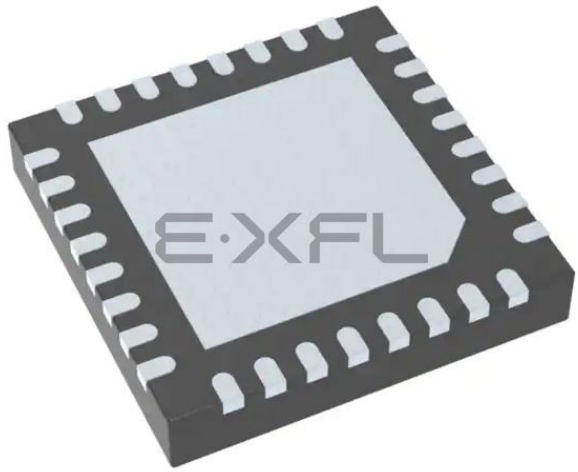


Welcome to E-XFL.COM

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"

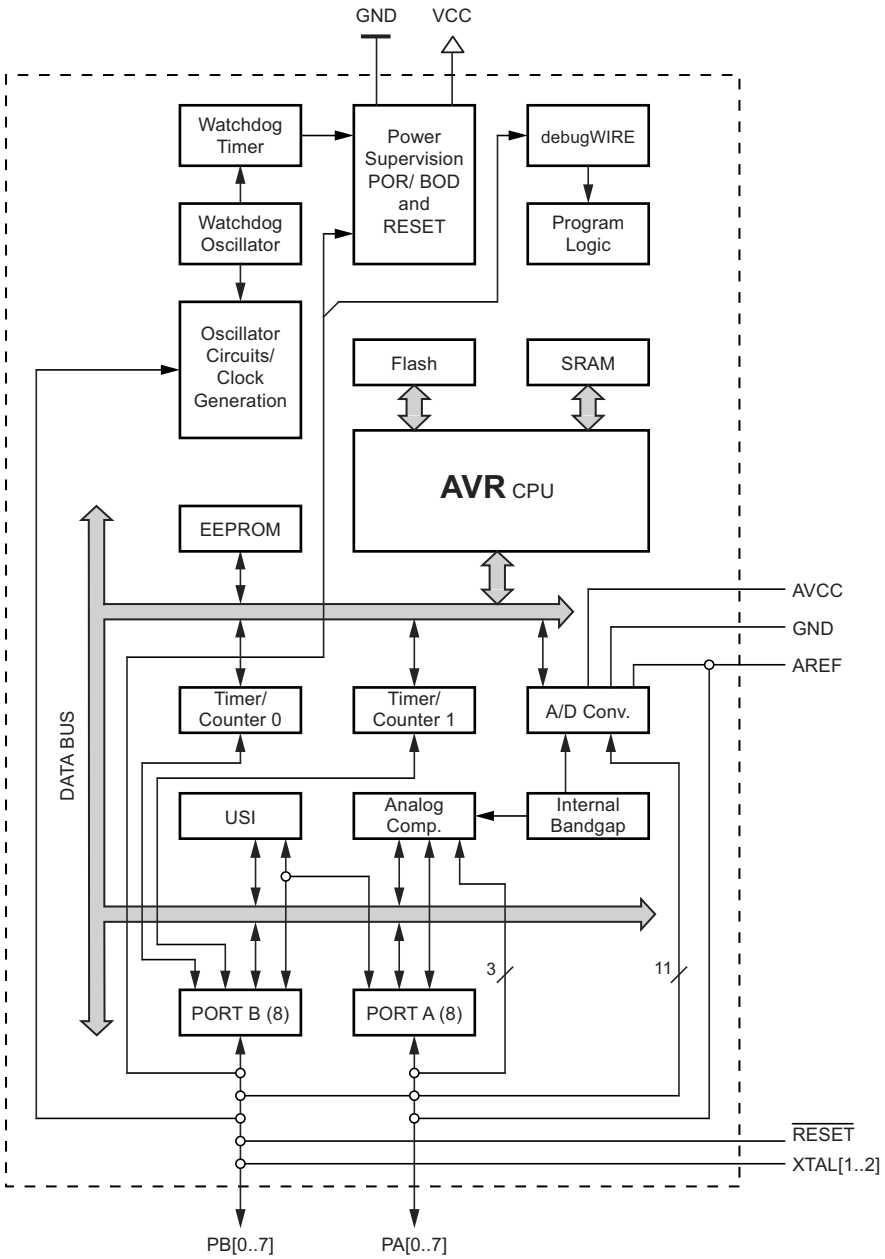
Details	
Product Status	Active
Core Processor	AVR
Core Size	8-Bit
Speed	16MHz
Connectivity	USI
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	16
Program Memory Size	4KB (2K x 16)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	256 x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 5.5V
Data Converters	A/D 11x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 150°C (TA)
Mounting Type	Surface Mount
Package / Case	32-VFQFN Exposed Pad
Supplier Device Package	32-QFN (5x5)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/attiny461-15md

2. Overview

The Atmel® ATtiny261/461/861 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 ATtiny261/461/861 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



5.3 Status Register

The status register contains information about the result of the most recently executed arithmetic instruction. This information can be used for altering program flow in order to perform conditional operations. Note that the status register is updated after all ALU operations, as specified in the instruction set reference. This will in many cases remove the need for using the dedicated compare instructions, resulting in faster and more compact code. The status register is not automatically stored when entering an interrupt routine and restored when returning from an interrupt. This must be handled by software.

5.3.1 SREG – AVR Status Register

The AVR® status register – SREG – is defined as:

Bit	7	6	5	4	3	2	1	0	
0x3F (0x5F)	**I**	**T**	**H**	**S**	**V**	**N**	**Z**	**C**	**SREG**
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 – I: Global Interrupt Enable**

The global interrupt enable bit must be set for the interrupts to be enabled. The individual interrupt enable control is then performed in separate control registers. If the global interrupt enable register is cleared, none of the interrupts are enabled independent of the individual interrupt enable settings. The I-bit is cleared by hardware after an interrupt has occurred, and is set by the RETI instruction to enable subsequent interrupts. The I-bit can also be set and cleared by the application with the SEI and CLI instructions, as described in the instruction set reference.

- **Bit 6 – T: Bit Copy Storage**

The bit copy instructions BLD (Bit LoaD) and BST (Bit STore) use the T-bit as source or destination for the operated bit. A bit from a register in the register file can be copied into T by the BST instruction, and a bit in T can be copied into a bit in a register in the register file by the BLD instruction.

- **Bit 5 – H: Half Carry Flag**

The half carry flag H indicates a half carry in some arithmetic operations. Half carry is useful in BCD arithmetic. See the “Instruction Set Description” for detailed information.

- **Bit 4 – S: Sign Bit, S = N ⊕ V**

The S-bit is always an exclusive or between the negative flag N and the two complement overflow flag V. See the “Instruction Set Description” for detailed information.

- **Bit 3 – V: Two’s Complement Overflow Flag**

The Two’s complement overflow flag V supports two’s complement arithmetics. See the “Instruction Set Description” for detailed information.

- **Bit 2 – N: Negative Flag**

The negative flag N indicates a negative result in an arithmetic or logic operation. See the “Instruction Set Description” for detailed information.

- **Bit 1 – Z: Zero Flag**

The zero flag Z indicates a zero result in an arithmetic or logic operation. See the “Instruction Set Description” for detailed information.

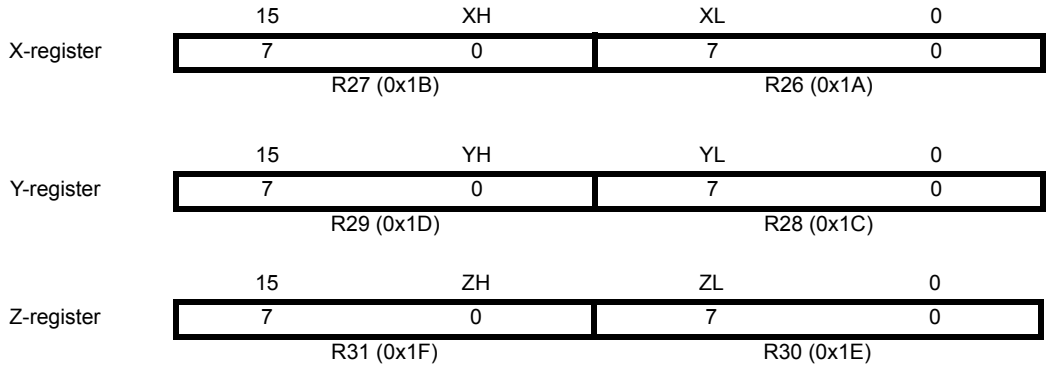
- **Bit 0 – C: Carry Flag**

The carry flag C indicates a carry in an arithmetic or logic operation. See the “Instruction Set Description” for detailed information.

5.4.1 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 5-3.

Figure 5-3. The X-, Y-, and Z-registers



In the different addressing modes these address registers have functions as fixed displacement, automatic increment, and automatic decrement (see the instruction set reference for details).

5.5 Stack Pointer

The stack is mainly used for storing temporary data, for storing local variables and for storing return addresses after interrupts and subroutine calls. The stack pointer register always points to the top of the stack. Note that the stack is implemented as growing from higher memory locations to lower memory locations. This implies that a stack PUSH command decreases the stack pointer.

The stack pointer points to the data SRAM stack area where the subroutine and interrupt stacks are located. This stack space in the data SRAM must be defined by the program before any subroutine calls are executed or interrupts are enabled. The stack pointer must be set to point above 0x60. The stack pointer is decremented by one when data is pushed onto the stack with the PUSH instruction, and it is decremented by two when the return address is pushed onto the stack with subroutine call or interrupt. The stack pointer is incremented by one when data is popped from the stack with the POP instruction, and it is incremented by two when data is popped from the stack with return from subroutine RET or return from interrupt RETI.

The AVR® stack pointer is implemented as two 8-bit registers in the I/O space. The number of bits actually used is implementation dependent. Note that the data space in some implementations of the AVR architecture is so small that only SPL is needed. In this case, the SPH register will not be present.

5.5.1 SPH and SPL – Stack Pointer Register

Bit	15	14	13	12	11	10	9	8	
0x3E (0x5E)	SP15	SP14	SP13	SP12	SP11	SP10	SP9	SP8	SPH
0x3D (0x5D)	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0	SPL
	7	6	5	4	3	2	1	0	
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	RAMEND	RAMEND	RAMEND	RAMEND	RAMEND	RAMEND	RAMEND	RAMEND	
	RAMEND	RAMEND	RAMEND	RAMEND	RAMEND	RAMEND	RAMEND	RAMEND	

5.6 Instruction Execution Timing

This section describes the general access timing concepts for instruction execution. The AVR® CPU is driven by the CPU clock clk_{CPU}, directly generated from the selected clock source for the chip. No internal clock division is used.

Figure 5-4 shows the parallel instruction fetches and instruction executions enabled by the Harvard architecture and the fast access register file concept. This is the basic pipelining concept to obtain up to 1MIPS per MHz with the corresponding unique results for functions per cost, functions per clocks, and functions per power-unit.

Figure 5-4. The Parallel Instruction Fetches and Instruction Executions

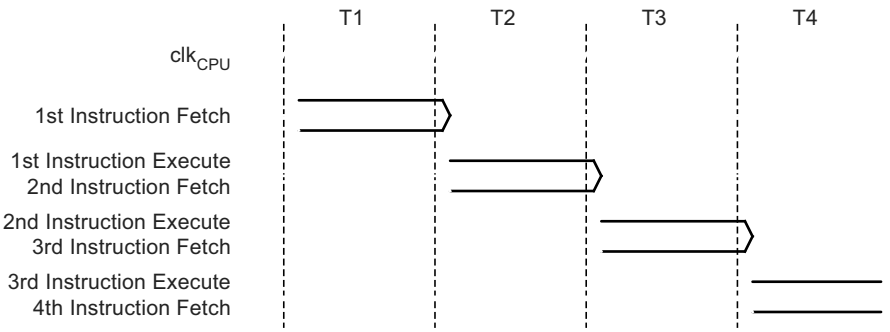


Figure 5-5 shows the internal timing concept for the register file. In a single clock cycle an ALU operation using two register operands is executed, and the result is stored back to the destination register.

Figure 5-5. Single Cycle ALU Operation

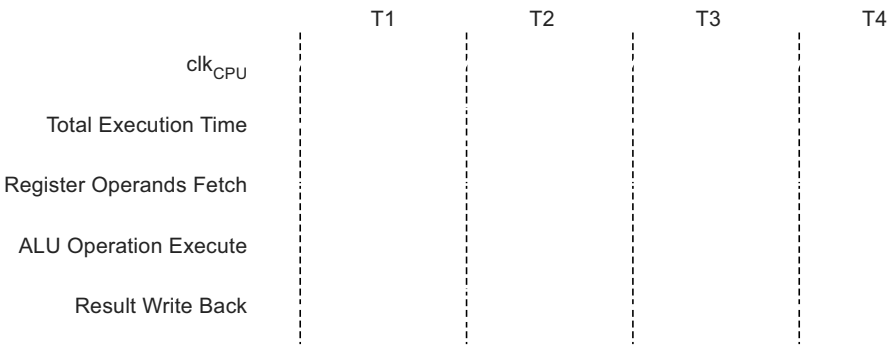
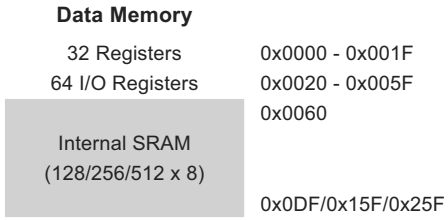


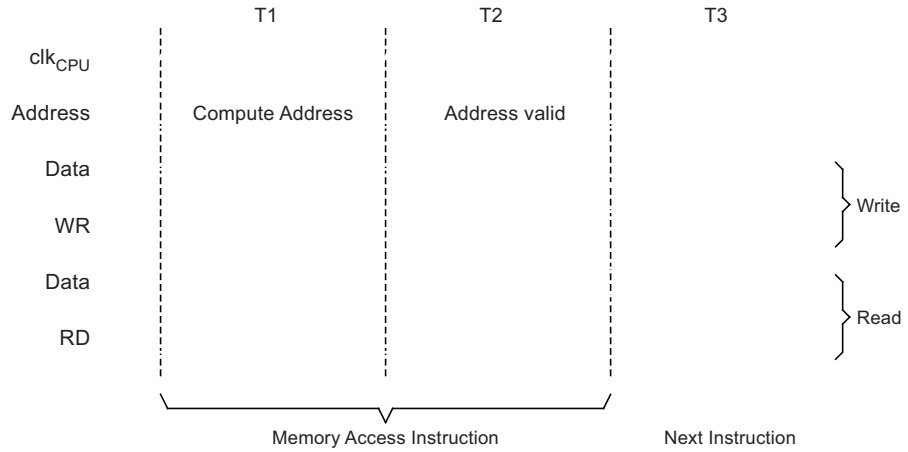
Figure 6-2. Data Memory Map



6.2.1 Data Memory Access Times

This section describes the general access timing concepts for internal memory access. The internal data SRAM access is performed in two clk_{CPU} cycles as described in Figure 6-3.

Figure 6-3. On-chip Data SRAM Access Cycles



7.2 Clock Sources

The device has the following clock source options, selectable by flash fuse bits as shown below. The clock from the selected source is input to the AVR® clock generator, and routed to the appropriate modules.

Table 7-1. Device Clocking Options Select(1) versus PB4 and PB5 Functionality

Device Clocking Option	CKSEL3..0	PB4	PB5
External clock	0000	XTAL1	I/O
PLL clock	0001	I/O	I/O
Calibrated internal RC oscillator 8.0MHz	0010	I/O	I/O
Watchdog oscillator 128kHz	0011	I/O	I/O
External low-frequency crystal	01xx	XTAL1	XTAL2
External crystal/ceramic resonator (0.4 - 0.9MHz)	1000	XTAL1	XTAL2
External crystal/ceramic resonator (0.4 - 0.9MHz)	1001	XTAL1	XTAL2
External crystal/ceramic resonator (0.9 - 3.0MHz)	1010	XTAL1	XTAL2
External crystal/ceramic resonator (0.9 - 3.0MHz)	1011	XTAL1	XTAL2
External crystal/ceramic resonator (3.0 - 8.0MHz)	1100	XTAL1	XTAL2
External crystal/ceramic resonator (3.0 - 8.0MHz)	1101	XTAL1	XTAL2
External crystal/ceramic resonator (8.0 - 16.0MHz)	1110	XTAL1	XTAL2
External crystal/ceramic resonator (8.0 - 16.0MHz)	1111	XTAL1	XTAL2

Note: 1. For all fuses “1” means unprogrammed while “0” means programmed.

The various choices for each clocking option is given in the following sections. When the CPU wakes up from power-down or power-save, the selected clock source is used to time the start-up, ensuring stable oscillator operation before instruction execution starts. When the CPU starts from reset, there is an additional delay allowing the power to reach a stable level before commencing normal operation. The watchdog oscillator is used for timing this real-time part of the start-up time. The number of WDT oscillator cycles used for each time-out is shown in Table 7-2.

Table 7-2. Number of Watchdog Oscillator Cycles

Typ Time-out	Number of Cycles
4ms	512
64ms	8K (8,192)

7.3 Default Clock Source

The device is shipped with CKSEL = “0010”, SUT = “10”, and CKDIV8 programmed. The default clock source setting is therefore the internal RC oscillator running at 8MHz with longest start-up time and an initial system clock prescaling of 8. This default setting ensures that all users can make their desired clock source setting using an in-system or high-voltage programmer.

8. Power Management and Sleep Modes

The high performance and industry leading code efficiency makes the AVR® microcontrollers an ideal choice for low power applications.

Sleep modes enable the application to shut down unused modules in the MCU, thereby saving power. The AVR provides various sleep modes allowing the user to tailor the power consumption to the application's requirements.

8.1 Sleep Modes

Figure 7-1 on page 24 presents the different clock systems in the Atmel® ATtiny261/461/861, and their distribution. The figure is helpful in selecting an appropriate sleep mode. Table 8-1 shows the different sleep modes and their wake up sources.

Table 8-1. Active Clock Domains and Wake-up Sources in the Different Sleep Modes

	Active Clock Domains					Oscillators	Wake-up Sources					
Sleep Mode	clk_{CPU}	clk_{FLASH}	clk_{IO}	clk_{ADC}	clk_{PCK}	Main Clock Source Enabled	INT0, INT1 and Pin Change	SPM/EEPROM Ready	ADC	WDT Interrupt	USI Interrupt	Other I/O
Idle			X	X	X	X	X	X	X	X	X	X
ADC noise reduction				X		X	X(1)	X	X	X	X	
Power-down							X(1)			X	X	
Standby							X(1)			X	X	

Note: 1. For INT0 and INT1, only level interrupt.

To enter any of the three sleep modes, the SE bit in MCUCR must be written to logic one and a SLEEP instruction must be executed. The SM1..0 bits in the MCUCR register select which sleep mode (idle, ADC noise reduction, power-down, or standby) will be activated by the SLEEP instruction. See Table 8-2 on page 37 for a summary.

If an enabled interrupt occurs while the MCU is in a sleep mode, the MCU wakes up. The MCU is then halted for four cycles in addition to the start-up time, executes the interrupt routine, and resumes execution from the instruction following SLEEP. The contents of the register file and SRAM are unaltered when the device wakes up from sleep. If a reset occurs during sleep mode, the MCU wakes up and executes from the reset vector.

8.2 Idle Mode

When the SM1..0 bits are written to 00, the SLEEP instruction makes the MCU enter idle mode, stopping the CPU but allowing analog comparator, ADC, 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 timer overflow. 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.

8.3 ADC Noise Reduction Mode

When the SM1..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, and the watchdog to continue operating (if enabled). This sleep mode 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 form the ADC conversion complete interrupt, only an external reset, a watchdog reset, a brown-out reset, an SPM/EEPROM ready interrupt, an external level interrupt on INT0 or a pin change interrupt can wake up the MCU from ADC noise reduction mode.

9. System Control and Reset

9.1 Resetting the AVR

During reset, all I/O registers are set to their initial values, and the program starts execution from the reset vector. The instruction placed at the reset vector must be a RJMP – relative jump – instruction to the reset handling routine. If the program never enables an interrupt source, the interrupt vectors are not used, and regular program code can be placed at these locations. The circuit diagram in Figure 9-1 on page 39 shows the reset logic. See Section 23.5 “System and Reset Characteristics” on page 174 defines the electrical parameters of the reset circuitry.

The I/O ports of the AVR® are immediately reset to their initial state when a reset source goes active. This does not require any clock source to be running.

After all reset sources have gone inactive, a delay counter is invoked, stretching the internal reset. This allows the power to reach a stable level before normal operation starts. The time-out period of the delay counter is defined by the user through the SUT and CKSEL fuses. The different selections for the delay period are presented
in Section 7.2 “Clock Sources” on page 26.

9.2 Reset Sources

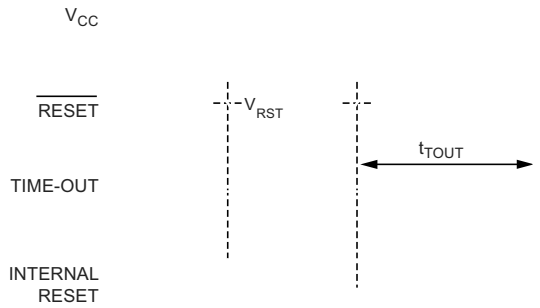
The Atmel® ATtiny261/461/861 has four sources of reset:

- Power-on reset. The MCU is reset when the supply voltage is below the power-on reset threshold (V_{POT}).
- External reset. The MCU is reset when a low level is present on the $\overline{\text{RESET}}$ pin for longer than the minimum pulse length.
- Watchdog reset. The MCU is reset when the watchdog timer period expires and the watchdog is enabled.
- Brown-out reset. The MCU is reset when the supply voltage V_{CC} is below the brown-out reset threshold (V_{BOT}) and the brown-out detector is enabled.

9.4 External Reset

An external reset is generated by a low level on the $\overline{\text{RESET}}$ pin if enabled. Reset pulses longer than the minimum pulse width (see Section 23.5 “System and Reset Characteristics” on page 174) will generate a reset, even if the clock is not running. Shorter pulses are not guaranteed to generate a reset. When the applied signal reaches the reset threshold voltage – V_{RST} – on its positive edge, the delay counter starts the MCU after the Time-out period – t_{TOUT} – has expired.

Figure 9-4. External Reset During Operation



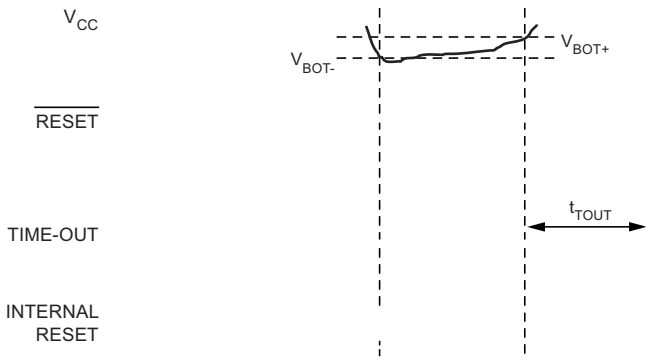
9.5 Brown-out Detection

Atmel® ATtiny261/461/861 has an on-chip brown-out detection (BOD) circuit for monitoring the V_{CC} level during operation by comparing it to a fixed trigger level. The trigger level for the BOD can be selected by the BODLEVEL fuses. The trigger level has a hysteresis to ensure spike free brown-out detection. The hysteresis on the detection level should be interpreted as $V_{BOT+} = V_{BOT} + V_{HYST}/2$ and $V_{BOT-} = V_{BOT} – V_{HYST}/2$.

When the BOD is enabled, and V_{CC} decreases to a value below the trigger level (V_{BOT-} in Figure 9-5), the brown-out reset is immediately activated. When V_{CC} increases above the trigger level (V_{BOT+} in Figure 9-5), the delay counter starts the MCU after the time-out period t_{TOUT} has expired.

The BOD circuit will only detect a drop in V_{CC} if the voltage stays below the trigger level for longer than t_{BOD} given in Section 23.5 “System and Reset Characteristics” on page 174.

Figure 9-5. Brown-out Reset During Operation



11.1.2 GIMSK – General Interrupt Mask Register

Bit	7	6	5	4	3	2	1	0	
0x3B (0x5B)	INT1	INT0	PCIE1	PCIE0	–	–	–	–	GIMSK
Read/Write	R/W	R/W	R/W	R/w	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – INT1: External Interrupt Request 1 Enable**

When the INT1 bit is set (one) and the I-bit in the status register (SREG) is set (one), the external pin interrupt is enabled. The interrupt sense control0 bits 1/0 (ISC01 and ISC00) in the MCU control register (MCUCR) define whether the external interrupt is activated on rising and/or falling edge of the INT1 pin or level sensed. Activity on the pin will cause an interrupt request even if INT1 is configured as an output. The corresponding interrupt of external interrupt request 1 is executed from the INT1 interrupt vector.

- **Bit 6 – INT0: External Interrupt Request 0 Enable**

When the INT0 bit is set (one) and the I-bit in the status register (SREG) is set (one), the external pin interrupt is enabled. The interrupt sense control0 bits 1/0 (ISC01 and ISC00) in the MCU control register (MCUCR) define whether the external interrupt is activated on rising and/or falling edge of the INT0 pin or level sensed. Activity on the pin will cause an interrupt request even if INT0 is configured as an output. The corresponding interrupt of external interrupt request 0 is executed from the INT0 interrupt vector.

- **Bit 5 – PCIE1: Pin Change Interrupt Enable**

When the PCIE1 bit is set (one) and the I-bit in the status register (SREG) is set (one), pin change interrupt is enabled. Any change on any enabled PCINT7..0 or PCINT15..12 pin will cause an interrupt. The corresponding interrupt of pin change interrupt request is executed from the PCI interrupt vector. PCINT7..0 and PCINT15..12 pins are enabled individually by the PCMSK0 and PCMSK1 register.

- **Bit 4 – PCIE0: Pin Change Interrupt Enable**

When the PCIE0 bit is set (one) and the I-bit in the status register (SREG) is set (one), pin change interrupt is enabled. Any change on any enabled PCINT11..8 pin will cause an interrupt. The corresponding interrupt of pin change interrupt request is executed from the PCI interrupt vector. PCINT11..8 pins are enabled individually by the PCMSK1 register.

- **Bits 3..0 – Res: Reserved Bits**

These bits are reserved bits in the Atmel® ATtiny261/461/861 and will always read as zero.

11.1.3 GIFR – General Interrupt Flag Register

Bit	7	6	5	4	3	2	1	0	
0x3A (0x5A)	INT1	INTF0	PCIF	–	–	–	–	–	GIFR
Read/Write	R/W	R/W	R/W	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7– INTF1: External Interrupt Flag 1**

When an edge or logic change on the INT1 pin triggers an interrupt request, INTF1 becomes set (one). If the I-bit in SREG and the INT1 bit in GIMSK are set (one), the MCU will jump to the corresponding interrupt vector. The flag is cleared when the interrupt routine is executed. Alternatively, the flag can be cleared by writing a logical one to it. This flag is always cleared when INT1 is configured as a level interrupt.

- **Bit 6 – INTF0: External Interrupt Flag 0**

When an edge or logic change on the INT0 pin triggers an interrupt request, INTF0 becomes set (one). If the I-bit in SREG and the INT0 bit in GIMSK are set (one), the MCU will jump to the corresponding interrupt vector. The flag is cleared when the interrupt routine is executed. Alternatively, the flag can be cleared by writing a logical one to it. This flag is always cleared when INT0 is configured as a level interrupt.

12.4.4 PINA – Port A Input Pins Address

Bit	7	6	5	4	3	2	1	0	
0x19 (0x39)	**PINA7**	**PINA6**	**PINA5**	**PINA4**	**PINA3**	**PINA2**	**PINA1**	**PINA0**	**PINA**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	

12.4.5 PORTB – Port B Data Register

Bit	7	6	5	4	3	2	1	0	
0x18 (0x38)	**PORTB7**	**PORTB6**	**PORTB5**	**PORTB4**	**PORTB3**	**PORTB2**	**PORTB1**	**PORTB0**	**PORTB**
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	

12.4.6 DDRB – Port B Data Direction Register

Bit	7	6	5	4	3	2	1	0	
0x17 (0x37)	**DDB7**	**DDB6**	**DDB5**	**DDB4**	**DDB3**	**DDB2**	**DDB1**	**DDB0**	**DDRB**
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	

12.4.7 PINB – Port B Input Pins Address

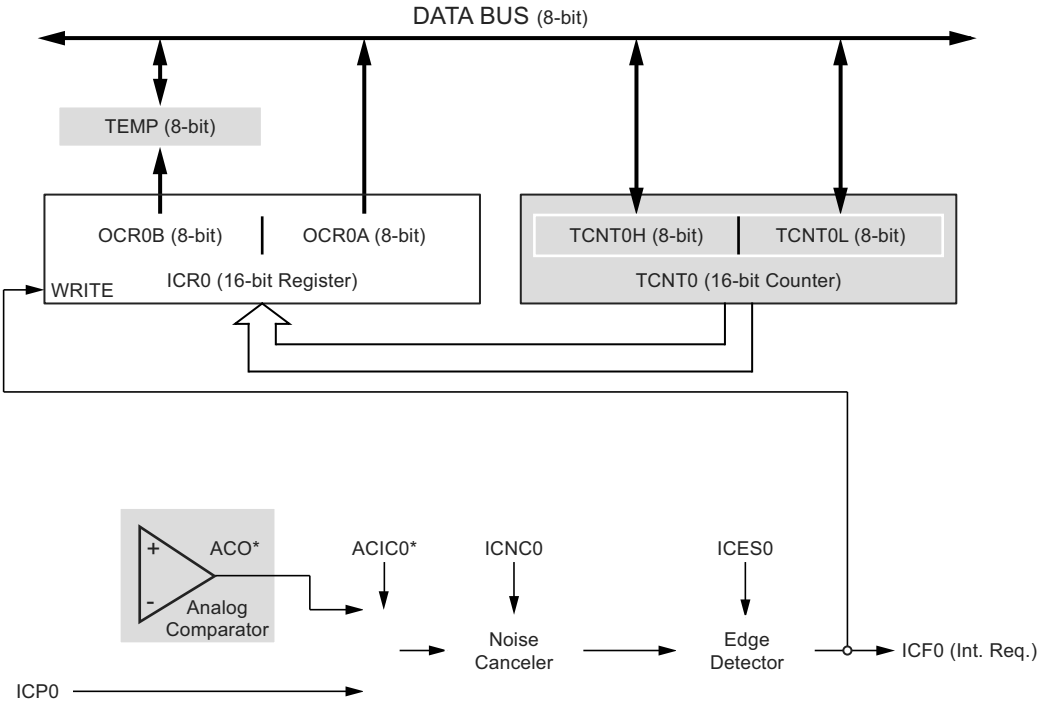
Bit	7	6	5	4	3	2	1	0	
0x16 (0x36)	**PINB7**	**PINB6**	**PINB5**	**PINB4**	**PINB3**	**PINB2**	**PINB1**	**PINB0**	**PINB**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	

14.6 Input Capture Unit

The Timer/Counter incorporates an input capture unit that can capture external events and give them a time-stamp indicating time of occurrence. The external signal indicating an event, or multiple events, can be applied via the ICP0 pin or alternatively, via the analog-comparator unit. The time-stamps can then be used to calculate frequency, duty-cycle, and other features of the signal applied. Alternatively the time-stamps can be used for creating a log of the events.

The input capture unit is illustrated by the block diagram shown in Figure 14-3. The elements of the block diagram that are not directly a part of the input capture unit are gray shaded.

Figure 14-3. Input Capture Unit Block Diagram



The output compare register OCR0A is a dual-purpose register that is also used as an 8-bit input capture register ICR0. In 16-bit input capture mode the output compare register OCR0B serves as the high byte of the input capture register ICR0. In 8-bit input capture mode the output compare register OCR0B is free to be used as a normal output compare register, but in 16-bit input capture mode the output compare unit cannot be used as there are no free output compare register(s). Even though the input capture register is called ICR0 in this section, it is referring to the output compare register(s).

When a change of the logic level (an event) occurs on the *Input Capture pin* (ICP0), alternatively on the *Analog Comparator Output* (ACO), and this change confirms to the setting of the edge detector, a capture will be triggered. When a capture is triggered, the value of the counter (TCNT0) is written to the *Input Capture Register* (ICR0). The *Input Capture Flag* (ICF0) is set at the same system clock as the TCNT0 value is copied into input capture register. If enabled (TICIE0=1), the input capture flag generates an input capture interrupt. The ICF0 flag is automatically cleared when the interrupt is executed. Alternatively the ICF0 flag can be cleared by software by writing a logical one to its I/O bit location.

14.6.1 Input Capture Trigger Source

The default trigger source for the input capture unit is the *Input Capture pin* (ICP0). Timer/Counter0 can alternatively use the analog comparator output as trigger source for the input capture unit. The analog comparator is selected as trigger source by setting the *Analog Comparator Input Capture Enable* (ACIC0) bit in the *Timer/Counter Control Register A* (TCCR0A). Be aware that changing trigger source can trigger a capture. The input capture flag must therefore be cleared after the change.

Both the *Input Capture pin* (ICP0) and the *Analog Comparator output* (ACO) inputs are sampled using the same technique as for the T0 pin (Figure 13-1 on page 68). The edge detector is also identical. However, when the noise canceler is enabled, additional logic is inserted before the edge detector, which increases the delay by four system clock cycles. An input capture can also be triggered by software by controlling the port of the ICP0 pin.

16.2.5 Definitions

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. A lower case “x” replaces the output compare unit, in this case compare unit A, B, C or D. However, when using the register or bit defines in a program, the precise form must be used, i.e., TCNT1 for accessing Timer/Counter1 counter value and so on. The definitions in Table 16-1 are used extensively throughout the document.

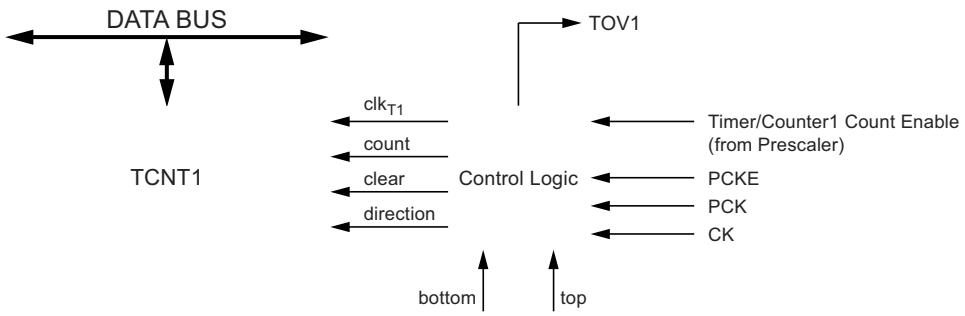
Table 16-1. Definitions

Parameter	Definition
BOTTOM	The counter reaches the BOTTOM when it becomes 0.
MAX	The counter reaches its MAXimum value when it becomes 0x3FF (decimal 1023).
TOP	The counter reaches the TOP value (stored in the OCR1C) when it becomes equal to the highest value in the count sequence. The TOP has a value 0x0FF as default after reset.

16.3 Counter Unit

The main part of the Timer/Counter1 is the programmable bi-directional counter unit. Figure 16-3 shows a block diagram of the counter and its surroundings.

Figure 16-3. Counter Unit Block Diagram



Signal description (internal signals):

count	TCNT1 increment or decrement enable.
direction	Select between increment and decrement.
clear	Clear TCNT1 (set all bits to zero).
clk_{Tn}	Timer/Counter clock, referred to as clk_{T1} in the following.
top	Signalize that TCNT1 has reached maximum value.
bottom	Signalize that TCNT1 has reached minimum value (zero).

Depending of the mode of operation used, the counter is cleared, incremented, or decremented at each timer clock (clk_{T1}). The timer clock is generated from an synchronous system clock or an asynchronous PLL clock using the clock select bits (CS13:0) and the PCK enable bit (PCKE). When no clock source is selected (CS13:0 = 0) the timer is stopped. However, the TCNT1 value can be accessed by the CPU, regardless of whether clk_{T1} is present or not. A CPU write overrides (has priority over) all counter clear or count operations.

The counting sequence of the Timer/Counter1 is determined by the setting of the WGM10 and PWM1x bits located in the Timer/Counter1 control registers (TCCR1A, TCCR1C and TCCR1D). For more details about advanced counting sequences and waveform generation, see Section 16.7 “Modes of Operation” on page 97. The Timer/Counter overflow flag (TOV1) is set according to the mode of operation selected by the PWM1x and WGM10 bits. The overflow flag can be used for generating a CPU interrupt.

Table 16-8 shows the COM1A1:0 bit functionality when the PWM1A, WGM10 and WGM11 bits are set to Phase and Frequency Correct PWM Mode.

Table 16-8. Compare Output Mode, Phase and Frequency Correct PWM Mode

COM1A1..0	OCW1A Behavior	OC1A Pin	$\overline{\text{OC1A}}$ Pin
00	Normal port operation.	Disconnected	Disconnected
01	Cleared on compare match when up-counting Set on compare match when down-counting	Connected	Connected
10	Cleared on compare match when up-counting Set on compare match when down-counting	Connected	Disconnected
11	Set on compare match when up-counting Cleared on compare match when down-counting	Connected	Disconnected

Table 16-9 shows the COM1A1:0 bit functionality when the PWM1A, WGM10 and WGM11 bits are set to single-slope PWM6 Mode. In the PWM6 Mode the same waveform output (OCW1A) is used for generating all waveforms and the output compare values OC1A and $\overline{\text{OC1A}}$ are connected on the all OC1x and $\overline{\text{OC1x}}$ pins as described below.

Table 16-9. Compare Output Mode, Single-Slope PWM6 Mode

COM1A1..0	OCW1A Behavior	OC1x Pin	$\overline{\text{OC1x}}$ Pin
00	Normal port operation	Disconnected	Disconnected
01	Cleared on compare match Set when TCNT1 = 0x000	OC1A	OC1A
10	Cleared on compare match Set when TCNT1 = 0x000	OC1A	OC1A
11	Set on compare match Cleared when TCNT1 = 0x000	OC1A	OC1A

Table 16-10 shows the COM1A1:0 bit functionality when the PWM1A, WGM10 and WGM11 bits are set to dual-slope PWM6 Mode.

Table 16-10. Compare Output Mode, Dual-Slope PWM6 Mode

COM1A1..0	OCW1A Behavior	OC1x Pin	$\overline{\text{OC1x}}$ Pin
00	Normal port operation	Disconnected	Disconnected
01	Cleared on compare match when up-counting Set on compare match when down-counting	OC1A	OC1A
10	Cleared on compare match when up-counting Set on compare match when down-counting	OC1A	OC1A
11	Set on compare match when up-counting Cleared on compare match when down-counting	OC1A	OC1A

- **Bits 5,4 - COM1B1, COM1B0: Comparator B Output Mode, Bits 1 and 0**

These bits control the behavior of the waveform output (OCW1B) and the connection of the output compare pin (OC1B). If one or both of the COM1B1:0 bits are set, the OC1B output overrides the normal port functionality of the I/O pin it is connected to. The complementary $\overline{\text{OC1B}}$ output is connected only in PWM modes when the COM1B1:0 bits are set to “01”.

Note that the data direction register (DDR) bit corresponding to the OC1B pin must be set in order to enable the output driver.

The function of the COM1D1:0 bits depends on the PWM1D and WGM10 bit settings. Table 16-16 shows the COM1D1:0 bit functionality when the PWM1D bit is set to a normal mode (non-PWM).

Table 16-16. Compare Output Mode, Normal Mode (non-PWM)

COM1D1..0	OCW1D Behavior	OC1D Pin	$\overline{\text{OC1D}}$ Pin
00	Normal port operation	Disconnected	Disconnected
01	Toggle on compare match	Connected	Disconnected
10	Clear on compare match	Connected	Disconnected
11	Set on compare match	Connected	Disconnected

Table 16-17 shows the COM1D1:0 bit functionality when the PWM1D and WGM10 bits are set to fast PWM mode.

Table 16-17. Compare Output Mode, Fast PWM Mode

COM1D1..0	OCW1D Behavior	OC1D Pin	$\overline{\text{OC1D}}$ Pin
00	Normal port operation	Disconnected	Disconnected
01	Cleared on compare match Set when TCNT1 = 0x000	Connected	Connected
10	Cleared on compare match Set when TCNT1 = 0x000	Connected	Disconnected
11	Set on compare match Clear when TCNT1 = 0x000	Connected	Disconnected

Table 16-18 on page 113 shows the COM1D1:0 bit functionality when the PWM1D and WGM10 bits are set to phase and frequency correct PWM mode.

Table 16-18. Compare Output Mode, Phase and Frequency Correct PWM Mode

COM1D1..0	OCW1D Behavior	OC1D Pin	$\overline{\text{OC1D}}$ Pin
00	Normal port operation.	Disconnected	Disconnected
01	Cleared on compare match when up-counting Set on compare match when down-counting	Connected	Connected
10	Cleared on compare match when up-counting Set on compare match when down-counting	Connected	Disconnected
11	Set on compare match when up-counting Cleared on compare match when down-counting	Connected	Disconnected

- **Bit 1 - FOC1D: Force Output Compare Match 1D**

The FOC1D bit is only active when the PWM1D bit specify a non-PWM mode.

Writing a logical one to this bit forces a change in the waveform output (OCW1D) and the output compare pin (OC1D) according to the values already set in COM1D1 and COM1D0. If COM1D1 and COM1D0 written in the same cycle as FOC1D, the new settings will be used. The force output compare bit can be used to change the output pin value regardless of the timer value. The automatic action programmed in COM1D1 and COM1D0 takes place as if a compare match had occurred, but no interrupt is generated. The FOC1D bit is always read as zero.

- **Bit 0 - PWM1D: Pulse Width Modulator D Enable**

When set (one) this bit enables PWM mode based on comparator OCR1D.

16.11.5 TCCR1E – Timer/Counter1 Control Register E

Bit	7	6	5	4	3	2	1	0	
0x00 (0x20)	-	-	OC1OE5	OC1OE4	OC1OE3	OC1OE2	OC1OE1	OC1OE0	TCCR1E
Read/Write	R	R	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

- **Bits 7:6 - Res: Reserved Bits**

These bits are reserved bits in the ATtiny261/461/861 and always reads as zero.

- **Bits 5:0 – OC1OE5:OC1OE0: Output Compare Override Enable Bits**

These bits are the output compare override enable bits that are used to connect or disconnect the output compare pins in PWM6 modes with an instant response on the corresponding output compare pins.

The actual value from the port register will be visible on the port pin, when the output compare override enable bit is cleared. Table 16-20 shows the output compare override enable bits and their corresponding output compare pins.

Table 16-20. Output Compare Override Enable Bits versus Output Compare Pins

OC1OE0	OC1OE1	OC1OE2	OC1OE3	OC1OE4	OC1OE5
OC1A (PB0)	OC1A (PB1)	OC1B (PB2)	OC1B (PB3)	OC1D (PB4)	OC1D (PB5)

16.11.6 TCNT1 – Timer/Counter1

Bit	7	6	5	4	3	2	1	0	
0x2E (0x4E)	MSB							LSB	TCNT1
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	

This 8-bit register contains the value of Timer/Counter1.

The Timer/Counter1 is realized as a 10-bit up/down counter with read and write access. Due to synchronization of the CPU, Timer/Counter1 data written into Timer/Counter1 is delayed by one and half CPU clock cycles in synchronous mode and at most one CPU clock cycles for asynchronous mode. When a 10-bit accuracy is preferred, special procedures must be followed for accessing the 10-bit TCNT1 register via the 8-bit AVR data bus. These procedures are described in Section 16.10 “Accessing 10-Bit Registers” on page 105. Alternatively the Timer/Counter1 can be used as an 8-bit Timer/Counter. Note that the Timer/Counter1 always starts counting up after writing the TCNT1 register.

16.11.7 TC1H – Timer/Counter1 High Byte

Bit	7	6	5	4	3	2	1	0	
0x25 (0x45)	-	-	-	-	-	-	TC19	TC18	TC1H
Read/Write	R	R	R	R	R	R	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

The temporary Timer/Counter1 register is an 2-bit read/write register.

- **Bits 7:2 - Res: Reserved Bits**

These bits are reserved bits in the ATtiny261/461/861 and always reads as zero.

- **Bits 1:0 - TC19, TC18: Two MSB bits of the 10-bit accesses**

If 10-bit accuracy is used, the Timer/Counter1 high byte register (TC1H) is used for temporary storing the MSB bits (TC19, TC18) of the 10-bit accesses. The same TC1H register is shared between all 10-bit registers within the Timer/Counter1. Note that special procedures must be followed when accessing the 10-bit TCNT1 register via the 8-bit AVR data bus. These procedures are described in Section 16.10 “Accessing 10-Bit Registers” on page 105.

16.11.8 OCR1A – Timer/Counter1 Output Compare Register A

Bit	7	6	5	4	3	2	1	0	
0x2D (0x4D)	MSB							LSB	OCR1A
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 output compare register A is an 8-bit read/write register.

The Timer/Counter output compare register A contains data to be continuously compared with Timer/Counter1. Actions on compare matches are specified in TCCR1A. A compare match does only occur if Timer/Counter1 counts to the OCR1A value. A software write that sets TCNT1 and OCR1A to the same value does not generate a compare match.

A compare match will set the compare interrupt flag OCF1A after a synchronization delay following the compare event.

Note that, if 10-bit accuracy is used special procedures must be followed when accessing the internal 10-bit output compare registers via the 8-bit AVR data bus. These procedures are described in Section 16.10 “Accessing 10-Bit Registers” on page 105.

16.11.9 OCR1B – Timer/Counter1 Output Compare Register B

Bit	7	6	5	4	3	2	1	0	
0x2C (0x4C)	MSB							LSB	OCR1B
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 output compare register B is an 8-bit read/write register.

The Timer/Counter output compare register B contains data to be continuously compared with Timer/Counter1. Actions on compare matches are specified in TCCR1. A compare match does only occur if Timer/Counter1 counts to the OCR1B value. A software write that sets TCNT1 and OCR1B to the same value does not generate a compare match.

A compare match will set the compare interrupt flag OCF1B after a synchronization delay following the compare event.

Note that, if 10-bit accuracy is used special procedures must be followed when accessing the internal 10-bit output compare registers via the 8-bit AVR data bus. These procedures are described in Section 16.10 “Accessing 10-Bit Registers” on page 105.

16.11.10 OCR1C – Timer/Counter1 Output Compare Register C

Bit	7	6	5	4	3	2	1	0	
0x2B (0x4B)	MSB							LSB	OCR1C
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	1	1	1	1	1	1	1	1	

The output compare register C is an 8-bit read/write register.

The Timer/Counter output compare register C contains data to be continuously compared with Timer/Counter1, and a compare match will clear TCNT1. This register has the same function in normal mode and PWM modes.

Note that, if a smaller value than three is written to the output compare register C, the value is automatically replaced by three as it is a minimum value allowed to be written to this register.

Note that, if 10-bit accuracy is used special procedures must be followed when accessing the internal 10-bit output compare registers via the 8-bit AVR data bus. These procedures are described in Section 16.10 “Accessing 10-Bit Registers” on page 105.

Figure 19-6. ADC Timing Diagram, Auto Triggered Conversion

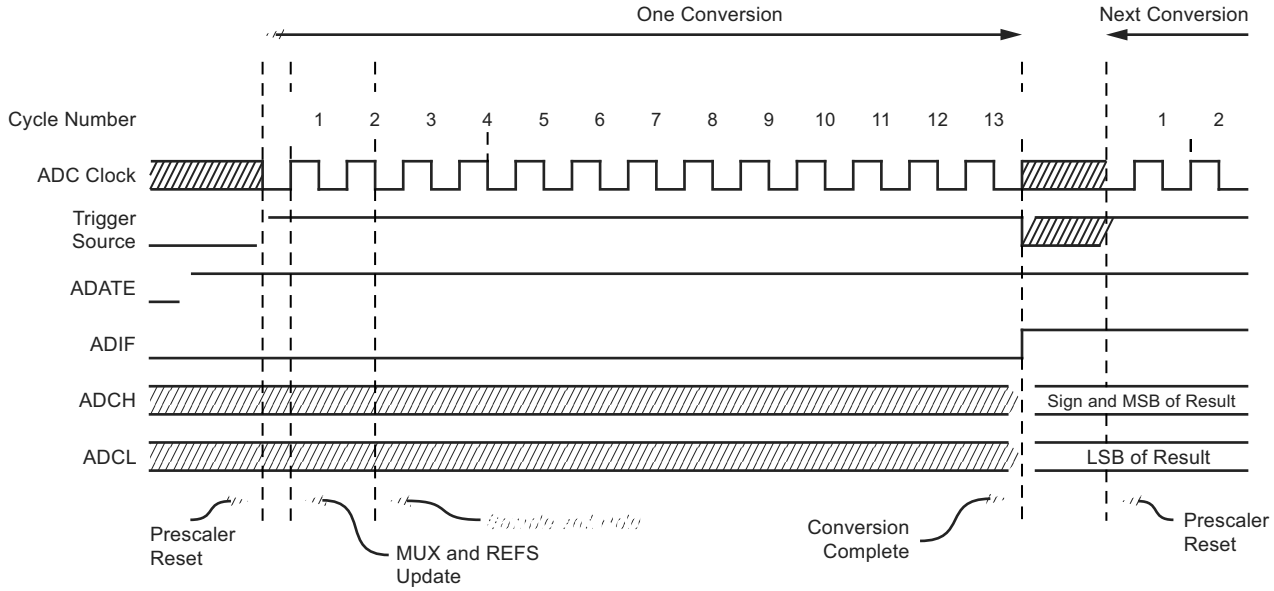


Figure 19-7. ADC Timing Diagram, Free Running Conversion

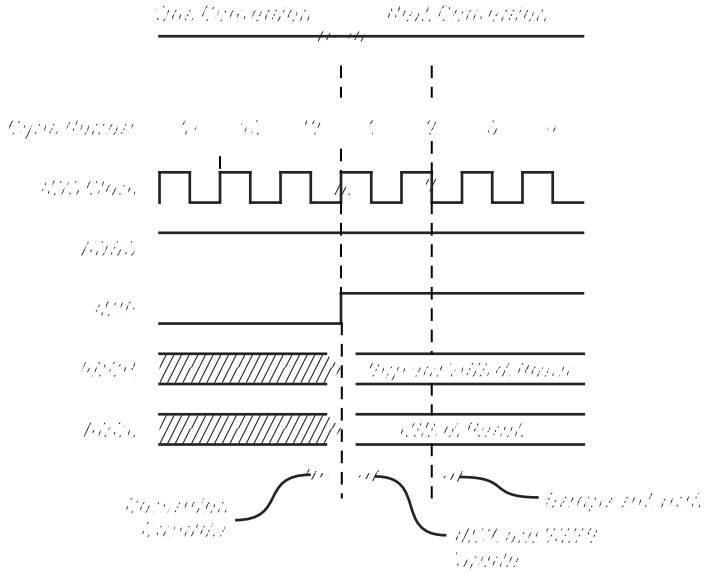


Table 19-1. ADC Conversion Time

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

- **Bit 5 – ADATE: ADC Auto Trigger Enable**

When this bit is written to one, auto triggering of the ADC is enabled. The ADC will start a conversion on a positive edge of the selected trigger signal. The trigger source is selected by setting the ADC trigger select bits, ADTS in ADCSRB.

- **Bit 4 – ADIF: ADC Interrupt Flag**

This bit is set when an ADC conversion completes and the data registers are updated. The ADC conversion complete interrupt is executed if the ADIE bit and the I-bit in SREG are set. ADIF is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, ADIF is cleared by writing a logical one to the flag. Beware that if doing a read-modify-write on ADCSRA, a pending interrupt can be disabled. This also applies if the SBI and CBI instructions are used.

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

When this bit is written to one and the I-bit in SREG is set, the ADC conversion complete interrupt is activated.

- **Bits 2:0 – ADPS2:0: ADC Prescaler Select Bits**

These bits determine the division factor between the system clock frequency and the input clock to the ADC.

Table 19-6. ADC Prescaler Selections

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

19.10.3 ADCL and ADCH – The ADC Data Register

19.10.3.1 ADLAR = 0

Bit	15	14	13	12	11	10	9	8	
0x05 (0x25)	–	–	–	–	–	–	**ADC9**	**ADC8**	**ADCH**
0x04 (0x24)	**ADC7**	**ADC6**	**ADC5**	**ADC4**	**ADC3**	**ADC2**	**ADC1**	**ADC0**	**ADCL**
	7	6	5	4	3	2	1	0	
Read/Write	R	R	R	R	R	R	R	R	
	R	R	R	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

24.9 Internal Oscillator Speed

Figure 24-19. Watchdog Oscillator Frequency versus Temperature

Figure 24-20. Calibrated 8.0MHz RC Oscillator Frequency versus Temperature