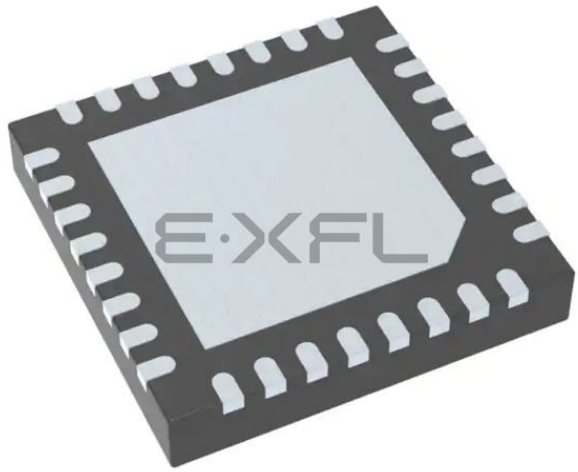


Welcome to E-XFL.COM

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 ~ 125°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/attiny461-15mz

1.1 Disclaimer

Typical values contained in this data sheet are based on simulations and characterization of other AVR® microcontrollers manufactured on the same process technology. Min and Max values will be available after the device is characterized.

1.2 Automotive Quality Grade

The Atmel® ATtiny261/461/861 have been developed and manufactured according to the most stringent requirements of the international standard ISO-TS 16949. This data sheet contains limit values extracted from the results of extensive characterization (temperature and voltage). The quality and reliability of the Atmel ATtiny261/461/861 have been verified during regular product qualification as per AEC-Q100 grade 1.

As indicated in the ordering information paragraph, the product is available in only one temperature grade, see Table 1-1.

Table 1-1. Temperature Grade Identification for Automotive Products

Temperature	Temperature Identifier	Comments
–40; +125	Z	Full automotive temperature range

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.

5.5.1 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

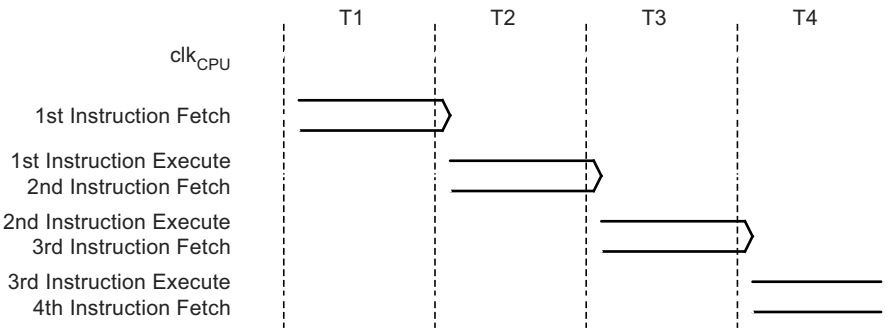
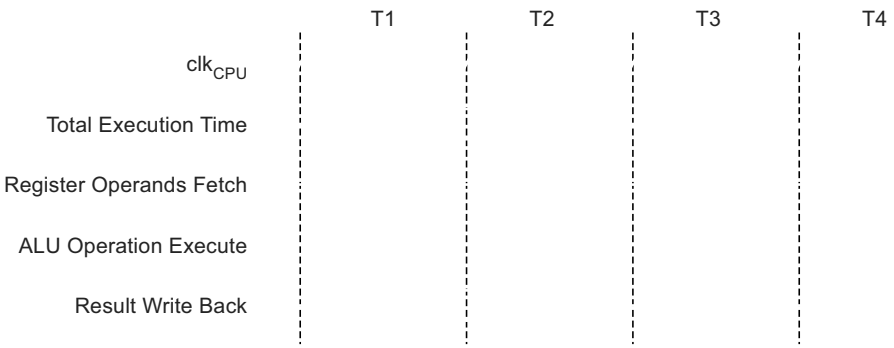


Figure 5-5 shows the internal timing concept for the register file. In a single clock cycle an ALU operation using two register operands is executed, and the result is stored back to the destination register.

Figure 5-5. Single Cycle ALU Operation



The following code examples show one assembly and one C function for erase, write, or atomic write of the EEPROM. The examples assume that interrupts are controlled (e.g., by disabling interrupts globally) so that no interrupts will occur during execution of these functions.

Assembly Code Example

```
EEPROM_write:
        ; Wait for completion of previous write
        sbic    EECR,EEPE
        rjmp    EEPROM_write
        ; Set Programming mode
        ldi     r16, (0<<EEPM1)|(0<<EEPM0)
        out     EECR, r16
        ; Set up address (r18:r17) in address register
        out     EEARH, r18
        out     EEARL, r17
        ; Write data (r16) to data register
        out     EEDR, r16
        ; Write logical one to EEMPE
        sbi     EECR,EEMPE
        ; Start eeprom write by setting EEPE
        sbi     EECR,EEPE
        ret
```

C Code Example

```
void EEPROM_write(unsigned char ucAddress, unsigned char ucData)
{
        /* Wait for completion of previous write */
        while(EECR & (1<<EEPE))
        ;
        /* Set Programming mode */
        EECR = (0<<EEPM1)|(0<<EEPM0);
        /* Set up address and data registers */
        EEAR = ucAddress;
        EEDR = ucData;
        /* Write logical one to EEMPE */
        EECR |= (1<<EEMPE);
        /* Start eeprom write by setting EEPE */
        EECR |= (1<<EEPE);
}
```

6.4 I/O Memory

The I/O space definition of the Atmel® ATtiny261/461/861 is shown in Section 25. “Register Summary” on page 189.

All Atmel ATtiny261/461/861 I/Os and peripherals are placed in the I/O space. All I/O locations may be accessed by the LD/LDS/LDD and ST/STS/STD instructions, transferring data between the 32 general purpose working registers and the I/O space. I/O registers within the address range 0x00 - 0x1F are directly bit-accessible using the SBI and CBI instructions. In these registers, the value of single bits can be checked by using the SBIS and SBIC instructions. Refer to the instruction set section for more details. When using the I/O specific commands IN and OUT, the I/O addresses 0x00 - 0x3F must be used. When addressing I/O registers as data space using LD and ST instructions, 0x20 must be added to these addresses.

For compatibility with future devices, reserved bits should be written to zero if accessed. Reserved I/O memory addresses should never be written.

Some of the status flags are cleared by writing a logical one to them. Note that, the CBI and SBI instructions will only operate on the specified bit, and can therefore be used on registers containing such status flags. The CBI and SBI instructions work with registers 0x00 to 0x1F only.

The I/O and peripherals control registers are explained in later sections.

6.4.1 General Purpose I/O Registers

The Atmel ATtiny261/461/861 contains three general purpose I/O registers. These registers can be used for storing any information, and they are particularly useful for storing global variables and status flags. General purpose I/O registers within the address range 0x00 - 0x1F are directly bit-accessible using the SBI, CBI, SBIS, and SBIC instructions.

6.5 Register Description

6.5.1 EEARH and EEARL – EEPROM Address Register

Bit	7	6	5	4	3	2	1	0	
0x1F (0x3F)	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**EEAR8**	**EEARH**
0x1E (0x3E)	**EEAR7**	**EEAR6**	**EEAR5**	**EEAR4**	**EEAR3**	**EEAR2**	**EEAR1**	**EEAR0**	**EEARL**
Bit	7	6	5	4	3	2	1	0	
Read/Write	R	R	R	R	R	R	R	R/W	
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	X	
Initial Value	X	X	X	X	X	X	X	X	

- **Bit 7:1 – Res6:0: Reserved Bits**

These bits are reserved for future use and will always read as 0 in Atmel ATtiny261/461/861.

- **Bits 8:0 – EEAR8:0: EEPROM Address**

The EEPROM address registers – EEARH and EEARL – specifies the high EEPROM address in the 128/256/512 bytes EEPROM space. The EEPROM data bytes are addressed linearly between 0 and 127/255/511. The initial value of EEAR is undefined. A proper value must be written before the EEPROM may be accessed.

- **Bit 2 – EEMPE: EEPROM Master Program Enable**

The EEMPE bit determines whether writing EEPE to one will have effect or not.

When EEMPE is set, setting EEPE within four clock cycles will program the EEPROM at the selected address. If EEMPE is zero, setting EEPE will have no effect. When EEMPE has been written to one by software, hardware clears the bit to zero after four clock cycles.

- **Bit 1 – EEPE: EEPROM Program Enable**

The EEPROM program enable signal EEPE is the programming enable signal to the EEPROM. When EEPE is written, the EEPROM will be programmed according to the EEPMn bits setting. The EEMPE bit must be written to one before a logical one is written to EEPE, otherwise no EEPROM write takes place. When the write access time has elapsed, the EEPE bit is cleared by hardware. When EEPE has been set, the CPU is halted for two cycles before the next instruction is executed.

- **Bit 0 – EERE: EEPROM Read Enable**

The EEPROM read enable signal – EERE – is the read strobe to the EEPROM. When the correct address is set up in the EEAR register, the EERE bit must be written to one to trigger the EEPROM read. The EEPROM read access takes one instruction, and the requested data is available immediately. When the EEPROM is read, the CPU is halted for four cycles before the next instruction is executed. The user should poll the EEPE bit before starting the read operation. If a write operation is in progress, it is neither possible to read the EEPROM, nor to change the EEAR register.

6.5.4 GPIOR2 – General Purpose I/O Register 2

Bit	7	6	5	4	3	2	1	0	
0x0C (0x2C)	**MSB**							**LSB**	**GPIOR2**
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	

6.5.5 GPIOR1 – General Purpose I/O Register 1

Bit	7	6	5	4	3	2	1	0	
0x0B (0x2B)	**MSB**							**LSB**	**GPIOR1**
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	

6.5.6 GPIOR0 – General Purpose I/O Register 0

Bit	7	6	5	4	3	2	1	0	
0x0A (0x2A)	**MSB**							**LSB**	**GPIOR0**
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	

7.6 Calibrated Internal RC Oscillator

By default, the internal RC oscillator provides an approximate 8.0MHz clock. Though voltage and temperature dependent, this clock can be very accurately calibrated by the user. See Table 23-1 on page 173 and Section 24.9 “Internal Oscillator Speed” on page 188 for more details. The device is shipped with the CKDIV8 fuse programmed. See Section 7.11 “System Clock Prescaler” on page 31 for more details.

This clock may be selected as the system clock by programming the CKSEL fuses as shown in Table 7-6. If selected, it will operate with no external components. During reset, hardware loads the pre-programmed calibration value into the OSCCAL register and thereby automatically calibrates the RC oscillator. The accuracy of this calibration is shown as factory calibration in Table 23-1 on page 173.

By changing the OSCCAL register from SW, see Section 7.12.1 “OSCCAL – Oscillator Calibration Register” on page 31, it is possible to get a higher calibration accuracy than by using the factory calibration. The accuracy of this calibration is shown as User calibration in Table 23-1 on page 173.

When this oscillator is used as the chip clock, the watchdog oscillator will still be used for the watchdog timer and for the reset time-out. For more information on the pre-programmed calibration value, see Section 22.4 “Calibration Byte” on page 158.

Table 7-6. Internal Calibrated RC Oscillator Operating Modes(1)(2)

Frequency Range (MHz)	CKSEL3..0
7.84 - 8.16	0010

Notes: 1. The device is shipped with this option selected.
2. If 8MHz frequency exceeds the specification of the device (depends on V_{CC}), the CKDIV8 Fuse can be programmed in order to divide the internal frequency by 8.

When this oscillator is selected, start-up times are determined by the SUT fuses as shown in Table 7-7.

Table 7-7. Start-up Times for the Internal Calibrated RC Oscillator Clock Selection

SUT1..0	Start-up Time from Power-down	Additional Delay from Reset (V_{CC} = 5.0V)	Recommended Usage
00	6CK	14CK	BOD enabled
01	6CK	14CK + 4ms	Fast rising power
10(1)	6CK	14CK + 64ms	Slowly rising power
11	Reserved		

Note: 1. The device is shipped with this option selected.

8.7.4 Internal Voltage Reference

The internal voltage reference will be enabled when needed by the brown-out detection, the analog comparator or the ADC. If these modules are disabled as described in the sections above, the internal voltage reference will be disabled and it will not be consuming power. When turned on again, the user must allow the reference to start up before the output is used. If the reference is kept on in sleep mode, the output can be used immediately. Refer to
Section 9.7 “Internal Voltage Reference” on page 42 for details on the start-up time.

8.7.5 Watchdog Timer

If the watchdog timer is not needed in the application, this 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 9.8 “Watchdog Timer” on page 42 for details on how to configure the Watchdog Timer.

8.7.6 Port Pins

When entering a sleep mode, all port pins should be configured to use minimum power. The most important thing 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 Section 12.2.5 “Digital Input Enable and Sleep Modes” on page 56 for details on which pins are enabled. If the input buffer is enabled and the input signal is left floating or has 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 (DIDR0, DIDR1). Refer to Section 19.10.5 “DIDR0 – Digital Input Disable Register 0” on page 149 or Section 19.10.6 “DIDR1 – Digital Input Disable Register 1” on page 149 for details.

8.8 Register Description

8.8.1 MCUCR – MCU Control Register

The MCU control register contains control bits for power management.

Bit	7	6	5	4	3	2	1	0	
0x35 (0x55)	**–**	**PUD**	**SE**	**SM1**	**SM0**	**—**	**ISC01**	**ISC00**	**MCUCR**
Read/Write	R	R/W	R/W	R/W	R/W	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 5 – SE: Sleep Enable**

The SE bit must be written to logic one to make the MCU enter the sleep mode when the SLEEP instruction is executed. To avoid the MCU entering the sleep mode unless it is the programmer’s purpose, it is recommended to write the sleep enable (SE) bit to one just before the execution of the SLEEP instruction and to clear it immediately after waking up.

- **Bits 4, 3 – SM1:0: Sleep Mode Select Bits 2..0**

These bits select between the three available sleep modes as shown in Table 8-2 on page 37.

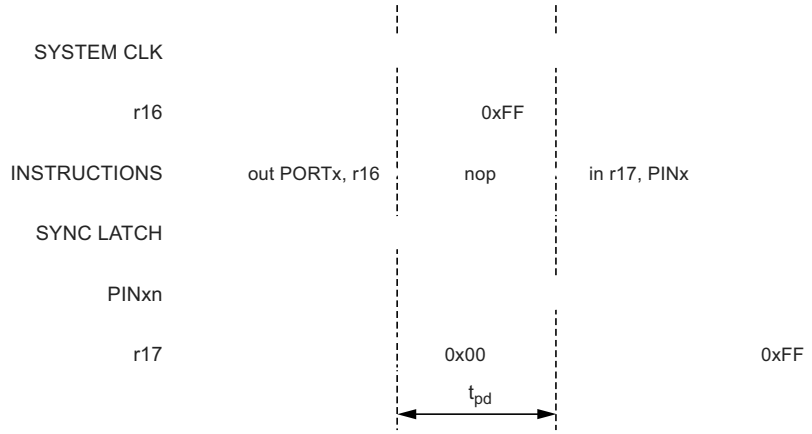
If the program never enables an interrupt source, the interrupt vectors are not used, and regular program code can be placed at these locations. The most typical and general program setup for the reset and interrupt vector addresses in Atmel® ATtiny261/461/861 is:

```
Address      Labels Code                Comments
0x0000              rjmp   RESET        ; Reset Handler
0x0001              rjmp   EXT_INT0     ; IRQ0 Handler
0x0002              rjmp   PCINT        ; PCINT Handler
0x0003              rjmp   TIM1_COMPA   ; Timer1 CompareA Handler
0x0004              rjmp   TIM1_COMPB   ; Timer1 CompareB Handler
0x0005              rjmp   TIM1_OVF     ; Timer1 Overflow Handler
0x0006              rjmp   TIM0_OVF     ; Timer0 Overflow Handler
0x0007              rjmp   USI_START    ; USI Start Handler
0x0008              rjmp   USI_OVF      ; USI Overflow Handler
0x0009              rjmp   EE_RDY       ; EEPROM Ready Handler
0x000A              rjmp   ANA_COMP     ; Analog Comparator Handler
0x000B              rjmp   ADC          ; ADC Conversion Handler
0x000C              rjmp   WDT          ; WDT Interrupt Handler
0x000D              rjmp   EXT_INT1     ; IRQ1 Handler
0x000E              rjmp   TIM0_COMPA   ; Timer0 CompareA Handler
0x000F              rjmp   TIM0_COMPB   ; Timer0 CompareB Handler
0x0010              rjmp   TIM0_CAPT    ; Timer0 Capture Event Handler
0x0011              rjmp   TIM1_COMPD   ; Timer1 CompareD Handler
0x0012              rjmp   FAULT_PROTECTION; Timer1 Fault Protection
0x0013       RESET: ldi    r16, low(RAMEND); Main program start
0x0014              ldi    r17, high(RAMEND); Tiny861 have also SPH
0x0015              out    SPL, r16     ; Set Stack Pointer to top of RAM
0x0016              out    SPH, r17     ; Tiny861 have also SPH
0x0017              sei                 ; Enable interrupts
0x0018       <instr> xxx
...    ...    ...    ...
```

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

When reading back a software assigned pin value, a nop instruction must be inserted as indicated in Figure 12-4. The out instruction sets the “SYNC LATCH” signal at the positive edge of the clock. In this case, the delay tpd through the synchronizer is one system clock period.

Figure 12-4. Synchronization when Reading a Software Assigned Pin Value



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.

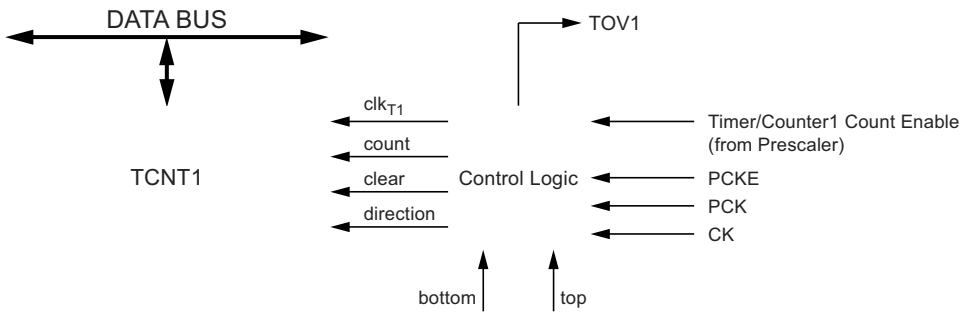
Table 16-1. Definitions

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.

16.3.1 Counter Initialization for Asynchronous Mode

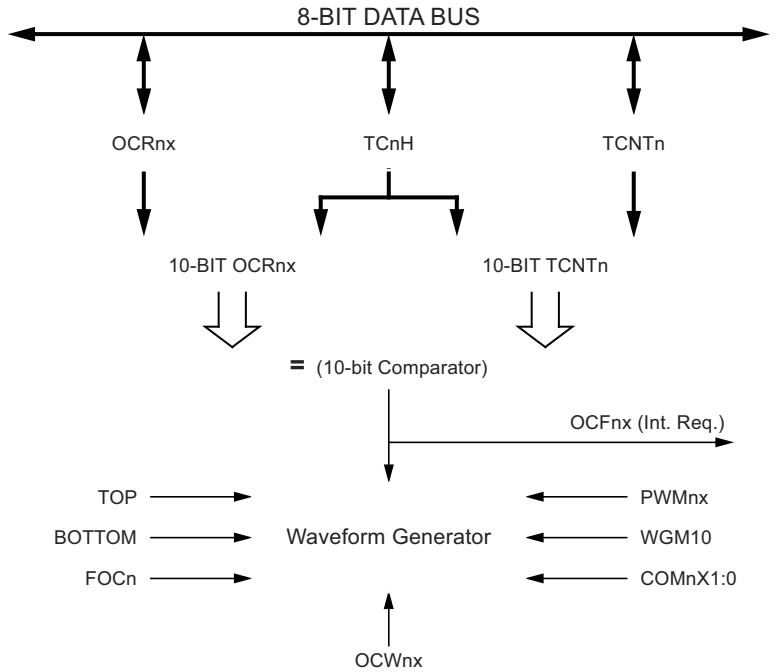
To change Timer/Counter1 to the asynchronous mode follow the procedure below:

1. Enable PLL.
2. Wait 100µs for PLL to stabilize.
3. Poll the PLOCK bit until it is set.
4. Set the PCKE bit in the PLLCSR register which enables the asynchronous mode.

16.4 Output Compare Unit

The comparator continuously compares TCNT1 with the output compare registers (OCR1A, OCR1B, OCR1C and OCR1D). Whenever TCNT1 equals to the output compare register, the comparator signals a match. A match will set the output compare flag (OCF1A, OCF1B or OCF1D) at the next timer clock cycle. If the corresponding interrupt is enabled, the output compare flag generates an output compare interrupt. The output compare flag is automatically cleared when the interrupt is executed. Alternatively, the 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 PWM1x, WGM10 and compare output mode (COM1x1:0) bits. The top and bottom signals are used by the waveform generator for handling the special cases of the extreme values in some modes of operation (see Section 16.7 “Modes of Operation” on page 97). Figure 16-4 shows a block diagram of the output compare unit.

Figure 16-4. Output Compare Unit, Block Diagram



The OCR1x registers are double buffered when using any of the pulse width modulation (PWM) modes. For the normal mode of operation, the double buffering is disabled. The double buffering synchronizes the update of the OCR1x 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. See Figure 16-5 for an example. During the time between the write and the update operation, a read from OCR1A, OCR1B, OCR1C or OCR1D will read the contents of the temporary location. This means that the most recently written value always will read out of OCR1A, OCR1B, OCR1C or OCR1D.

The Timer/Counter overflow flag (TOV1) is set each time the counter reaches TOP. If the interrupt is enabled, the interrupt handler routine can be used for updating the compare value.

In fast PWM mode, the compare unit allows generation of PWM waveforms on the OC1x pins. Setting the COM1x1:0 bits to two will produce a non-inverted PWM and setting the COM1x1:0 to three will produce an inverted PWM output. Setting the COM1x1:0 bits to one will enable complementary compare output mode and produce both the non-inverted (OC1x) and inverted output ($\overline{OC1x}$). The actual value will only be visible on the port pin if the data direction for the port pin is set as output. The PWM waveform is generated by setting (or clearing) the waveform output (OCW1x) at the compare match between OCR1x and TCNT1, and clearing (or setting) the waveform output at the timer clock cycle the counter is cleared (changes from TOP to BOTTOM).

The PWM frequency for the output can be calculated by the following equation:

$$f_{OCnxPWM} = \frac{f_{\mathrm{clkT1}}}{N}$$

The *N* variable represents the number of steps in single-slope operation. The value of *N* equals either to the TOP value.

The extreme values for the OCR1C register represents special cases when generating a PWM waveform output in the fast PWM mode. If the OCR1C is set equal to BOTTOM, the output will be a narrow spike for each MAX+1 timer clock cycle. Setting the OCR1C equal to MAX will result in a constantly high or low output (depending on the polarity of the output set by the COM1x1:0 bits.)

A frequency (with 50% duty cycle) waveform output in fast PWM mode can be achieved by setting the waveform output (OCW1x) to toggle its logical level on each Compare Match (COM1x1:0 = 1). The waveform generated will have a maximum frequency of $f_{OC1} = f_{clkT1}/4$ when OCR1C is set to three.

The general I/O port function is overridden by the output compare value (OC1x / $\overline{OC1x}$) from the dead time generator, if either of the COM1x1:0 bits are set and the data direction register bits for the OC1X and $\overline{OC1X}$ pins are set as an output. If the COM1x1:0 bits are cleared, the actual value from the port register will be visible on the port pin. The output compare pin configurations are described in Table 16-3.

Table 16-3. Output Compare Pin Configurations in Fast PWM Mode

COM1x1	COM1x0	$\overline{OC1x}$ Pin	OC1x Pin
0	0	Disconnected	Disconnected
0	1	OC1x	OC1x
1	0	Disconnected	OC1x
1	1	Disconnected	OC1x

16.7.3 Phase and Frequency Correct PWM Mode

The phase and frequency correct PWM mode (PWMx = 1 and WGM10 = 1) provides a high resolution phase and frequency correct PWM waveform generation option. The phase and frequency correct PWM mode is based on a dual-slope operation. The counter counts repeatedly from BOTTOM to TOP (defined as OCR1C) and then from TOP to BOTTOM. In non-inverting compare output mode the waveform output (OCW1x) is cleared on the compare match between TCNT1 and OCR1x while upcounting, and set on the compare match while down-counting. In inverting output compare mode, the operation is inverted. In complementary compare output mode, the waveform output is cleared on the compare match and set at BOTTOM. The dual-slope operation has lower maximum operation frequency than single slope operation. However, due to the symmetric feature of the dual-slope PWM modes, these modes are preferred for motor control applications.

The timing diagram for the phase and frequency correct PWM mode is shown on Figure 16-12 in which the TCNT1 value is shown as a histogram for illustrating the dual-slope operation. The counter is incremented until the counter value matches TOP. When the counter reaches TOP, it changes the count direction. The TCNT1 value will be equal to TOP for one timer clock cycle. The diagram includes the waveform output (OCW1x) in non-inverted and inverted compare output mode. The small horizontal line marks on the TCNT1 slopes represent compare matches between OCR1x and TCNT1.

The function of the COM1B1:0 bits depends on the PWM1B and WGM10 bit settings. Table 16-11 shows the COM1B1:0 bit functionality when the PWM1B bit is set to normal mode (non-PWM).

Table 16-11. Compare Output Mode, Normal Mode (non-PWM)

COM1B1..0	OCW1B Behavior	OC1B Pin	$\overline{\text{OC1B}}$ 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 16-12 shows the COM1B1:0 bit functionality when the PWM1B and WGM10 bits are set to Fast PWM Mode.

Table 16-12. Compare Output Mode, Fast PWM Mode

COM1B1..0	OCW1B Behavior	OC1B Pin	$\overline{\text{OC1B}}$ 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 Cleared when TCNT1 = 0x000	Connected	Disconnected

Table 16-13 shows the COM1B1:0 bit functionality when the PWM1B and WGM10 bits are set to phase and frequency correct PWM mode.

Table 16-13. Compare Output Mode, Phase and Frequency Correct PWM Mode

COM1B1..0	OCW1B Behavior	OC1B Pin	$\overline{\text{OC1B}}$ 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 3 - FOC1A: Force Output Compare Match 1A**

The FOC1A bit is only active when the PWM1A bit specify a non-PWM mode.

Writing a logical one to this bit forces a change in the waveform output (OCW1A) and the output compare pin (OC1A) according to the values already set in COM1A1 and COM1A0. If COM1A1 and COM1A0 written in the same cycle as FOC1A, 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 COM1A1 and COM1A0 takes place as if a compare match had occurred, but no interrupt is generated. The FOC1A bit is always read as zero.

- **Bit 2 - FOC1B: Force Output Compare Match 1B**

The FOC1B bit is only active when the PWM1B bit specify a non-PWM mode.

Writing a logical one to this bit forces a change in the waveform output (OCW1B) and the output compare pin (OC1B) according to the values already set in COM1B1 and COM1B0. If COM1B1 and COM1B0 written in the same cycle as FOC1B, 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 COM1B1 and COM1B0 takes place as if a compare match had occurred, but no interrupt is generated.

The FOC1B bit is always read as zero.

- **Bit 1 - PWM1A: Pulse Width Modulator A Enable**

When set (one) this bit enables PWM mode based on comparator OCR1A

- **Bit 0 - PWM1B: Pulse Width Modulator B Enable**

When set (one) this bit enables PWM mode based on comparator OCR1B.

16.11.2 TCCR1B – Timer/Counter1 Control Register B

Bit	7	6	5	4	3	2	1	0	
0x2F (0x4F)	**PWM1X**	**PSR1**	**DTPS11**	**DTPS10**	**CS13**	**CS12**	**CS11**	**CS10**	**TCCR1B**
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 - PWM1X: PWM Inversion Mode**

When this bit is set (one), the PWM Inversion Mode is selected and the dead time generator outputs, OC1x and $\overline{OC1x}$ are inverted.

- **Bit 6 - PSR1: Prescaler Reset Timer/Counter1**

When this bit is set (one), the Timer/Counter1 prescaler (TCNT1 is unaffected) will be reset. The bit will be cleared by hardware after the operation is performed. Writing a zero to this bit will have no effect. This bit will always read as zero.

- **Bits 5,4 - DTPS11, DTPS10: Dead Time Prescaler Bits**

The Timer/Counter1 control register B is a 8-bit read/write register.

The dedicated dead time prescaler in front of the dead time generator can divide the Timer/Counter1 clock (PCK or CK) by 1, 2, 4 or 8 providing a large range of dead times that can be generated. The dead time prescaler is controlled by two bits DTPS11 and DTPS10 from the dead time Prescaler register. These bits define the division factor of the dead time prescaler. The division factors are given in Table 16-14.

Table 16-14. Division factors of the Dead Time prescaler

DTPS11	DTPS10	Prescaler divides the T/C1 clock by
0	0	1x (no division)
0	1	2x
1	0	4x
1	1	8x

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.

19.10.3.2ADLAR = 1

Bit	15	14	13	12	11	10	9	8	
0x05 (0x25)	ADC9	ADC8	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2	ADCH
0x04 (0x24)	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	

When an ADC conversion is complete, the result is found in these two registers.

When ADCL is read, the ADC data register is not updated until ADCH is read. Consequently, if the result is left adjusted and no more than 8-bit precision is required, it is sufficient to read ADCH. Otherwise, ADCL must be read first, then ADCH.

The ADLAR bit in ADMUX, and the MUXn bits in ADMUX affect the way the result is read from the registers. If ADLAR is set, the result is left adjusted. If ADLAR is cleared (default), the result is right adjusted.

- **ADC9:0: ADC Conversion Result**

These bits represent the result from the conversion, as detailed in Section 19.8 “ADC Conversion Result” on page 142.

19.10.4 ADCSRB – ADC Control and Status Register B

Bit	7	6	5	4	3	2	1	0	
0x03 (0x23)	BIN	GSEL	-	REFS2	MUX5	ADTS2	ADTS1	ADTS0	ADCSRB
Read/Write	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7– BIN: Bipolar Input Mode**

The gain stage is working in the unipolar mode as default, but the bipolar mode can be selected by writing the BIN bit in the ADCSRB register. In the unipolar mode only one-sided conversions are supported and the voltage on the positive input must always be larger than the voltage on the negative input. Otherwise the result is saturated to the voltage reference. In the bipolar mode two-sided conversions are supported and the result is represented in the two’s complement form. In the unipolar mode the resolution is 10 bits and the bipolar mode the resolution is 9 bits + 1 sign bit.

- **Bits 6 – GSEL: Gain Select**

The gain select bit selects the 32x gain instead of the 20x gain and the 8x gain instead of the 1x gain when the gain select bit is written to one.

- **Bits 5 – Res: Reserved Bit**

This bit is a reserved bit in the ATtiny261/461/861 and will always read as zero.

- **Bits 4 – REFS2: Reference Selection Bit**

These bit selects either the voltage reference of 1.1 V or 2.56 V for the ADC, as shown in Table 19-4. If active channels are used, using AVCC or an external AREF higher than (AVCC - 1V) is not recommended, as this will affect ADC accuracy. The internal voltage reference options may not be used if an external voltage is being applied to the AREF pin.

- **Bits 3 – MUX5: Analog Channel and Gain Selection Bit 5**

The MUX5 bit is the MSB of the analog channel and gain selection bits. Refer to Table 19-5 for details. If this bit is changed during a conversion, the change will not go into effect until this conversion is complete (ADIF in ADCSRA is set).

21.4 Addressing the Flash During Self-Programming

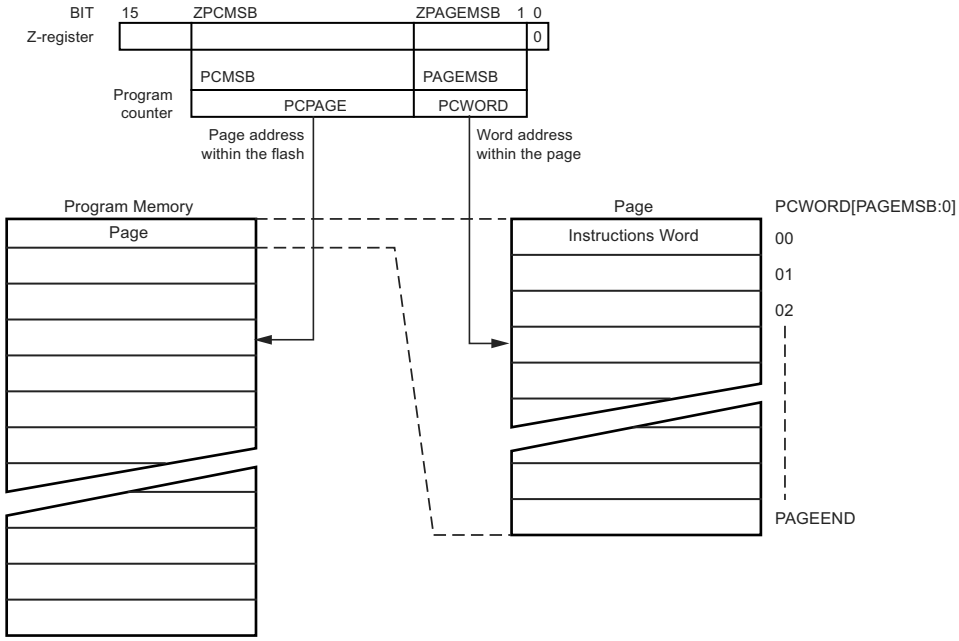
The Z-pointer is used to address the SPM commands.

Bit	15	14	13	12	11	10	9	8
ZH (R31)	**Z15**	**Z14**	**Z13**	**Z12**	**Z11**	**Z10**	**Z9**	**Z8**
ZL (R30)	**Z7**	**Z6**	**Z5**	**Z4**	**Z3**	**Z2**	**Z1**	**Z0**
	7	6	5	4	3	2	1	0

Since the flash is organized in pages (see Table 22-7 on page 159), the program counter can be treated as having two different sections. One section, consisting of the least significant bits, is addressing the words within a page, while the most significant bits are addressing the pages. This is shown in Figure 21-1. Note that the page erase and page write operations are addressed independently. Therefore it is of major importance that the software addresses the same page in both the page erase and page write operation.

The LPM instruction uses the Z-pointer to store the address. Since this instruction addresses the flash byte-by-byte, also the LSB (bit Z0) of the Z-pointer is used.

Figure 21-1. Addressing the Flash During SPM(1)



Note: 1. The different variables used in Figure 21-1 are listed in Table 22-7 on page 159.

21.4.1 EEPROM Write Prevents Writing to SPMCSR

Note that an EEPROM write operation will block all software programming to flash. Reading the fuses and lock bits from software will also be prevented during the EEPROM write operation. It is recommended that the user checks the status bit (EEPE) in the EECR register and verifies that the bit is cleared before writing to the SPMCSR register.

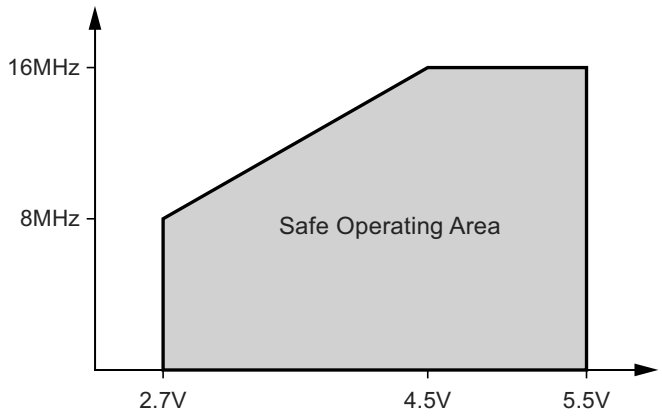
21.4.2 Reading the Fuse and Lock Bits from Software

It is possible to read both the fuse and lock bits from software. To read the lock bits, load the Z-pointer with 0x0001 and set the RFLB and SPMEN bits in SPMCSR. When an LPM instruction is executed within three CPU cycles after the RFLB and SPMEN bits are set in SPMCSR, the value of the lock bits will be loaded in the destination register. The RFLB and SPMEN bits will auto-clear upon completion of reading the lock bits or if no LPM instruction is executed within three CPU cycles or no SPM instruction is executed within four CPU cycles. When RFLB and SPMEN are cleared, LPM will work as described in the instruction set manual.

Bit	7	6	5	4	3	2	1	0
Rd	**–**	**–**	**–**	**–**	**–**	**–**	**LB2**	**LB1**

23.3 Speed Grades

Figure 23-1. Maximum Frequency versus V_{CC}



23.4 Clock Characteristics

23.4.1 Calibrated Internal RC Oscillator Accuracy

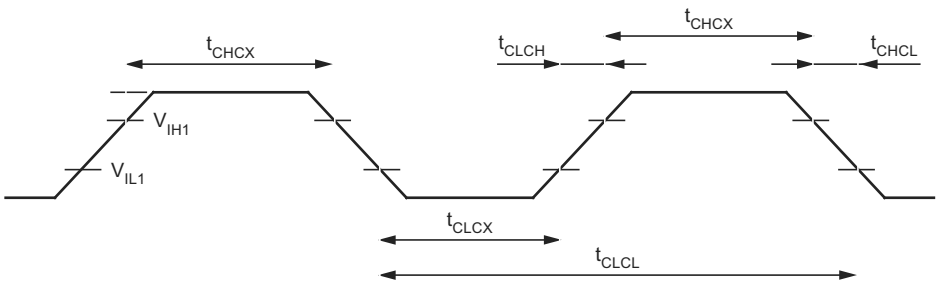
Table 23-1. Calibration Accuracy of Internal RC Oscillator

	Frequency	V_{CC}	Temperature	Calibration Accuracy
Factory Calibration	8.0MHz	3V	25°C	±2%
		2.7V to 5.5V(1)	–40°C to +125°C	±17%

Notes: 1. Voltage range for ATtiny261/461/861.

23.4.2 External Clock Drive Waveforms

Figure 23-2. External Clock Drive Waveforms



23.8 Serial Programming Characteristics

Figure 23-6. Serial Programming Waveforms

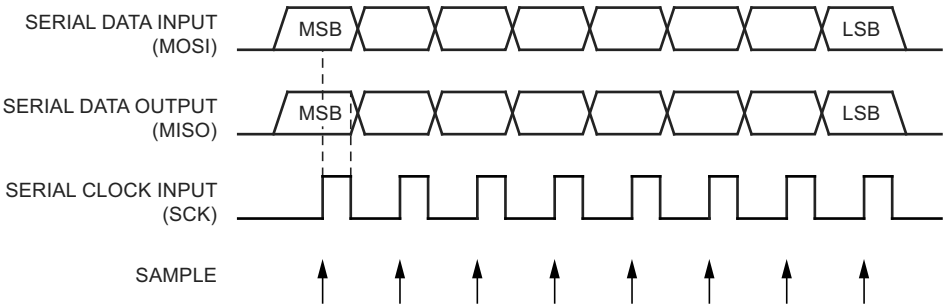


Figure 23-7. Serial Programming Timing

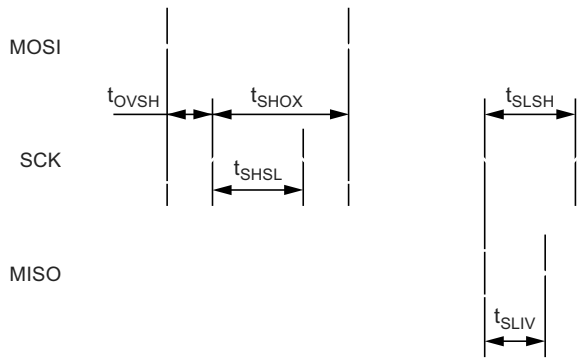


Table 23-8. Serial Programming Characteristics, T_A = –40°C to +125°C, V_{CC} = 2.7 - 5.5V (Unless Otherwise Noted)

Parameter	Symbol	Min	Typ	Max	Unit
Oscillator frequency (ATtiny261/461/861V)	$1/t_{CLCL}$	0		4	MHz
Oscillator period (ATtiny261/461/861V)	t_{CLCL}	250			ns
Oscillator frequency (ATtiny261/461/861L, VCC = 2.7 - 5.5V)	$1/t_{CLCL}$	0		10	MHz
Oscillator period (ATtiny261/461/861L, VCC = 2.7 - 5.5V)	t_{CLCL}	100			ns
Oscillator frequency (ATtiny261/461/861, V_{CC} = 4.5V - 5.5V)	$1/t_{CLCL}$	0		16	MHz
Oscillator period (ATtiny261/461/861, V_{CC} = 4.5V - 5.5V)	t_{CLCL}	50			ns
SCK pulse width high	t_{SHSL}	2 t_{CLCL}*			ns
SCK pulse width low	t_{SLSH}	2 t_{CLCL}*			ns
MOSI setup to SCK high	t_{OVSH}	t_{CLCL}			ns
MOSI hold after SCK high	t_{SHOX}	2 t_{CLCL}			ns
SCK low to MISO valid	t_{SLIV}	TBD	TBD	TBD	ns

Note: 1. 2 t_{CLCL} for f_{ck} < 12MHz, 3 t_{CLCL} for f_{ck} ≥ 12MHz