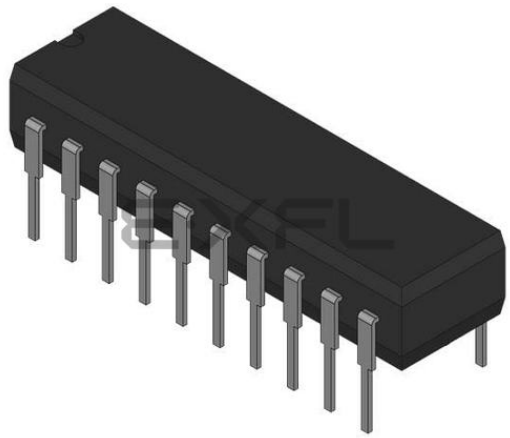


Welcome to E-XFL.COM

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	Obsolete
Core Processor	AVR
Core Size	8-Bit
Speed	20MHz
Connectivity	USI
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	16
Program Memory Size	2KB (1K x 16)
Program Memory Type	FLASH
EEPROM Size	128 x 8
RAM Size	128 x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 5.5V
Data Converters	A/D 11x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Through Hole
Package / Case	20-DIP (0.300", 7.62mm)
Supplier Device Package	20-PDIP
Purchase URL	https://www.e-xfl.com/product-detail/atmel/attiny261-20pu

The Status Register is neither automatically stored when entering an interrupt routine, nor restored when returning from an interrupt. This must be handled by software.

4.3.1 SREG – AVR Status Register

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’s 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.

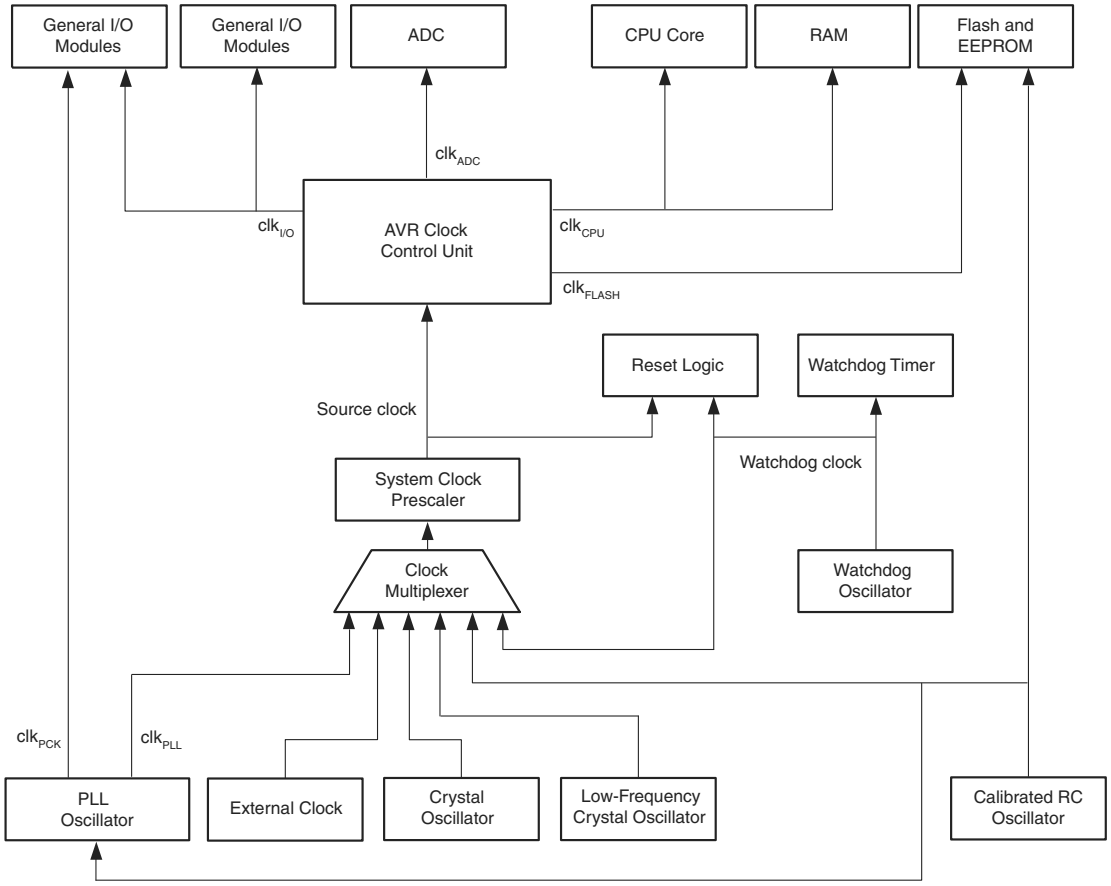
- **Bit 0 – C: Carry Flag**

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

6. Clock System

Figure 6-1 presents the principal clock systems and their distribution in ATtiny261/461/861. All of the clocks need not be active at a given time. In order to reduce power consumption, the clocks to modules not being used can be halted by using different sleep modes, as described in “Power Management and Sleep Modes” on page 36.

Figure 6-1. Clock Distribution



6.1 Clock Subsystems

The clock subsystems are detailed in the sections below.

6.1.1 CPU Clock – clk_{CPU}

The CPU clock is routed to parts of the system concerned with operation of the AVR core. Examples of such modules are the General Purpose Register File, the Status Register and the Data memory holding the Stack Pointer. Halting the CPU clock inhibits the core from performing general operations and calculations.

6.1.2 I/O Clock – $clk_{I/O}$

The I/O clock is used by the majority of the I/O modules, like Timer/Counter. The I/O clock is also used by the External Interrupt module, but note that some external interrupts are detected by asynchronous logic, allowing such interrupts to be detected even if the I/O clock is halted.

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.

7.1.2 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 mode 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.

7.1.3 Power-Down Mode

When the SM1:0 bits are written to 10, the SLEEP instruction makes the MCU enter Power-down mode. In this mode, the Oscillator is stopped, while the external interrupts, and the Watchdog continue operating (if enabled). Only an External Reset, a Watchdog Reset, a Brown-out Reset, an external level interrupt on INT0, or a pin change interrupt can wake up the MCU. This sleep mode halts all generated clocks, allowing operation of asynchronous modules, only.

7.1.4 Standby Mode

When the SM1:0 bits are written to 11 and an external crystal/resonator clock option is selected, the SLEEP instruction makes the MCU enter Standby mode. This mode is identical to Power-down with the exception that the Oscillator is kept running. From Standby mode, the device wakes up in six clock cycles.

7.2 Power Reduction Register

The Power Reduction Register (PRR), see “PRR – Power Reduction Register” on page 39, provides a method to stop the clock to individual peripherals to reduce power consumption. The current state of the peripheral is frozen and the I/O registers can not be read or written. Resources used by the peripheral when stopping the clock will remain occupied, hence the peripheral should in most cases be disabled before stopping the clock. Waking up a module, which is done by clearing the bit in PRR, puts the module in the same state as before shutdown.

Module shutdown can be used in Idle mode and Active mode to significantly reduce the overall power consumption. In all other sleep modes, the clock is already stopped. See “Supply Current of I/O modules” on page 197 for examples. In all other sleep modes, the clock is already stopped.

7.3 Minimizing Power Consumption

There are several issues to consider when trying to minimize the power consumption in an AVR controlled system. In general, sleep modes should be used as much as possible, and the sleep mode should be selected so that as few as possible of the device’s functions are operating. All functions not needed should be disabled. In particular, the following modules may need special consideration when trying to achieve the lowest possible power consumption.

8.4 Register Description

8.4.1 MCUSR – MCU Status Register

The MCU Status Register provides information on which reset source caused an MCU Reset.

Bit	7	6	5	4	3	2	1	0	
0x34 (0x54)	–	–	–	–	WDRF	BORF	EXTRF	PORF	MCUSR
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	See Bit Description				

- **Bits 7:4 – Res: Reserved Bits**

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

- **Bit 3 – WDRF: Watchdog Reset Flag**

This bit is set if a Watchdog Reset occurs. The bit is reset by a Power-on Reset, or by writing a logic zero to the flag.

- **Bit 2 – BORF: Brown-out Reset Flag**

This bit is set if a Brown-out Reset occurs. The bit is reset by a Power-on Reset, or by writing a logic zero to the flag.

- **Bit 1 – EXTRF: External Reset Flag**

This bit is set if an External Reset occurs. The bit is reset by a Power-on Reset, or by writing a logic zero to the flag.

- **Bit 0 – PORF: Power-on Reset Flag**

This bit is set if a Power-on Reset occurs. The bit is reset only by writing a logic zero to the flag.

To make use of the Reset Flags to identify a reset condition, the user should read and then reset the MCUSR as early as possible in the program. If the register is cleared before another reset occurs, the source of the reset can be found by examining the Reset Flags.

8.4.2 WDTCR – Watchdog Timer Control Register

Bit	7	6	5	4	3	2	1	0	
0x21 (0x41)	WDIF	WDIE	WDP3	WDCE	WDE	WDP2	WDP1	WDP0	WDTCR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	X	0	0	0	

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

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

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

When this bit is written to one, WDE is cleared, and the I-bit in the Status Register is set, the Watchdog Time-out Interrupt is enabled. In this mode the corresponding interrupt is executed instead of a reset if a timeout in the Watchdog Timer occurs.

If WDE is set, WDIE is automatically cleared by hardware when a time-out occurs. This is useful for keeping the Watchdog Reset security while using the interrupt. After the WDIE bit is cleared,

9.3.4 PCMSK0 – Pin Change Mask Register A

Bit	7	6	5	4	3	2	1	0	
0x23 (0x43)	PCINT7	PCINT6	PCINT5	PCINT4	PCINT3	PCINT2	PCINT1	PCINT0	PCMSK0
Read/Write	R/W	R/W	R/W	R/w	R/W	R/W	R/W	R/W	
Initial Value	1	1	0	0	1	0	0	0	

- **Bits 7:0 – PCINT7:0: Pin Change Enable Mask 7:0**

Each PCINT7:0 bit selects whether pin change interrupt is enabled on the corresponding I/O pin. If PCINT7:0 is set and the PCIE1 bit in GIMSK is set, pin change interrupt is enabled on the corresponding I/O pin. If PCINT7:0 is cleared, pin change interrupt on the corresponding I/O pin is disabled.

9.3.5 PCMSK1 – Pin Change Mask Register B

Bit	7	6	5	4	3	2	1	0	
0x22 (0x42)	PCINT15	PCINT14	PCINT13	PCINT12	PCINT11	PCINT10	PCINT9	PCINT8	PCMSK1
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	

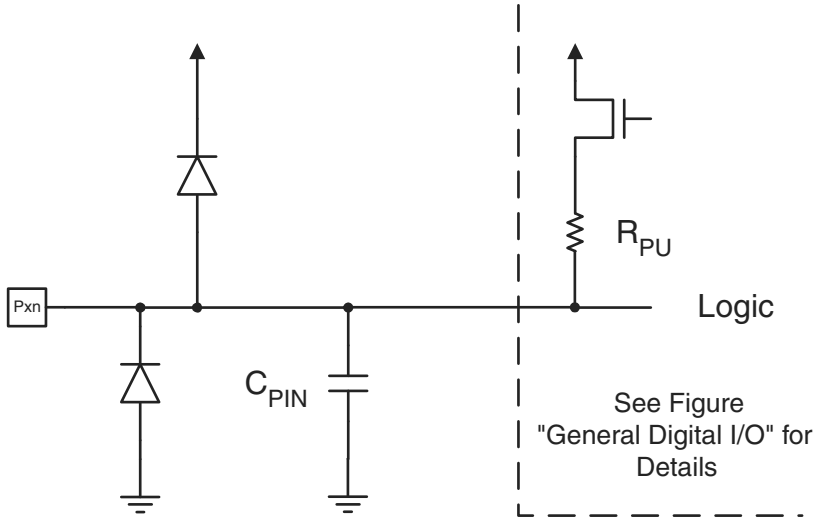
- **Bits 7:0 – PCINT15:8: Pin Change Enable Mask 15:8**

Each PCINT15:8 bit selects whether pin change interrupt is enabled on the corresponding I/O pin. If PCINT11:8 is set and the PCIE0 bit in GIMSK is set, pin change interrupt is enabled on the corresponding I/O pin, and if PCINT15:12 is set and the PCIE1 bit in GIMSK is set, pin change interrupt is enabled on the corresponding I/O pin. If PCINT15:8 is cleared, pin change interrupt on the corresponding I/O pin is disabled.

10. I/O Ports

All AVR ports have true Read-Modify-Write functionality when used as general digital I/O ports. This means that the direction of one port pin can be changed without unintentionally changing the direction of any other pin with the SBI and CBI instructions. The same applies when changing drive value (if configured as output) or enabling/disabling of pull-up resistors (if configured as input). Each output buffer has symmetrical drive characteristics with both high sink and source capability. The pin driver is strong enough to drive LED displays directly. All port pins have individually selectable pull-up resistors with a supply-voltage invariant resistance. All I/O pins have protection diodes to both V_{CC} and Ground as indicated in Figure 10-1. See “Electrical Characteristics” on page 187 for a complete list of parameters.

Figure 10-1. I/O Pin Equivalent Schematic



All registers and bit references in this section are written in general form. A lower case “x” represents the numbering letter for the port, and a lower case “n” represents the bit number. However, when using the register or bit defines in a program, the precise form must be used. For example, PORTB3 for bit no. 3 in Port B, here documented generally as PORTxn. The physical I/O Registers and bit locations are listed in “Register Description” on page 69.

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

Using the I/O port as General Digital I/O is described in “Ports as General Digital I/O” on page 56. Most port pins are multiplexed with alternate functions for the peripheral features on the device. How each alternate function interferes with the port pin is described in “Alternate Port Functions” on page 60. Refer to the individual module sections for a full description of the alternate functions.

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

10.1.5 Digital Input Enable and Sleep Modes

As shown in Figure 10-2, the digital input signal can be clamped to ground at the input of the schmitt-trigger. The signal denoted SLEEP in the figure, is set by the MCU Sleep Controller in Power-down mode, Power-save mode, and Standby mode to avoid high power consumption if some input signals are left floating, or have an analog signal level close to $V_{CC}/2$.

SLEEP is overridden for port pins enabled as external interrupt pins. If the external interrupt request is not enabled, SLEEP is active also for these pins. SLEEP is also overridden by various other alternate functions as described in “Alternate Port Functions” on page 60.

If a logic high level (“one”) is present on an asynchronous external interrupt pin configured as “Interrupt on Rising Edge, Falling Edge, or Any Logic Change on Pin” while the external interrupt is *not* enabled, the corresponding External Interrupt Flag will be set when resuming from the above mentioned Sleep mode, as the clamping in these sleep mode produces the requested logic change.

10.1.6 Unconnected Pins

If some pins are unused, it is recommended to ensure that these pins have a defined level. Even though most of the digital inputs are disabled in the deep sleep modes as described above, floating inputs should be avoided to reduce current consumption in all other modes where the digital inputs are enabled (Reset, Active mode and Idle mode).

The simplest method to ensure a defined level of an unused pin, is to enable the internal pull-up. In this case, the pull-up will be disabled during reset. If low power consumption during reset is important, it is recommended to use an external pull-up or pulldown. Connecting unused pins directly to V_{CC} or GND is not recommended, since this may cause excessive currents if the pin is accidentally configured as an output.

10.1.7 Program Examples

The following code examples show how to set port B pins 0 and 1 high, 2 and 3 low, and define the port pins from 4 to 5 as input with a pull-up assigned to port pin 4. The resulting pin values are read back again, but as previously discussed, a *nop* instruction is included to be able to read back the value recently assigned to some of the pins.

Assembly Code Example

```
...
; Define pull-ups and set outputs high
; Define directions for port pins
ldi  r16,(1<<PB4)|(1<<PB1)|(1<<PB0)
ldi  r17,(1<<DDB3)|(1<<DDB2)|(1<<DDB1)|(1<<DDB0)
out  PORTB,r16
out  DDRB,r17
; Insert nop for synchronization
nop
; Read port pins
in   r16,PINB
...
```

Note: Two temporary registers are used to minimize the time from pull-ups are set on pins 0, 1 and 4, until the direction bits are correctly set, defining bit 2 and 3 as low and redefining bits 0 and 1 as strong high drivers.

Table 10-8. Overriding Signals for Alternate Functions in PB3:PB0

Signal Name	PB3/OC1B/ PCINT11	PB2/SCK/USCK/SCL/OC1B/PCINT10	PB1/MISO/DO/OC1A/ PCINT9	PB0/MOSI/DI/SDA/ OC1A/PCINT8
PUOE	0	0	0	0
PUOV	0	0	0	0
DDOE	0	USI_TWO_WIRE • USIPOS	0	USI_TWO_WIRE • USIPOS
DDOV	0	(USI_SCL_HOLD + PORTB2) • DDB2 • USIPOS	0	(SDA + PORTB0) • DDB0 • USIPOS
PVOE	OC1B Enable	OC1B Enable + USIPOS • USI_TWO_WIRE • DDB2	OC1A Enable + USIPOS • USI_THREE_WIRE	OC1A Enable + (USI_TWO_WIRE • DDB0 • USIPOS)
PVOV	OC1B	OC1B	OC1A + (DO • USIPOS)	OC1A
PTOE	0	USITC • USIPOS	0	0
DIEOE	PCINT11 • PCIE	PCINT10 • PCIE + USISIE • USIPOS	PCINT9 • PCIE	PCINT8 • PCIE + (USISIE • USIPOS)
DIEOV	0	0	0	0
DI	PCINT11	USCK/SCL/PCINT10	PCINT9	DI/SDA/PCINT8
AIO				

Note: 1. INTRC means that one of the internal RC Oscillators is selected (by the CKSEL fuses), EXTCK means that external clock is selected (by the CKSEL fuses).

10.3 Register Description

10.3.1 MCUCR – MCU Control Register

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	R	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 6 – PUD: Pull-up Disable**

When this bit is written to one, the pull-ups in the I/O ports are disabled even if the DDxn and PORTxn Registers are configured to enable the pull-ups ({DDxn, PORTxn} = 0b01). See “Configuring the Pin” on page 56 for more details about this feature.

10.3.2 PORTA – Port A Data Register

Bit	7	6	5	4	3	2	1	0	
0x1B (0x3B)	PORTA7	PORTA6	PORTA5	PORTA4	PORTA3	PORTA2	PORTA1	PORTA0	PORTA
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	

10.3.3 DDRA – Port A Data Direction Register

Bit	7	6	5	4	3	2	1	0	
0x1A (0x3A)	DDA7	DDA6	DDA5	DDA4	DDA3	DDA2	DDA1	DDA0	DDRA
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 following code examples show how to do an atomic write of the TCNT0H/L register contents. Writing any of the OCR0A/B registers can be done by using the same principle.

Assembly Code Example

```
TIM0_WriteTCNT0:
  ; Save global interrupt flag
  in  r18,SREG
  ; Disable interrupts
  cli
  ; Set TCNT0 to r17:r16
  out TCNT0H,r17
  out TCNT0L,r16
  ; Restore global interrupt flag
  out SREG,r18
  ret
```

C Code Example

```
void TIM0_WriteTCNT0( unsigned int i )
{
  unsigned char sreg;
  /* Save global interrupt flag */
  sreg = SREG;
  /* Disable interrupts */
  _CLI();
  /* Set TCNT0 to i */
  TCNT0H = (i >> 8);
  TCNT0L = (unsigned char)i;
  /* Restore global interrupt flag */
  SREG = sreg;
}
```

Note: See “Code Examples” on page 6.

The assembly code example requires that the r17:r16 register pair contains the value to be written to TCNT0H/L.

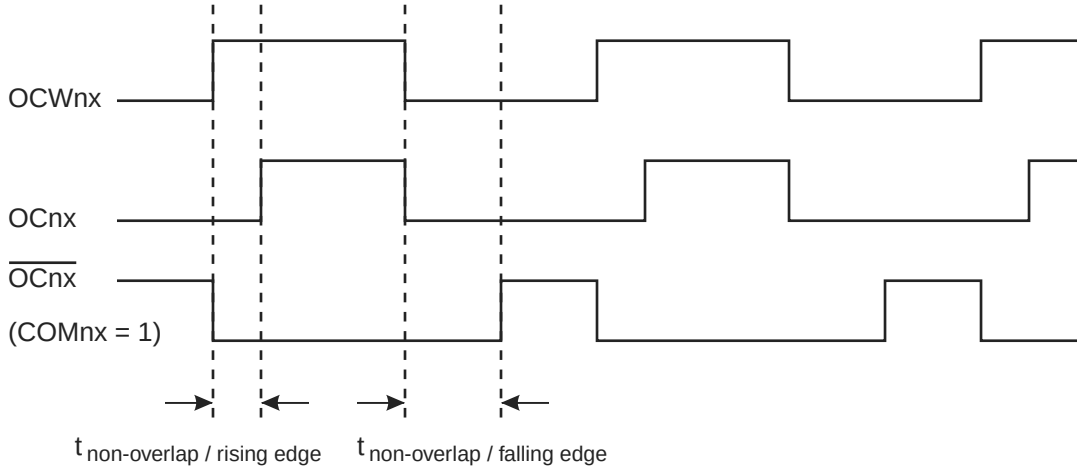
11.9.1 Reusing the temporary high byte register

If writing to more than one 16-bit register where the high byte is the same for all registers written, then the high byte only needs to be written once. However, note that the same rule of atomic operation described previously also applies in this case.

zero. The outputs OC1x and $\overline{OC1x}$ are inverted, if the PWM Inversion Mode bit PWM1X is set. This will also cause both outputs to be high during the dead time.

The length of the counting period is user adjustable by selecting the dead time prescaler setting by using the DTPS11:10 control bits, and selecting then the dead time value in I/O register DT1. The DT1 register consists of two 4-bit fields, DT1H and DT1L that control the dead time periods of the PWM output and its' complementary output separately in terms of the number of prescaled dead time generator clock cycles. Thus the rising edge of OC1x and $\overline{OC1x}$ can have different dead time periods as the $t_{non\text{-}overlap\ /\ rising\ edge}$ is adjusted by the 4-bit DT1H value and the $t_{non\text{-}overlap\ /\ falling\ edge}$ is adjusted by the 4-bit DT1L value.

Figure 12-9. The Complementary Output Pair, COM1x1:0 = 1



12.7 Compare Match Output Unit

The Compare Output Mode (COM1x1:0) bits have two functions. The Waveform Generator uses the COM1x1:0 bits for defining the inverted or non-inverted Waveform Output (OCW1x) at the next Compare Match. Also, the COM1x1:0 bits control the OC1x and $\overline{OC1x}$ pin output source. Figure 12-10 on page 98 shows a simplified schematic of the logic affected by the COM1x1:0 bit setting. The I/O Registers, I/O bits, and I/O pins in the figure are shown in bold. Only the parts of the general I/O Port Control Registers (DDR and PORT) that are affected by the COM1x1:0 bits are shown.

In Normal Mode (non-PWM) the Dead Time Generator is disabled and it is working like a synchronizer: the Output Compare (OC1x) is delayed from the Waveform Output (OCW1x) by one timer clock cycle. Whereas in Fast PWM Mode and in Phase and Frequency Correct PWM Mode when the COM1x1:0 bits are set to “01” both the non-inverted and the inverted Output Compare output are generated, and an user programmable Dead Time delay is inserted for these complementary output pairs (OC1x and $\overline{OC1x}$). The functionality in PWM modes is similar to Normal mode when any other COM1x1:0 bit setup is used. When referring to the OC1x state, the reference is for the Output Compare output (OC1x) from the Dead Time Generator, not the OC1x pin. If a system reset occur, the OC1x is reset to “0”.

The design of the Output Compare Pin Configuration logic allows initialization of the OC1x state before the output is enabled. Note that some COM1x1:0 bit settings are reserved for certain modes of operation. For Output Compare Pin Configurations refer to Table 12-2 on page 100, Table 12-3 on page 102, Table 12-4 on page 104, Table 12-5 on page 105, Table 12-6 on page 105, and Table 12-7 on page 106.

12.7.1 Compare Output Mode and Waveform Generation

The Waveform Generator uses the COM1x1:0 bits differently in Normal mode and PWM modes. For all modes, setting the COM1x1:0 = 0 tells the Waveform Generator that no action on the OCW1x Output is to be performed on the next Compare Match. For compare output actions in the non-PWM modes refer to Table 12-8 on page 112. For fast PWM mode, refer to Table 12-9 on page 112, and for the Phase and Frequency Correct PWM refer to Table 12-10 on page 113. A change of the COM1x1:0 bits state will have effect at the first Compare Match after the bits are written. For non-PWM modes, the action can be forced to have immediate effect by using the FOC1x strobe bits.

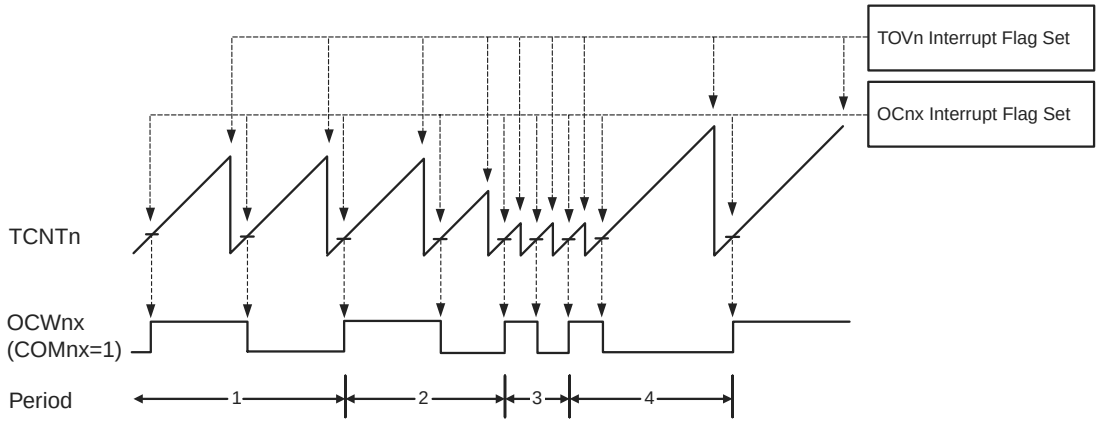
12.8 Modes of Operation

The mode of operation, i.e., the behavior of the Timer/Counter and the Output Compare pins, is defined by the combination of waveform generation mode bits (PWM1A, PWM1B, and WGM11:10) and compare output mode bits (COM1x1:0). The Compare Output mode bits do not affect the counting sequence, while the Waveform Generation mode bits do. The COM1x1:0 bits control whether the PWM output generated should be inverted, non-inverted or complementary. For non-PWM modes the COM1x1:0 bits control whether the output should be set, cleared, or toggled at a Compare Match.

12.8.1 Normal Mode

The simplest mode of operation is Normal mode (PWM1A/PWM1B = 0), where the counter counts from BOTTOM to TOP (defined as OCR1C) then restarts from BOTTOM. The OCR1C defines the TOP value for the counter, hence also its resolution, and allows control of the Compare Match output frequency. In toggle Compare Output Mode the Waveform Output (OCW1x) is toggled at Compare Match between TCNT1 and OCR1x. In non-inverting Compare Output Mode the Waveform Output is cleared on the Compare Match. In inverting Compare Output Mode the Waveform Output is set on Compare Match. The timing diagram for Normal mode is shown in Figure 12-11.

Figure 12-11. Normal Mode, Timing Diagram



12.12.3 TCCR1C – Timer/Counter1 Control Register C

Bit	7	6	5	4	3	2	1	0	
0x27 (0x47)	COM1A1S	COM1A0S	COM1B1S	COM1B0S	COM1D1	COM1D0	FOC1D	PWM1D	TCCR1C
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	

- **Bits 7,6 – COM1A1S, COM1A0S: Comparator A Output Mode, Shadow Bits 1 and 0**

These are shadow bits of COM1A1 and COM1A0 in TCCR1A. Writing to bits COM1A1S and COM1A0S will also change bits COM1A1 and COM1A0 in TCCR1A. Similary, changes written to bits COM1A1 and COM1A0 in TCCR1A will show here.

See “TCCR1A – Timer/Counter1 Control Register A” on page 112 for information on bit usage.

- **Bits 5,4 – COM1B1S, COM1B0S: Comparator B Output Mode, Shadow Bits 1 and 0**

These are shadow bits of COM1B1 and COM1B0 in TCCR1A. Writing to bits COM1B1S and COM1B0S will also change bits COM1B1 and COM1B0 in TCCR1A. Similary, changes written to bits COM1B1 and COM1B0 in TCCR1A will show here.

See “TCCR1A – Timer/Counter1 Control Register A” on page 112 for information on bit usage.

- **Bits 3,2 – COM1D1, COM1D0: Comparator D Output Mode, Bits 1 and 0**

These bits control the behaviour of the Waveform Output (OCW1D) and the connection of the Output Compare pin (OC1D). If one or both of the COM1D1:0 bits are set, the OC1D output overrides the normal port functionality of the I/O pin it is connected to. The complementary $\overline{\text{OC1D}}$ output is connected only in PWM modes when the COM1D1:0 bits are set to “01”. Note that the Data Direction Register (DDR) bit corresponding to the OC1D pin must be set in order to enable the output driver.

The function of the COM1D1:0 bits depends on the PWM1D and WGM11:10 bit settings. Table 12-18 shows the COM1D1:0 bit functionality when the PWM1D bit is set to a Normal Mode (non-PWM).

Table 12-18. Compare Output Mode, Normal Mode (non-PWM)

COM1D1:0	OCW1D Behaviour	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 12-19 shows the COM1D1:0 bit functionality when the PWM1D and WGM11:10 bits are set to Fast PWM Mode.

Table 12-19. Compare Output Mode, Fast PWM Mode

COM1D1:0	OCW1D Behaviour	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. Cleared when TCNT1=0x000.	Connected	Disconnected

13.4 Alternative USI Usage

The flexible design of the USI allows it to be used for other tasks when serial communication is not needed. Below are some examples.

13.4.1 Half-Duplex Asynchronous Data Transfer

Using the USI Data Register in three-wire mode it is possible to implement a more compact and higher performance UART than by software, only.

13.4.2 4-Bit Counter

The 4-bit counter can be used as a stand-alone counter with overflow interrupt. Note that if the counter is clocked externally, both clock edges will increment the counter value.

13.4.3 12-Bit Timer/Counter

Combining the 4-bit USI counter with one of the 8-bit timer/counters creates a 12-bit counter.

13.4.4 Edge Triggered External Interrupt

By setting the counter to maximum value (F) it can function as an additional external interrupt. The Overflow Flag and Interrupt Enable bit are then used for the external interrupt. This feature is selected by the USICS1 bit.

13.4.5 Software Interrupt

The counter overflow interrupt can be used as a software interrupt triggered by a clock strobe.

13.5 Register Descriptions

13.5.1 USIDR – USI Data Register

Bit	7	6	5	4	3	2	1	0	
0x0F (0x2F)	**MSB**							**LSB**	**USIDR**
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 USI Data Register can be accessed directly.

Depending on the USICS1:0 bits of the USI Control Register a (left) shift operation may be performed. The shift operation can be synchronised to an external clock edge, to a Timer/Counter0 Compare Match, or directly to software via the USICLK bit. If a serial clock occurs at the same cycle the register is written, the register will contain the value written and no shift is performed.

Note that even when no wire mode is selected (USIWM1:0 = 0) both the external data input (DI/SDA) and the external clock input (USCK/SCL) can still be used by the USI Data Register.

The output pin (DO or SDA, depending on the wire mode) is connected via the output latch to the most significant bit (bit 7) of the USI Data Register. The output latch ensures that data input is sampled and data output is changed on opposite clock edges. The latch is open (transparent) during the first half of a serial clock cycle when an external clock source is selected (USICS1 = 1) and constantly open when an internal clock source is used (USICS1 = 0). The output will be changed immediately when a new MSB is written as long as the latch is open.

Note that the Data Direction Register bit corresponding to the output pin must be set to one in order to enable data output from the USI Data Register.

reference may be decoupled by an external capacitor at the AREF pin to improve noise immunity.

The analog input channel and differential gain are selected by writing to the MUX5:0 bits in ADMUX. Any of the 11 ADC input pins ADC10:0 can be selected as single ended inputs to the ADC. The positive and negative inputs to the differential gain amplifier are described in Table 15-4.

If differential channels are selected, the differential gain stage amplifies the voltage difference between the selected input pair by the selected gain factor, 1x, 8x, 20x or 32x, according to the setting of the MUX5:0 bits in ADMUX and the GSEL bit in ADCSRB. This amplified value then becomes the analog input to the ADC. If single ended channels are used, the gain amplifier is bypassed altogether.

If the same ADC input pin is selected as both the positive and negative input to the differential gain amplifier, the remaining offset in the gain stage and conversion circuitry can be measured directly as the result of the conversion. This figure can be subtracted from subsequent conversions with the same gain setting to reduce offset error to below 1 LSW.

The on-chip temperature sensor is selected by writing the code “111111” to the MUX5:0 bits in ADMUX register when the ADC11 channel is used as an ADC input.

The ADC is enabled by setting the ADC Enable bit, ADEN in ADCSRA. Voltage reference and input channel selections will not go into effect until ADEN is set. The ADC does not consume power when ADEN is cleared, so it is recommended to switch off the ADC before entering power saving sleep modes.

The ADC generates a 10-bit result which is presented in the ADC Data Registers, ADCH and ADCL. By default, the result is presented right adjusted, but can optionally be presented left adjusted by setting the ADLAR bit in ADMUX.

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, to ensure that the content of the data registers belongs to the same conversion. Once ADCL is read, ADC access to data registers is blocked. This means that if ADCL has been read, and a conversion completes before ADCH is read, neither register is updated and the result from the conversion is lost. When ADCH is read, ADC access to the ADCH and ADCL Registers is re-enabled.

The ADC has its own interrupt which can be triggered when a conversion completes. When ADC access to the data registers is prohibited between reading of ADCH and ADCL, the interrupt will trigger even if the result is lost.

15.4 Starting a Conversion

A single conversion is started by writing a logical one to the ADC Start Conversion bit, ADSC. This bit stays high as long as the conversion is in progress and will be cleared by hardware when the conversion is completed. If a different data channel is selected while a conversion is in progress, the ADC will finish the current conversion before performing the channel change.

Alternatively, a conversion can be triggered automatically by various sources. Auto Triggering is enabled by setting the ADC Auto Trigger Enable bit, ADATE in ADCSRA. The trigger source is selected by setting the ADC Trigger Select bits, ADTS in ADCSRB (see description of the ADTS bits for a list of the trigger sources). When a positive edge occurs on the selected trigger signal, the ADC prescaler is reset and a conversion is started. This provides a method of starting conversions at fixed intervals. If the trigger signal still is set when the conversion completes, a new

17. Self-Programming the Flash

The device provides a Self-Programming mechanism for downloading and uploading program code by the MCU itself. The Self-Programming can use any available data interface and associated protocol to read code and write (program) that code into the Program memory. The SPM instruction is disabled by default but it can be enabled by programming the SELFPRGEN fuse (to “0”).

The Program memory is updated in a page by page fashion. Before programming a page with the data stored in the temporary page buffer, the page must be erased. The temporary page buffer is filled one word at a time using SPM and the buffer can be filled either before the Page Erase command or between a Page Erase and a Page Write operation:

Alternative 1, fill the buffer before a Page Erase

- Fill temporary page buffer
- Perform a Page Erase
- Perform a Page Write

Alternative 2, fill the buffer after Page Erase

- Perform a Page Erase
- Fill temporary page buffer
- Perform a Page Write

If only a part of the page needs to be changed, the rest of the page must be stored (for example in the temporary page buffer) before the erase, and then be re-written. When using alternative 1, the Boot Loader provides an effective Read-Modify-Write feature which allows the user software to first read the page, do the necessary changes, and then write back the modified data. If alternative 2 is used, it is not possible to read the old data while loading since the page is already erased. The temporary page buffer can be accessed in a random sequence. It is essential that the page address used in both the Page Erase and Page Write operation is addressing the same page.

17.1 Performing Page Erase by SPM

To execute Page Erase, set up the address in the Z-pointer, write “00000011” to SPMCSR and execute SPM within four clock cycles after writing SPMCSR. The data in R1 and R0 is ignored. The page address must be written to PCPAGE in the Z-register. Other bits in the Z-pointer will be ignored during this operation.

Note: The CPU is halted during the Page Erase operation.

17.2 Filling the Temporary Buffer (Page Loading)

To write an instruction word, set up the address in the Z-pointer and data in R1:R0, write “00000001” to SPMCSR and execute SPM within four clock cycles after writing SPMCSR. The content of PCWORD in the Z-register is used to address the data in the temporary buffer. The temporary buffer will auto-erase after a Page Write operation or by writing the CTPB bit in SPMCSR. It is also erased after a system reset. Note that it is not possible to write more than one time to each address without erasing the temporary buffer.

18.2 Fuse Bytes

The ATtiny261/461/861 have three fuse bytes. Table 18-3, Table 18-4 and Table 18-5 describe briefly the functionality of all the fuses and how they are mapped into the fuse bytes. Note that the fuses are read as logical zero, “0”, if they are programmed.

Table 18-3. Fuse Extended Byte

Fuse High Byte	Bit No	Description	Default Value
	7	-	1 (unprogrammed)
	6	-	1 (unprogrammed)
	5	-	1 (unprogrammed)
	4	-	1 (unprogrammed)
	3	-	1 (unprogrammed)
	2	-	1 (unprogrammed)
	1	-	1 (unprogrammed)
SELFPRGEN (1)	0	Self-Programming Enable	1 (unprogrammed)

Notes: 1. Enables SPM instruction. See “Self-Programming the Flash” on page 165.

Table 18-4. Fuse High Byte

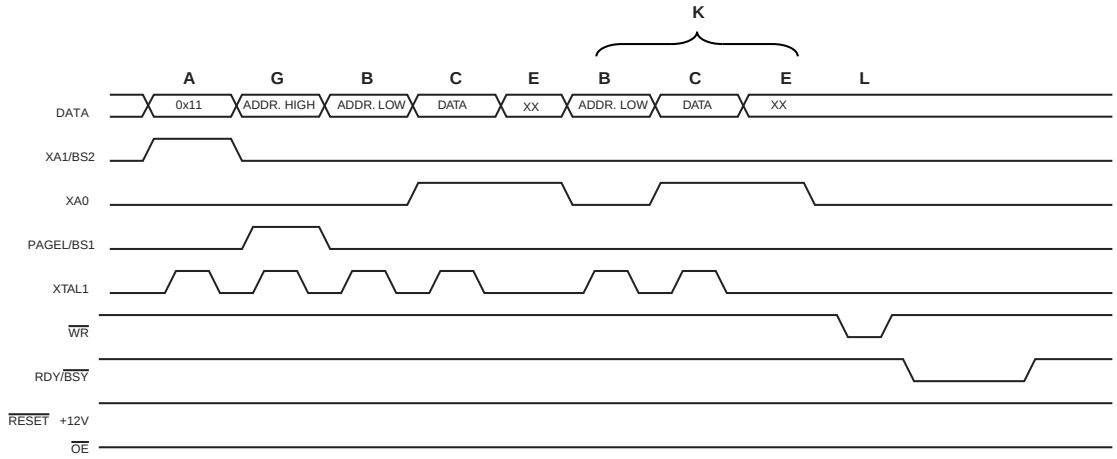
Fuse High Byte	Bit No	Description	Default Value
RSTDISBL (1)	7	External Reset disable	1 (unprogrammed)
DWEN (2)	6	DebugWIRE Enable	1 (unprogrammed)
SPIEN (3))	6	Enable Serial Program and Data Downloading	0 (programmed, SPI prog. enabled)
WDTON (4)	4	Watchdog Timer always on	1 (unprogrammed)
EESAVE	3	EEPROM memory is preserved through the Chip Erase	1 (unprogrammed, EEPROM not preserved)
BODLEVEL2 (5)	2	Brown-out Detector trigger level	1 (unprogrammed)
BODLEVEL1 (5)	1	Brown-out Detector trigger level	1 (unprogrammed)
BODLEVEL0 (5)	0	Brown-out Detector trigger level	1 (unprogrammed)

Notes:
1. See “Alternate Functions of Port B” on page 66 for description of RSTDISBL and DWEN Fuses. After programming the RSTDISBL fuse, parallel programming must be used to change fuses and allow further programming.
2. DWEN must be unprogrammed when Lock Bit security is required. See “Program And Data Memory Lock Bits” on page 170.
3. The SPIEN Fuse is not accessible in SPI programming mode.
4. Programming this fues will disable the Watchdog Timer Interrupt. See “WDTCR – Watchdog Timer Control Register” on page 47 for details.
5. See Table 19-6 on page 191 for BODLEVEL Fuse decoding.

programmed simultaneously. The programming algorithm for the EEPROM data memory is as follows (refer to “Programming the Flash” on page 180 for details on Command, Address and Data loading):

1. A: Load Command “0001 0001”.
2. G: Load Address High Byte (0x00 - 0xFF).
3. B: Load Address Low Byte (0x00 - 0xFF).
4. C: Load Data (0x00 - 0xFF).
5. E: Latch data (give PAGEL a positive pulse).
6. K: Repeat 3 through 5 until the entire buffer is filled.
7. L: Program EEPROM page
   a. Set BS to “0”.
   b. Give $\overline{\text{WR}}$ a negative pulse. This starts programming of the EEPROM page. RDY/$\overline{\text{BSY}}$ goes low.
   c. Wait until to RDY/$\overline{\text{BSY}}$ goes high before programming the next page (See Figure 18-6 for signal waveforms).

Figure 18-6. Programming the EEPROM Waveforms



18.7.7 Reading the Flash

The algorithm for reading the Flash memory is as follows (refer to “Programming the Flash” on page 180 for details on Command and Address loading):

1. A: Load Command “0000 0010”.
2. G: Load Address High Byte (0x00 - 0xFF).
3. B: Load Address Low Byte (0x00 - 0xFF).
4. Set $\overline{\text{OE}}$ to “0”, and BS1 to “0”. The Flash word low byte can now be read at DATA.
5. Set BS to “1”. The Flash word high byte can now be read at DATA.
6. Set $\overline{\text{OE}}$ to “1”.

19.8 Parallel Programming Characteristics

Figure 19-6. Parallel Programming Timing, Including some General Timing Requirements

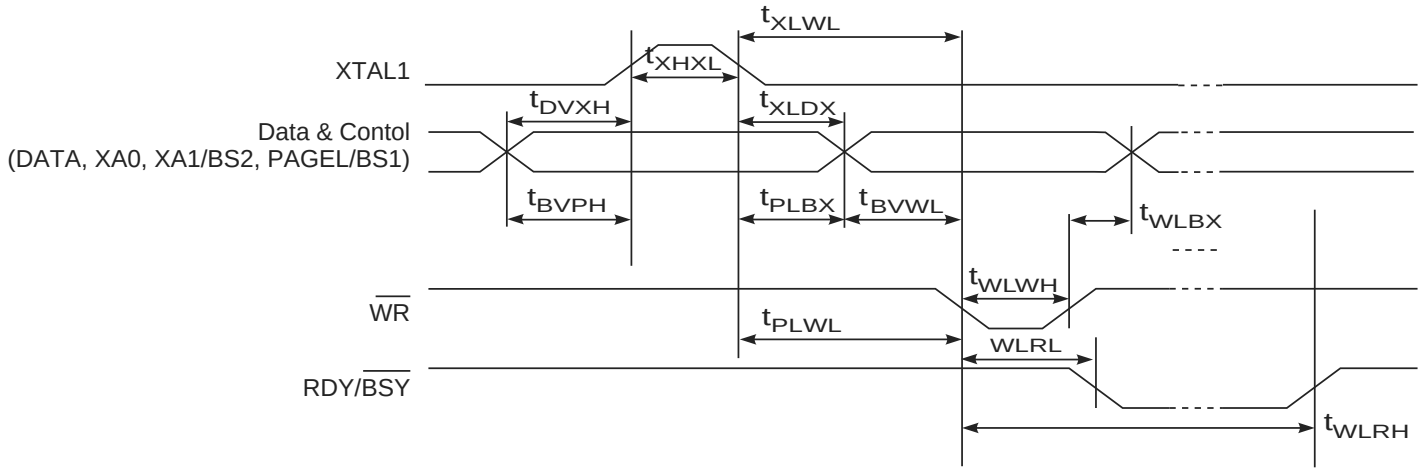
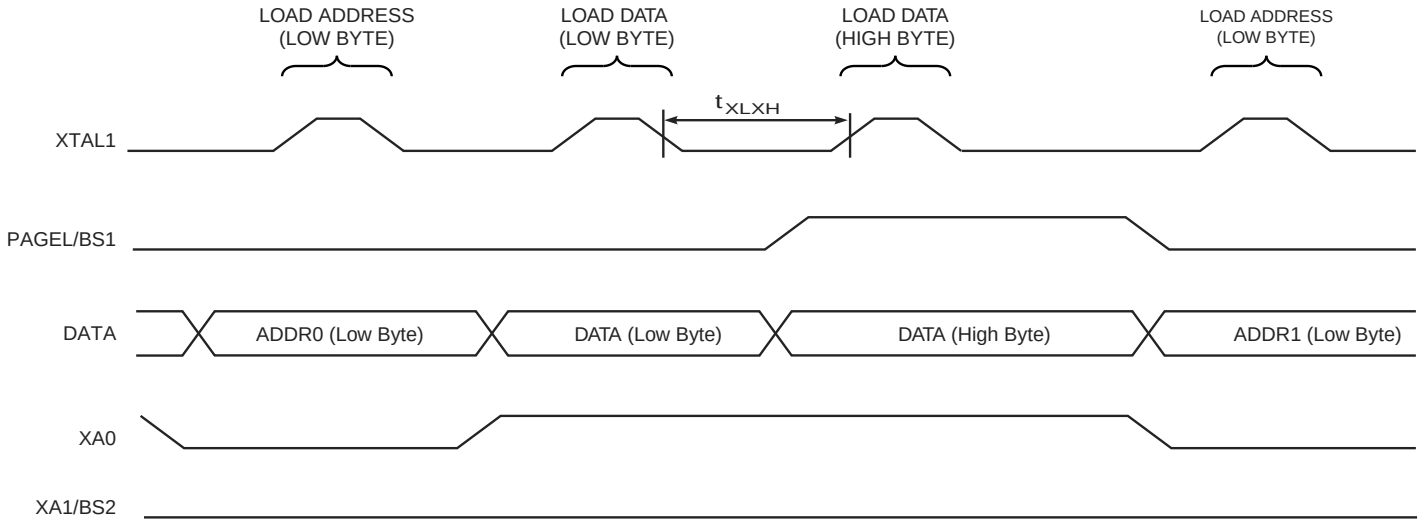


Figure 19-7. Parallel Programming Timing, Loading Sequence with Timing Requirements



Note: The timing requirements shown in Figure 19-6 (i.e., t_{DVXH}, t_{XHXL}, and t_{XLDX}) also apply to loading operation.

23.3 ATtiny861

Speed (MHz)[3]	Power Supply (V)	Ordering Code[4]	Package[2]	Operational Range
10	1.8 - 5.5	ATtiny861V-10MU ATtiny861V-10MUR ATtiny861V-10PU ATtiny861V-10SU ATtiny861V-10SUR	32M1-A 32M1-A 20P3 20S2 20S2	Industrial (-40°C to +85°C)[1]
20	2.7 - 5.5	ATtiny861-20MU ATtiny861-20MUR ATtiny861-20PU ATtiny861-20SU ATtiny861-20SUR	32M1-A 32M1-A 20P3 20S2 20S2	Industrial (-40°C to +85°C)[1]

Notes:
1. These devices can also be supplied in wafer form. Please contact your local Atmel sales office for detailed ordering information and minimum quantities.
2. All packages are Pb-free, halide-free and fully green and they comply with the European directive for Restriction of Hazardous Substances (RoHS).
3. For Speed vs. V_{CC}, see Figure 19.3 on page 188.
4. Code indicators:
   - U: matte tin
   - R: tape & reel

Package Type	
32M1-A	32-pad, 5 x 5 x 1.0 mm Body, Lead Pitch 0.50 mm, Micro Lead Frame Package (MLF)
20P3	20-lead, 0.300" Wide, Plastic Dual Inline Package (PDIP)
20S2	20-lead, 0.300" Wide, Plastic Gull Wing Smal Outline Package (SOIC)

26. Datasheet Revision History

Please note that the referring page numbers in this section refer to the complete document.

26.1 Rev. 2588F – 06/13

1. ATtiny261 changed status to “Mature”.

26.2 Rev. 2588E – 08/10

1. Added tape and reel in “Ordering Information” on page 227.
2. Clarified Section 6.4 “Clock Output Buffer” on page 32.
3. Removed text "Not recommended for new designs" from cover page.

26.3 Rev. 2588D – 06/10

1. Removed “Preliminary” from cover page.
2. Added clarification before Table 6-10, “Capacitance for Low-Frequency Crystal Oscillator,” on page 29.
3. Updated Figure 15-1 “Analog to Digital Converter Block Schematic” on page 143, changed INTERNAL 1.18V REFERENCE to 1.1V.
4. Updated Table 18-8, “No. of Words in a Page and No. of Pages in the EEPROM,” on page 173, No. of Pages from 64 to 32 for ATtiny261.
5. Adjusted notes in Table 19-1, “DC Characteristics. TA = -40°C to +85°C, VCC = 1.8V to 5.5V (unless otherwise noted).,” on page 187.

26.4 Rev. 2588C – 10/09

1. Updated document template. Re-arranged some sections.
2. Changed device status to "Not Recommended for New Designs".
3. Added Sections:
   – “Data Retention” on page 6
   – “Clock Sources” on page 25
   – “Low Level Interrupt” on page 51
   – “Prescaling and Conversion Timing” on page 145
   – “Clock speed considerations” on page 131
4. Updated Sections:
   – “Code Examples” on page 6
   – “High-Frequency PLL Clock” on page 26
   – “Normal Mode” on page 99
   – “Features” on page 142
   – “Temperature Measurement” on page 154
   – “Limitations of debugWIRE” on page 164
   – Step 1. on page 174
   – “Programming the Flash” on page 180
   – “System and Reset Characteristics” on page 190
5. Added Figures:
   – “Flash Programming Waveforms” on page 182