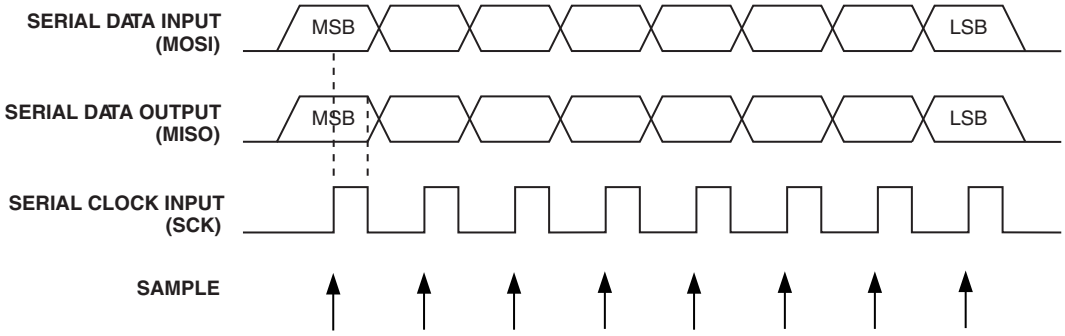
When updating ADMUX in one of these conditions, the new settings will affect the next ADC conversion.

Figure 21-8. Serial programming Instruction Example



21.9 Serial Programming Characteristics

Figure 21-9. Serial Programming Waveforms



For characteristics of the SPI module, See ”SPI Timing Characteristics” on page 252.

T_A = -40°C to +85°C, V_{CC} = 1.8V to 5.5V (unless otherwise noted) (Continued)

Symbol	Parameter	Condition	Min.	Typ.[1]	Max.	Units
R_{RST}	Reset Pull-up Resistor		30		60	kΩ
R_{pu}	I/O Pin Pull-up Resistor		20		50	kΩ
I_{CC}	Power Supply Current [6] Active Mode (external clock)	8 MHz, V_{CC} = 5V		5.5	7.0	mA
		4 MHz, V_{CC} = 3V		1.8	2.5	mA
		1 MHz, V_{CC} = 2V		0.3	0.6	mA
	Power Supply Current [6] Idle Mode (external clock)	8 MHz, V_{CC} = 5V		1.8	2.5	mA
		4 MHz, V_{CC} = 3V		0.5	0.8	mA
		1 MHz, V_{CC} = 2V		0.05	0.2	mA
	Power Supply Current [7] Power-down Mode	WDT enabled, V_{CC} = 3V		5	10	µA
		WDT disabled, V_{CC} = 3V		0.15	4	µA
V_{ACIO}	Analog Comparator Input Offset Voltage	V_{CC} = 5V V_{in} = V_{CC}/2	-10	10	40	mV
I_{ACLK}	Analog Comparator Input Leakage Current	V_{CC} = 5V V_{in} = V_{CC}/2	-50		50	nA
t_{ACID}	Analog Comparator Propagation Delay Common Mode V_{CC}/2	V_{CC} = 2.7V		170		ns
		V_{CC} = 5.0V		180		ns

Notes:
1. “Typ.", typical values at 25°C. Maximum values are characterized values and not test limits in production.
2. "Max.” means the highest value where the pin is guaranteed to be read as low.
3. “Min.” means the lowest value where the pin is guaranteed to be read as high.
4. Although each I/O port can sink more than the test conditions (10 mA at V_{CC}= 5V, 5 mA at V_{CC} = 3V) under steady state conditions (non-transient), the following must be observed:
   - The sum of all IOL, for all ports, should not exceed 120 mA.
   If IOL exceeds the test condition, VOL may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test condition.
5. Although each I/O port can source more than the test conditions (10 mA at V_{CC} = 5V, 5 mA at V_{CC} = 3V) under steady state conditions (non-transient), the following must be observed:
   - The sum of all IOH, for all ports, should not exceed 120 mA.
   If IOH exceeds the test condition, VOH may exceed the related specification. Pins are not guaranteed to source current greater than the listed test condition.
6. Values are with external clock using methods described in “Minimizing Power Consumption” on page 45. Power Reduction is enabled (PRR = 0xFF) and there is no I/O drive.
7. BOD Disabled.