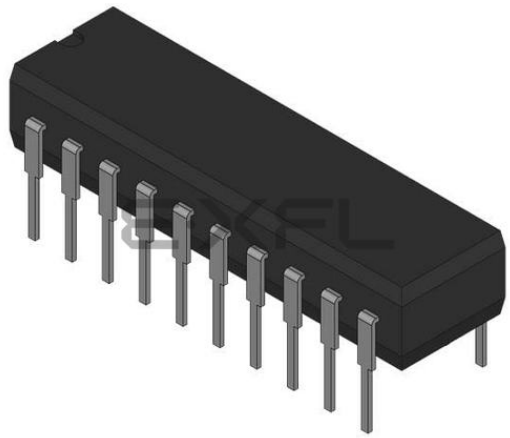


Welcome to E-XFL.COM

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	10MHz
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)	1.8V ~ 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/attiny261v-10pu

to be cleared. If an interrupt condition occurs while the corresponding interrupt enable bit is cleared, the Interrupt Flag will be set and remembered until the interrupt is enabled, or the flag is cleared by software. Similarly, if one or more interrupt conditions occur while the Global Interrupt Enable bit is cleared, the corresponding Interrupt Flag(s) will be set and remembered until the Global Interrupt Enable bit is set, and will then be executed by order of priority.

The second type of interrupts will trigger as long as the interrupt condition is present. These interrupts do not necessarily have Interrupt Flags. If the interrupt condition disappears before the interrupt is enabled, the interrupt will not be triggered.

When the AVR exits from an interrupt, it will always return to the main program and execute one more instruction before any pending interrupt is served.

Note that the Status Register is not automatically stored when entering an interrupt routine, nor restored when returning from an interrupt routine. This must be handled by software.

When using the CLI instruction to disable interrupts, the interrupts will be immediately disabled. No interrupt will be executed after the CLI instruction, even if it occurs simultaneously with the CLI instruction. The following example shows how this can be used to avoid interrupts during the timed EEPROM write sequence.

Assembly Code Example

```
in  r16, SREG     ; store SREG value
cli    ; disable interrupts during timed sequence
sbi EECR, EEMPE   ; start EEPROM write
sbi EECR, EEPE
out SREG, r16     ; restore SREG value (I-bit)
```

C Code Example

```
char cSREG;
cSREG = SREG; /* store SREG value */
/* disable interrupts during timed sequence */
_CLI();
EECR |= (1<<EEMPE); /* start EEPROM write */
EECR |= (1<<EEPE);
SREG = cSREG; /* restore SREG value (I-bit) */
```

Note: See “Code Examples” on page 6.

When using the SEI instruction to enable interrupts, the instruction following SEI will be executed before any pending interrupts, as shown in the following examples.

Assembly Code Example

```
sei       ; set Global Interrupt Enable
sleep     ; enter sleep, waiting for interrupt
          ; note: will enter sleep before any pending interrupt(s)
```

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.

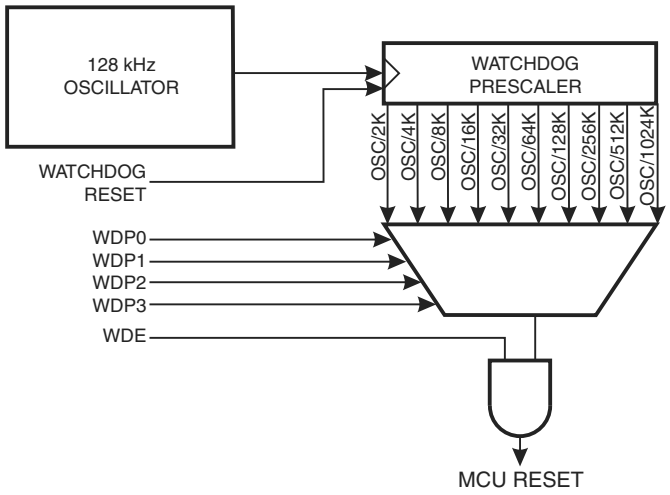
The Wathdog Timer can also be configured to generate an interrupt instead of a reset. This can be very helpful when using the Watchdog to wake-up from Power-down.

To prevent unintentional disabling of the Watchdog or unintentional change of time-out period, two different safety levels are selected by the fuse WDTON as shown in Table 8-1 Refer to “Timed Sequences for Changing the Configuration of the Watchdog Timer” on page 45 for details.

Table 8-1. WDT Configuration as a Function of the Fuse Settings of WDTON

WDTON	Safety Level	WDT Initial State	How to Disable the WDT	How to Change Time-out
Unprogrammed	1	Disabled	Timed sequence	No limitations
Programmed	2	Enabled	Always enabled	Timed sequence

Figure 8-7. Watchdog Timer



8.3.1 Timed Sequences for Changing the Configuration of the Watchdog Timer

The sequence for changing configuration differs slightly between the two safety levels. Separate procedures are described for each level.

8.3.1.1 Safety Level 1

In this mode, the Watchdog Timer is initially disabled, but can be enabled by writing the WDE bit to one without any restriction. A timed sequence is needed when disabling an enabled Watchdog Timer. To disable an enabled Watchdog Timer, the following procedure must be followed:

1. In the same operation, write a logic one to WDCE and WDE. A logic one must be written to WDE regardless of the previous value of the WDE bit.
2. Within the next four clock cycles, in the same operation, write the WDE and WDP bits as desired, but with the WDCE bit cleared.

8.3.1.2 Safety Level 2

In this mode, the Watchdog Timer is always enabled, and the WDE bit will always read as one. A timed sequence is needed when changing the Watchdog Time-out period. To change the Watchdog Time-out, the following procedure must be followed:

- **Bits 5, 2:0 – WDP3:0: Watchdog Timer Prescaler 3, 2, 1, and 0**

The WDP3:0 bits determine the Watchdog Timer prescaling when the Watchdog Timer is enabled. The different prescaling values and their corresponding Timeout Periods are shown in Table 8-3.

Table 8-3. Watchdog Timer Prescale Select

WDP3	WDP2	WDP1	WDP0	Number of WDT Oscillator Cycles	Typical Time-out at V_{CC} = 5.0V
0	0	0	0	2K (2048) cycles	16 ms
0	0	0	1	4K (4096) cycles	32 ms
0	0	1	0	8K (8192) cycles	64 ms
0	0	1	1	16K (16384) cycles	0.125 s
0	1	0	0	32K (32764) cycles	0.25 s
0	1	0	1	64K (65536) cycles	0.5 s
0	1	1	0	128K (131072) cycles	1.0 s
0	1	1	1	256K (262144) cycles	2.0 s
1	0	0	0	512K (524288) cycles	4.0 s
1	0	0	1	1024K (1048576) cycles	8.0 s
1	0	1	0	Reserved[1]	
1	0	1	1		
1	1	0	0		
1	1	0	1		
1	1	1	0		
1	1	1	1		

Notes: 1. If selected, one of the valid settings below 0b1010 will be used.

rupt will be generated. The start-up time is defined by the SUT and CKSEL Fuses as described in “Clock System” on page 24.

If the low level on the interrupt pin is removed before the device has woken up then program execution will not be diverted to the interrupt service routine but continue from the instruction following the SLEEP command.

9.3 Register Description

9.3.1 MCUCR – MCU Control Register

The MCU Register contains control bits for interrupt sense control.

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	

- **Bits 1, 0 – ISC01, ISC00: Interrupt Sense Control 0 Bit 1 and Bit 0**

The External Interrupt 0 is activated by the external pin INT0 or INT1 if the SREG I-flag and the corresponding interrupt mask are set. The level and edges on the external INT0 or INT1 pin that activate the interrupt are defined in Table 9-2. The value on the INT0 or INT1 pin is sampled before detecting edges. If edge or toggle interrupt is selected, pulses that last longer than one clock period will generate an interrupt. Shorter pulses are not guaranteed to generate an interrupt. If low level interrupt is selected, the low level must be held until the completion of the currently executing instruction to generate an interrupt.

Table 9-2. Interrupt 0 Sense Control

ISC01	ISC00	Description
0	0	The low level of INT0 or INT1 generates an interrupt request.
0	1	Any logical change on INT0 or INT1 generates an interrupt request.
1	0	The falling edge of INT0 or INT1 generates an interrupt request.
1	1	The rising edge of INT0 or INT1 generates an interrupt request.

9.3.2 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

- ADC10: ADC input Channel 10. Note that ADC input channel 10 uses analog power.
- PCINT15: Pin Change Interrupt source 15.

- **Port B, Bit 6 – ADC9/ T0/ INT0/ PCINT14**
  - ADC9: ADC input Channel 9. Note that ADC input channel 9 uses analog power.
  - T0: Timer/Counter0 counter source.
  - INT0: The PB6 pin can serve as an External Interrupt source 0.
  - PCINT14: Pin Change Interrupt source 14.

- **Port B, Bit 5 – XTAL2/ CLKO/ ADC8/ PCINT13**
  - XTAL2: Chip clock Oscillator pin 2. Used as clock pin for crystal Oscillator or Low-frequency crystal Oscillator. When used as a clock pin, the pin can not be used as an I/O pin.
  - CLKO: The divided system clock can be output on the PB5 pin, if the CKOUT Fuse is programmed, regardless of the PORTB5 and DDB5 settings. It will also be output during reset.
  - OC1D Output Compare Match output: The PB5 pin can serve as an external output for the Timer/Counter1 Compare Match D when configured as an output (DDA1 set). The OC1D pin is also the output pin for the PWM mode timer function.
  - ADC8: ADC input Channel 8. Note that ADC input channel 8 uses analog power.
  - PCINT13: Pin Change Interrupt source 13.

- **Port B, Bit 4 – XTAL1/ CLKI/ OC1B/ ADC7/ PCINT12**
  - XTAL1/CLKI: Chip clock Oscillator pin 1. Used for all chip clock sources except internal calibrated RC Oscillator. When used as a clock pin, the pin can not be used as an I/O pin.
  - $\overline{\text{OC1D}}$: Inverted Output Compare Match output: The PB4 pin can serve as an external output for the Timer/Counter1 Compare Match D when configured as an output (DDA0 set). The $\overline{\text{OC1D}}$ pin is also the inverted output pin for the PWM mode timer function.
  - ADC7: ADC input Channel 7. Note that ADC input channel 7 uses analog power.
  - PCINT12: Pin Change Interrupt source 12.

- **Port B, Bit 3 – OC1B/ PCINT11**
  - OC1B, Output Compare Match output: The PB3 pin can serve as an external output for the Timer/Counter1 Compare Match B. The PB3 pin has to be configured as an output (DDB3 set (one)) to serve this function. The OC1B pin is also the output pin for the PWM mode timer function.
  - PCINT11: Pin Change Interrupt source 11.

- **Port B, Bit 2 – SCK/ USCK/ SCL/ $\overline{\text{OC1B}}$/ PCINT10**
  - USCK: Three-wire mode Universal Serial Interface Clock.
  - SCL: Two-wire mode Serial Clock for USI Two-wire mode.
  - $\overline{\text{OC1B}}$: Inverted Output Compare Match output: The PB2 pin can serve as an external output for the Timer/Counter1 Compare Match B when configured as an output (DDB2 set). The $\overline{\text{OC1B}}$ pin is also the inverted output pin for the PWM mode timer function.
  - PCINT10: Pin Change Interrupt source 10.

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.

if a Compare Match in Timer/Counter0 occurs, i.e., when the OCF0A bit is set in the Timer/Counter 0 Interrupt Flag Register – TIFR0.

- **Bit 3 – OCIE0B: Timer/Counter Output Compare Match B Interrupt Enable**

When the OCIE0B bit is written to one, and the I-bit in the Status Register is set, the Timer/Counter Compare Match B interrupt is enabled. The corresponding interrupt is executed if a Compare Match in Timer/Counter occurs, i.e., when the OCF0B bit is set in the Timer/Counter Interrupt Flag Register – TIFR0.

- **Bit 1 – TOIE0: Timer/Counter0 Overflow Interrupt Enable**

When the TOIE0 bit is written to one, and the I-bit in the Status Register is set, the Timer/Counter0 Overflow interrupt is enabled. The corresponding interrupt is executed if an overflow in Timer/Counter0 occurs, i.e., when the TOV0 bit is set in the Timer/Counter 0 Interrupt Flag Register – TIFR0.

- **Bit 0 – TICIE0: Timer/Counter0, Input Capture Interrupt Enable**

When this bit is written to one, and the I-flag in the Status Register is set (interrupts globally enabled), the Timer/Counter1 Input Capture interrupt is enabled. The corresponding Interrupt Vector (See “Interrupts” on page 50.) is executed when the ICF0 flag, located in TIFR, is set.

11.10.8 TIFR – Timer/Counter0 Interrupt Flag Register

Bit	7	6	5	4	3	2	1	0	
0x38 (0x58)	OCF1D	OCF1A	OCF1B	OCF0A	OCF0B	TOV1	TOV0	ICF0	TIFR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 4 – OCF0A: Output Compare Flag 0 A**

The OCF0A bit is set when a Compare Match occurs between the Timer/Counter0 and the data in OCR0A – Output Compare Register0. OCF0A is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, OCF0A is cleared by writing a logic one to the flag. When the I-bit in SREG, OCIE0A (Timer/Counter0 Compare Match Interrupt Enable), and OCF0A are set, the Timer/Counter0 Compare Match Interrupt is executed.

The OCF0A is also set in 16-bit mode when a Compare Match occurs between the Timer/Counter and 16-bit data in OCR0B/A. The OCF0A is not set in Input Capture mode when the Output Compare Register OCR0A is used as an Input Capture Register.

- **Bit 3 – OCF0B: Output Compare Flag 0 B**

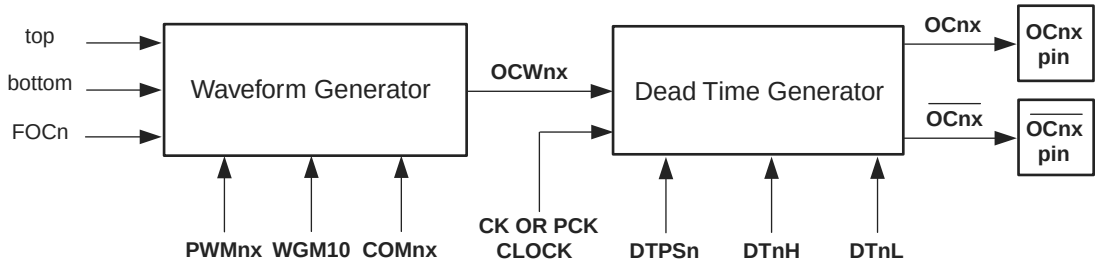
The OCF0B bit is set when a Compare Match occurs between the Timer/Counter and the data in OCR0B – Output Compare Register0 B. OCF0B is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, OCF0B is cleared by writing a logic one to the flag. When the I-bit in SREG, OCIE0B (Timer/Counter Compare B Match Interrupt Enable), and OCF0B are set, the Timer/Counter Compare Match Interrupt is executed.

The OCF0B is not set in 16-bit Output Compare mode when the Output Compare Register OCR0B is used as the high byte of the 16-bit Output Compare Register or in 16-bit Input Capture mode when the Output Compare Register OCR0B is used as the high byte of the Input Capture Register.

12.6 Dead Time Generator

The Dead Time Generator is provided for the Timer/Counter1 PWM output pairs to allow driving external power control switches safely. The Dead Time Generator is a separate block that can be used to insert dead times (non-overlapping times) for the Timer/Counter1 complementary output pairs OC1x and $\overline{OC1x}$ when the PWM mode is enabled and the COM1x1:0 bits are set to “01”. See Figure 12-7 below.

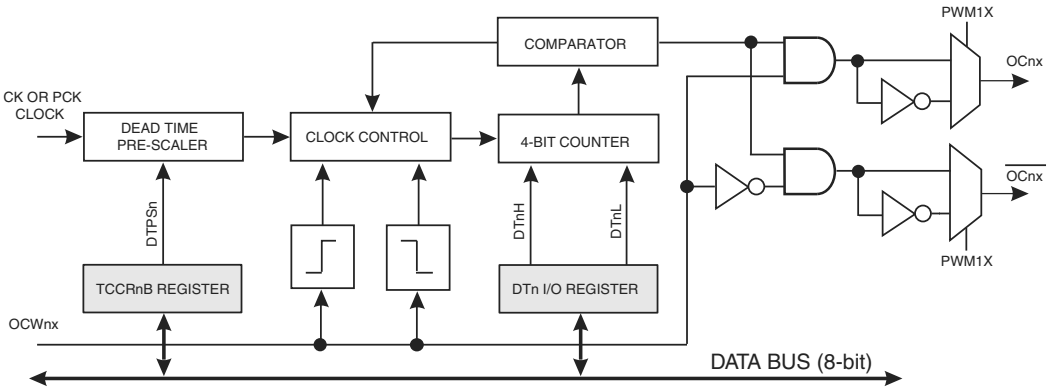
Figure 12-7. Block Diagram of Waveform Generator and Dead Time Generator.



The tasks are shared as follows: the Waveform Generator generates the output (OCW1x) and the Dead Time Generator generates the non-overlapping PWM output pair from the output. Three Dead Time Generators are provided, one for each PWM output. The non-overlap time is adjustable and the PWM output and it’s complementary output are adjusted separately, and independently for both PWM outputs.

The Dead Time Generation is based on 4-bit down counters that count the dead time, as shown in Figure 12-8.

Figure 12-8. Dead Time Generator



There is a dedicated prescaler in front of the Dead Time Generator that can divide the Timer/Counter1 clock (PCK or CK) by 1, 2, 4 or 8. This provides for large range of dead times that can be generated. The prescaler is controlled by two control bits DTPS11:10. The block has also a rising and falling edge detector that is used to start the dead time counting period. Depending on the edge, one of the transitions on the rising edges, OC1x or $\overline{OC1x}$ is delayed until the counter has counted to zero. The comparator is used to compare the counter with zero and stop the dead time insertion when zero has been reached. The counter is loaded with a 4-bit DT1H or DT1L value from DT1 I/O register, depending on the edge of the Waveform Output (OCW1x) when the dead time insertion is started. The Output Compare Output are delayed by one timer clock cycle at minimum from the Waveform Output when the Dead Time is adjusted to

- **Bit 3 – FPAC1: Fault Protection Analog Comparator Enable**

When written logic one, this bit enables the Fault Protection function in Timer/Counter1 to be triggered by the Analog Comparator. The comparator output is in this case directly connected to the Fault Protection front-end logic, making the comparator utilize the noise canceler and edge select features of the Timer/Counter1 Fault Protection interrupt. When written logic zero, no connection between the Analog Comparator and the Fault Protection function exists. To make the comparator trigger the Timer/Counter1 Fault Protection interrupt, the FPIE1 bit in the Timer/Counter1 Control Register D (TCCR1D) must be set.

- **Bit 2 – FPF1: Fault Protection Interrupt Flag**

When the FPIE1 bit is set (one), the Fault Protection Interrupt is enabled. Activity on the pin will cause an interrupt request even, if the Fault Protection pin is configured as an output. The corresponding interrupt of Fault Protection Interrupt Request is executed from the Fault Protection Interrupt Vector. The bit FPF1 is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, FPF1 is cleared after a synchronization clock cycle by writing a logical one to the flag. When the SREG I-bit, FPIE1 and FPF1 are set, the Fault Interrupt is executed.

- **Bits 1:0 – WGM11, WGM10: Waveform Generation Mode Bits**

These bits together with the PWM1A/PWM1B bits control the counting sequence of the counter and the type of waveform generation to be used, as shown in Table 12-21. Modes of operation supported by the Timer/Counter1 are: Normal mode (counter), Fast PWM Mode, Phase and Frequency Correct PWM and PWM6 Modes.

Table 12-21. Waveform Generation Mode Bit Description

PWM1A/ PWM1B	WGM11:10	Timer/Counter Mode of Operation	TOP	Update OCR1x at	Set TOV1 Flag at
0	XX	Normal	OCR1C	Immediate	TOP
1	00	Fast PWM	OCR1C	TOP	TOP
1	01	Phase & Frequency Correct PWM	OCR1C	BOTTOM	BOTTOM
1	10	PWM6 / Single-slope	OCR1C	TOP	TOP
1	11	PWM6 / Dual-slope	OCR1C	BOTTOM	BOTTOM

12.12.5 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 and always read zero.

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

These bits are the Ouput 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

13. USI – Universal Serial Interface

13.1 Features

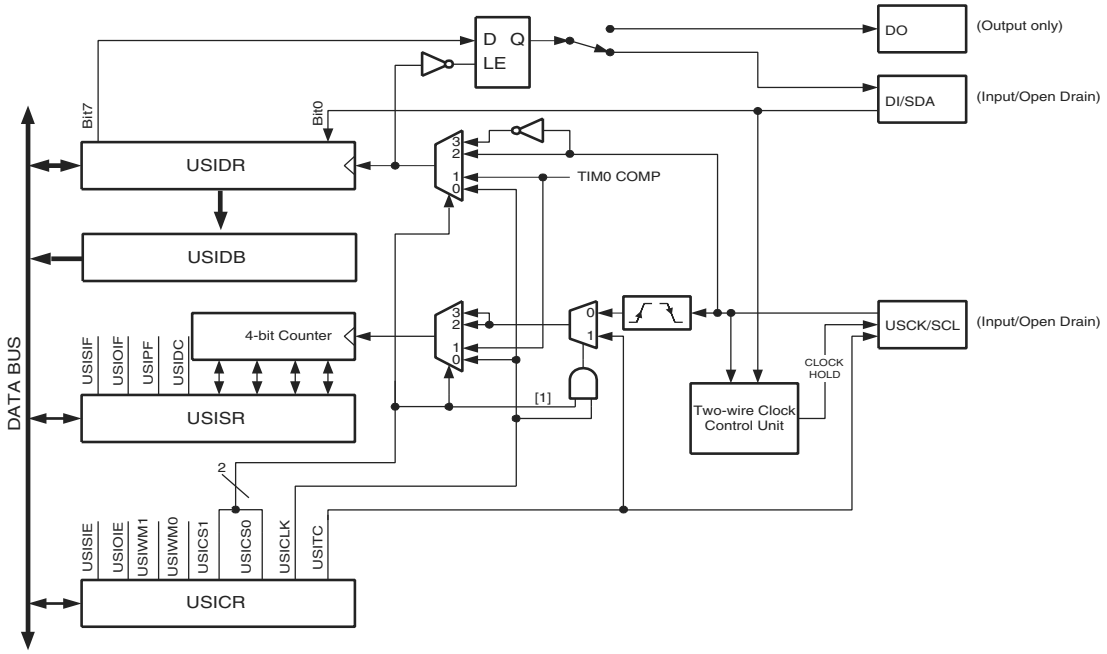
- **Two-wire Synchronous Data Transfer (Master or Slave)**
- **Three-wire Synchronous Data Transfer (Master or Slave)**
- **Data Received Interrupt**
- **Wakeup from Idle Mode**
- **In Two-wire Mode: Wake-up from All Sleep Modes, Including Power-down Mode**
- **Two-wire Start Condition Detector with Interrupt Capability**

13.2 Overview

The Universal Serial Interface, or USI, provides the basic hardware resources needed for serial communication. Combined with a minimum of control software, the USI allows significantly higher transfer rates and uses less code space than solutions based on software only. Interrupts are included to minimize the processor load.

A simplified block diagram of the USI is shown in Figure 13-1 For actual placement of I/O pins refer to “Pinout ATtiny261/461/861 and ATtiny261V/461V/861V” on page 2. Device-specific I/O Register and bit locations are listed in the “Register Descriptions” on page 132.

Figure 13-1. Universal Serial Interface, Block Diagram



The 8-bit USI Data Register (USIDR) is directly accessible via the data bus and contains the incoming and outgoing data. The register has no buffering so the data must be read as quickly as possible to ensure that no data is lost. The data register is a serial shift register where the most significant bit is connected to one of two output pins depending of the wire mode configuration. A transparent latch between the output of the data register and the output pin delays the change of data output to the opposite clock edge of the data input sampling. The serial input is always sampled from the Data Input (DI) pin, regardless of the configuration.

The 4-bit counter increments by one for each clock generated either by the external clock edge detector, by a Timer/Counter0 Compare Match, or by software using USICLK or USITC strobe bits. The clock source depends of the setting of the USICS1:0 bits. For external clock operation a special feature is added that allows the clock to be generated by writing to the USITC strobe bit. This feature is enabled by write a one to the USICLK bit while setting an external clock source (USICS1 = 1).

Note that even when no wire mode is selected (USIWM1:0 = 0) the external clock input (USCK/SCL) are can still be used by the counter.

13.5.4 USICR – USI Control Register

Bit	7	6	5	4	3	2	1	0	
0x0D (0x2D)	**USISIE**	**USIOIE**	**USIWM1**	**USIWM0**	**USICS1**	**USICS0**	**USICLK**	**USITC**	**USICR**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	W	W	
Initial Value	0	0	0	0	0	0	0	0	

The Control Register includes interrupt enable control, wire mode setting, Clock Select setting, and clock strobe.

- **Bit 7 – USISIE: Start Condition Interrupt Enable**

Setting this bit to one enables the Start Condition detector interrupt. If there is a pending interrupt when the USISIE and the Global Interrupt Enable Flag is set to one, this will immediately be executed. Refer to the USISIF bit description on page 133 for further details.

- **Bit 6 – USIOIE: Counter Overflow Interrupt Enable**

Setting this bit to one enables the Counter Overflow interrupt. If there is a pending interrupt when the USIOIE and the Global Interrupt Enable Flag is set to one, this will immediately be executed. Refer to the USIOIF bit description on page 133 for further details.

- **Bits 5:4 – USIWM1:0: Wire Mode**

These bits set the type of wire mode to be used, as shown in Table 13-1 on page 134.

Basically, only the function of the outputs are affected by these bits. Data and clock inputs are not affected by the mode selected and will always have the same function. The counter and USI Data Register can therefore be clocked externally, and data input sampled, even when outputs are disabled.

Table 13-1. Relationship between USIWM1:0 and USI Operation

USIWM1	USIWM0	Description
0	0	Outputs, clock hold, and start detector disabled. Port pins operate as normal.
0	1	Three-wire mode. Uses DO, DI, and USCK pins. The *Data Output* (DO) pin overrides the corresponding bit in the PORTA register. However, the corresponding DDRA bit still controls the data direction. When the port pin is set as input the pin pull-up is controlled by the PORTA bit. The *Data Input* (DI) and *Serial Clock* (USCK) pins do not affect the normal port operation. When operating as master, clock pulses are software generated by toggling the PORTA register, while the data direction is set to output. The USITC bit in the USICR Register can be used for this purpose.

14.2 Register Description

14.2.1 ACSRA – Analog Comparator Control and Status Register A

Bit	7	6	5	4	3	2	1	0	
0x08 (0x28)	ACD	ACBG	ACO	ACI	ACIE	ACME	ACIS1	ACIS0	ACSRA
Read/Write	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	N/A	0	0	0	0	0	

- **Bit 7 – ACD: Analog Comparator Disable**

When this bit is written logic one, the power to the analog comparator is switched off. This bit can be set at any time to turn off the analog comparator, thus reducing power consumption in Active and Idle mode. When changing the ACD bit, the analog comparator Interrupt must be disabled by clearing the ACIE bit in ACSRA. Otherwise an interrupt can occur when the bit is changed.

- **Bit 6 – ACBG: Analog Comparator Bandgap Select**

When this bit is set an internal 1.1V reference voltage replaces the positive input to the analog comparator. The selection of the internal voltage reference is done by writing the REFS2:0 bits in ADCSRB and ADMUX registers. When this bit is cleared, AIN0, AIN1 or AIN2 depending on the ACM2:0 bits is applied to the positive input of the analog comparator.

- **Bit 5 – ACO: Analog Comparator Output**

Enables output of analog comparator. The output of the analog comparator is synchronized and then directly connected to ACO. The synchronization introduces a delay of 1 - 2 clock cycles.

- **Bit 4 – ACI: Analog Comparator Interrupt Flag**

This bit is set by hardware when a comparator output event triggers the interrupt mode defined by ACIS1 and ACIS0. The analog comparator interrupt routine is executed if the ACIE bit is set and the I-bit in SREG is set. ACI is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, ACI is cleared by writing a logic one to the flag.

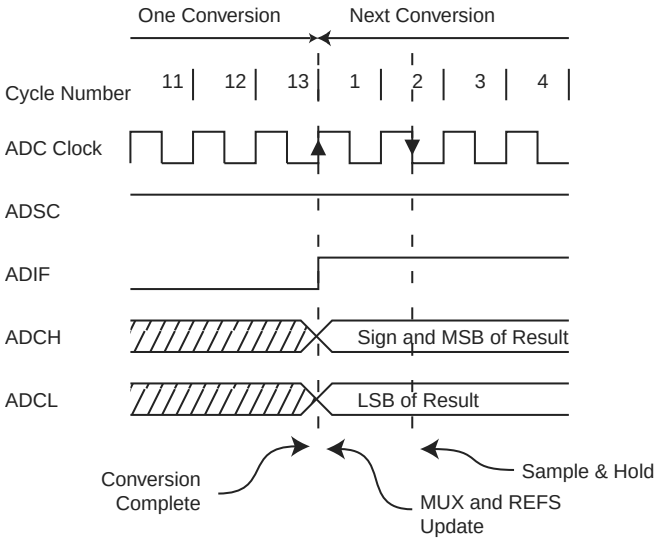
- **Bit 3 – ACIE: Analog Comparator Interrupt Enable**

When the ACIE bit is written logic one and the I-bit in the status register is set, the analog comparator interrupt is activated. When written logic zero, the interrupt is disabled.

- **Bit 2 – ACME: Analog Comparator Multiplexer Enable**

When this bit is written logic one and the ADC is switched off (ADEN in ADCSRA is zero), the ADC multiplexer selects the negative input to the analog comparator. When this bit is written logic zero, AIN1 is applied to the negative input of the analog comparator. For a detailed description of this bit, see Table 14-1 on page 137.

Figure 15-7. ADC Timing Diagram, Free Running Conversion



For a summary of conversion times, see Table 15-1.

Table 15-1. ADC Conversion Time

Condition	Sample & 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

15.6 Changing Channel or Reference Selection

The MUX5:0 and REFS2:0 bits in the ADCSRB and ADMUX registers are single buffered through a temporary register to which the CPU has random access. This ensures that the channels and reference selection only takes place at a safe point during the conversion. The channel and reference selection is continuously updated until a conversion is started. Once the conversion starts, the channel and reference selection is locked to ensure a sufficient sampling time for the ADC. Continuous updating resumes in the last ADC clock cycle before the conversion completes (ADIF in ADCSRA is set). Note that the conversion starts on the following rising ADC clock edge after ADSC is written. The user is thus advised not to write new channel or reference selection values to ADMUX until one ADC clock cycle after ADSC is written.

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

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

ADMUX can be safely updated in the following ways:

Figure 18-3. Parallel Programming.

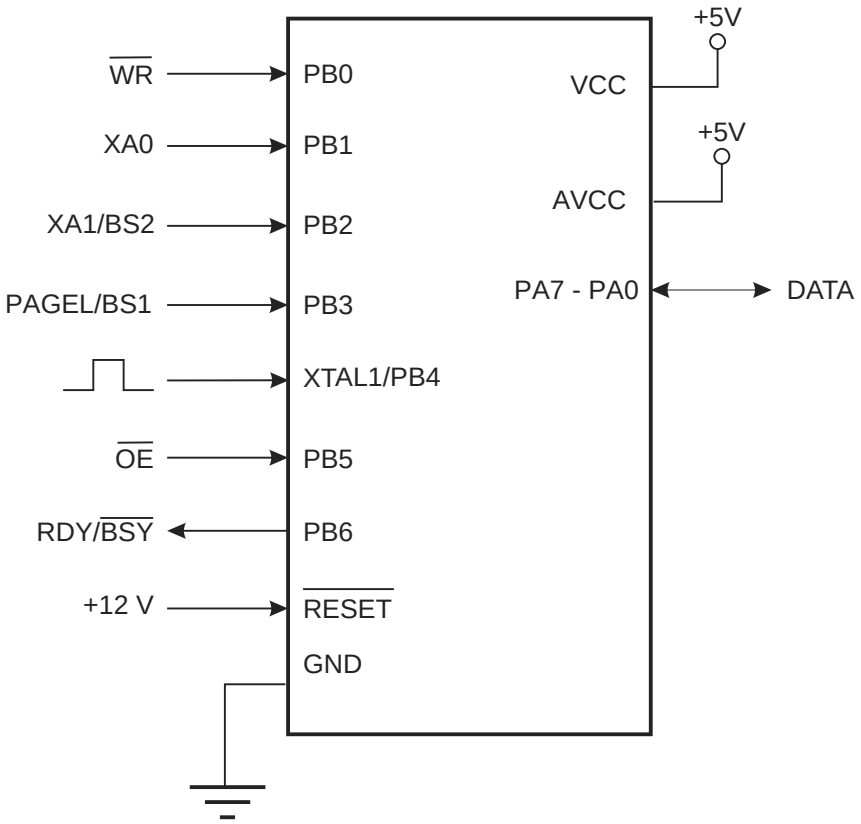


Table 18-12. Pin Name Mapping

Signal Name in Programming Mode	Pin Name	I/O	Function
$\overline{WR}$	PB0	I	Write Pulse (Active low).
XA0	PB1	I	XTAL Action Bit 0
XA1/BS2	PB2	I	XTAL Action Bit 1. Byte Select 2 (“0” selects low byte, “1” selects 2’nd high byte).
PAGEL/BS1	PB3	I	Byte Select 1 (“0” selects low byte, “1” selects high byte). Program Memory and EEPROM Data Page Load.
$\overline{OE}$	PB5	I	Output Enable (Active low).
RDY/$\overline{BSY}$	PB6	O	0: Device is busy programming, 1: Device is ready for new command.
DATA I/O	PA7-PA0	I/O	Bi-directional Data bus (Output when $\overline{OE}$ is low).

3. Load Data Low Byte:
   a. Set XA1, XA0 to “01”. This enables data loading.
   b. Set DATA = Data low byte (0x00 - 0xFF).
   c. Give XTAL1 a positive pulse. This loads the data byte.
4. Load Data High Byte:
   a. Set BS1 to “1”. This selects high data byte.
   b. Keep XA1, XA0 at “01”. This enables data loading.
   c. Set DATA = Data high byte (0x00 - 0xFF).
   d. Give XTAL1 a positive pulse. This loads the data byte.
5. Repeat steps 2 to 4 until the entire buffer is filled or until all data within the page is loaded.
6. Load Address High byte:
   a. Set XA1, XA0 to “00”. This enables address loading.
   b. Set BS1 to “1”. This selects high address.
   c. Set DATA = Address high byte (0x00 - 0xFF).
   d. Give XTAL1 a positive pulse. This loads the address high byte.
7. Program Page:
   a. Give $\overline{\text{WR}}$ a negative pulse. This starts programming of the entire page of data. RDY/$\overline{\text{BSY}}$ goes low.
   b. Wait until RDY/$\overline{\text{BSY}}$ goes high.
8. Repeat steps 2 to 7 until the entire Flash is programmed or until all data has been programmed.
9. End Page Programming:
   a. Set XA1, XA0 to “10”. This enables command loading.
   b. Set DATA to “0000 0000”. This is the command for No Operation.
   c. Give XTAL1 a positive pulse. This loads the command, and the internal write signals are reset.

While the lower bits in the address are mapped to words within the page, the higher bits address the pages within the FLASH. This is illustrated in Figure 18-4. Note that if less than eight bits are required to address words in the page (pagesize < 256), the most significant bit(s) in the address low byte are used to address the page when performing a Page Write.

19.8 Parallel Programming Characteristics

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

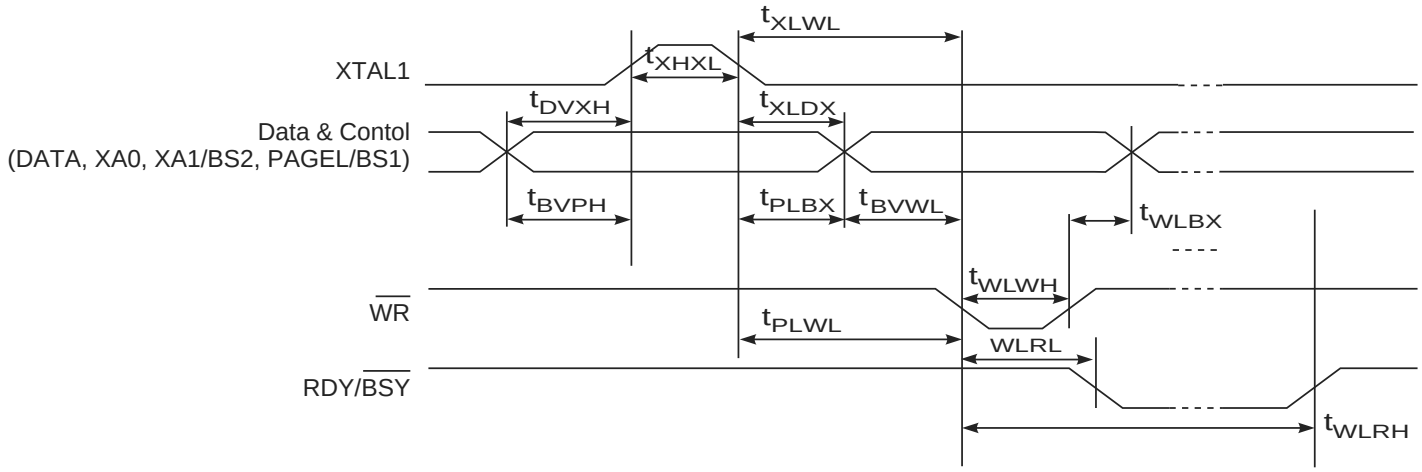
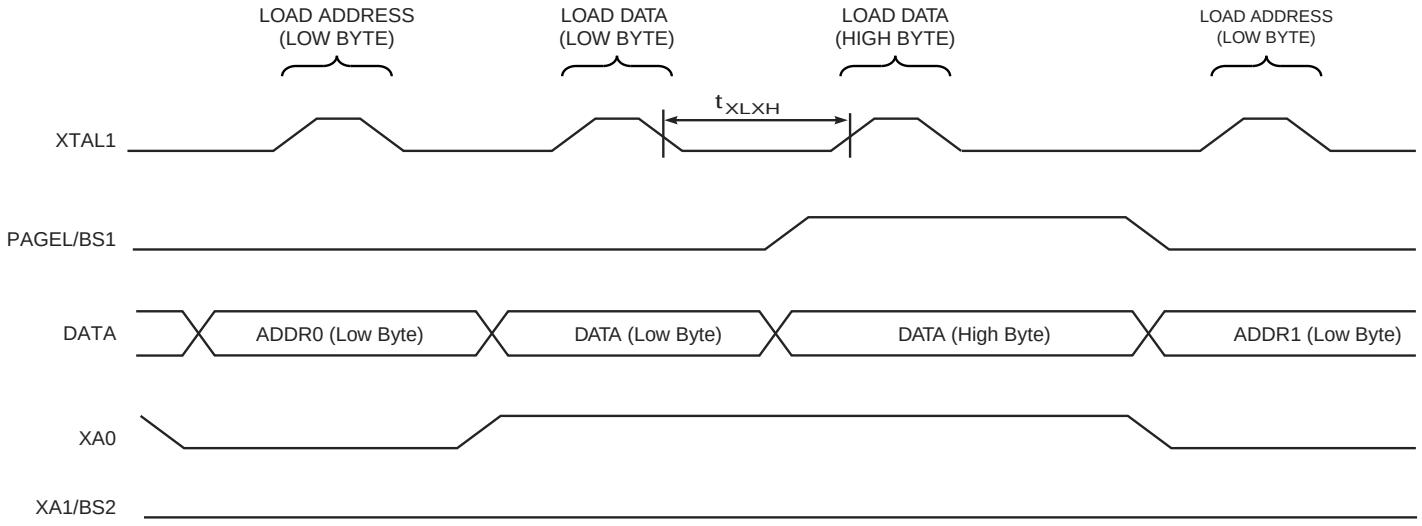
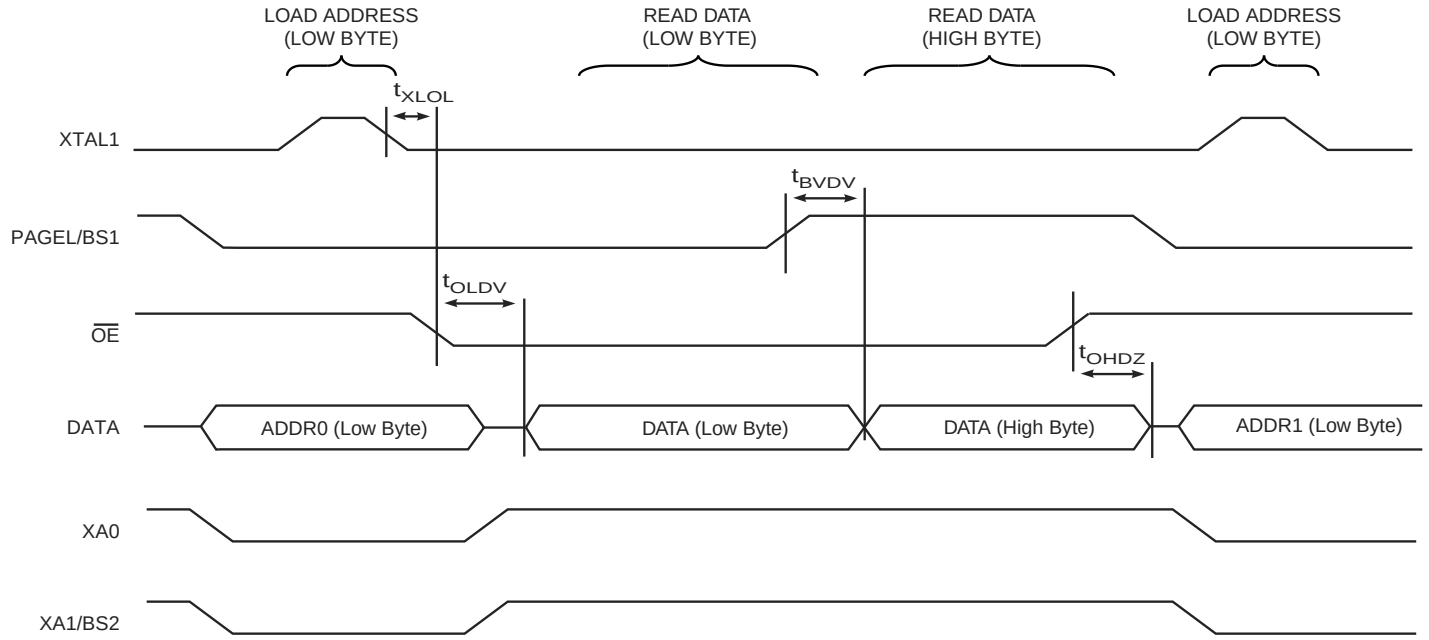


Figure 19-7. Parallel Programming Timing, Loading Sequence with Timing Requirements



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

Figure 19-8. Parallel Programming Timing, Reading Sequence (within the Same Page) with Timing Requirements



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

Table 19-9. Parallel Programming Characteristics, V_{CC} = 5V ± 10%

Symbol	Parameter	Min	Typ	Max	Units
V_{PP}	Programming Enable Voltage	11.5		12.5	V
I_{PP}	Programming Enable Current			250	µA
t_{DVXH}	Data and Control Valid before XTAL1 High	67			ns
t_{XLXH}	XTAL1 Low to XTAL1 High	200			ns
t_{XHXL}	XTAL1 Pulse Width High	150			ns
t_{XLDX}	Data and Control Hold after XTAL1 Low	67			ns
t_{XLWL}	XTAL1 Low to $\overline{WR}$ Low	0			ns
t_{BVPH}	BS1 Valid before PAGEL High	67			ns
t_{PHPL}	PAGEL Pulse Width High	150			ns
t_{PLBX}	BS1 Hold after PAGEL Low	67			ns
t_{WLBX}	BS2/1 Hold after $\overline{WR}$ Low	67			ns
t_{PLWL}	PAGEL Low to $\overline{WR}$ Low	67			ns
t_{BVWL}	BS1 Valid to $\overline{WR}$ Low	67			ns
t_{WLWH}	$\overline{WR}$ Pulse Width Low	150			ns
t_{WLRL}	$\overline{WR}$ Low to RDY/$\overline{BSY}$ Low	0		1	µs
t_{WLRH}	$\overline{WR}$ Low to RDY/$\overline{BSY}$ High[1]	3.7		4.5	ms
t_{WLRH_CE}	$\overline{WR}$ Low to RDY/$\overline{BSY}$ High for Chip Erase[2]	7.5		9	ms
t_{XLOL}	XTAL1 Low to $\overline{OE}$ Low	0			ns

Figure 20-22. I/O Pin Output Voltage vs. Source Current (V_{CC} = 5V)

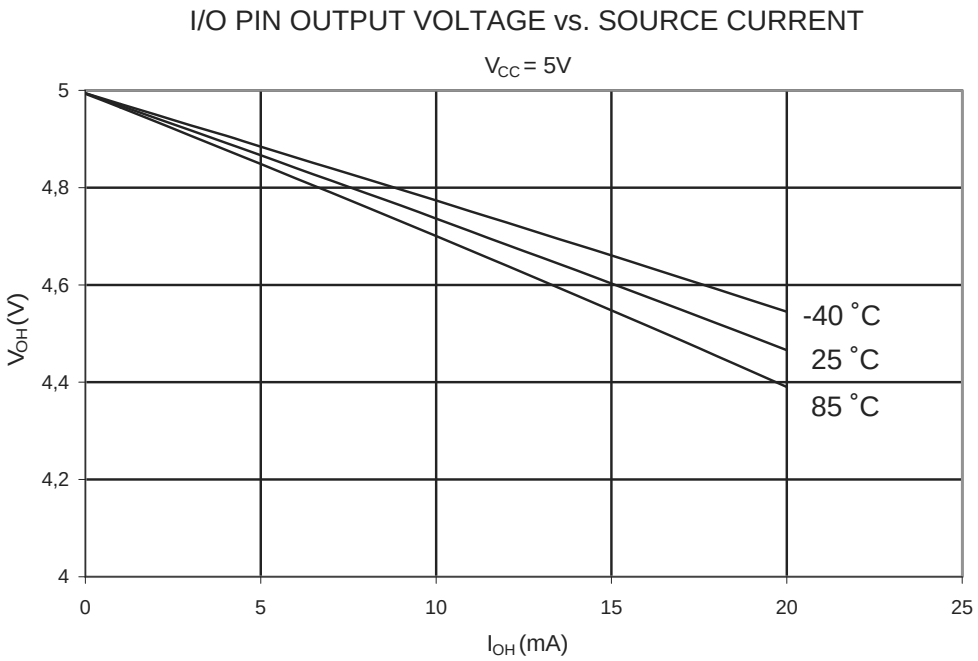


Figure 20-23. Reset Pin Output Voltage vs. Sink Current (V_{CC} = 3V)

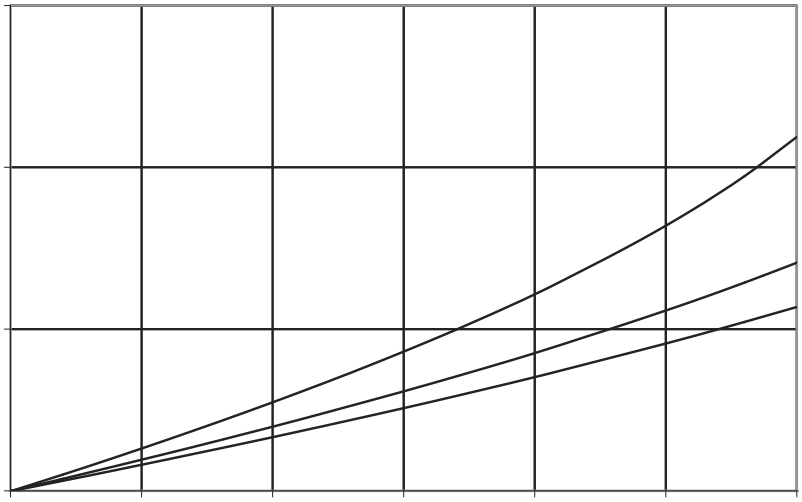


Figure 20-28. I/O Pin Input Threshold Voltage vs. V_{CC} (V_{IL}, I/O Pin Read as '0')

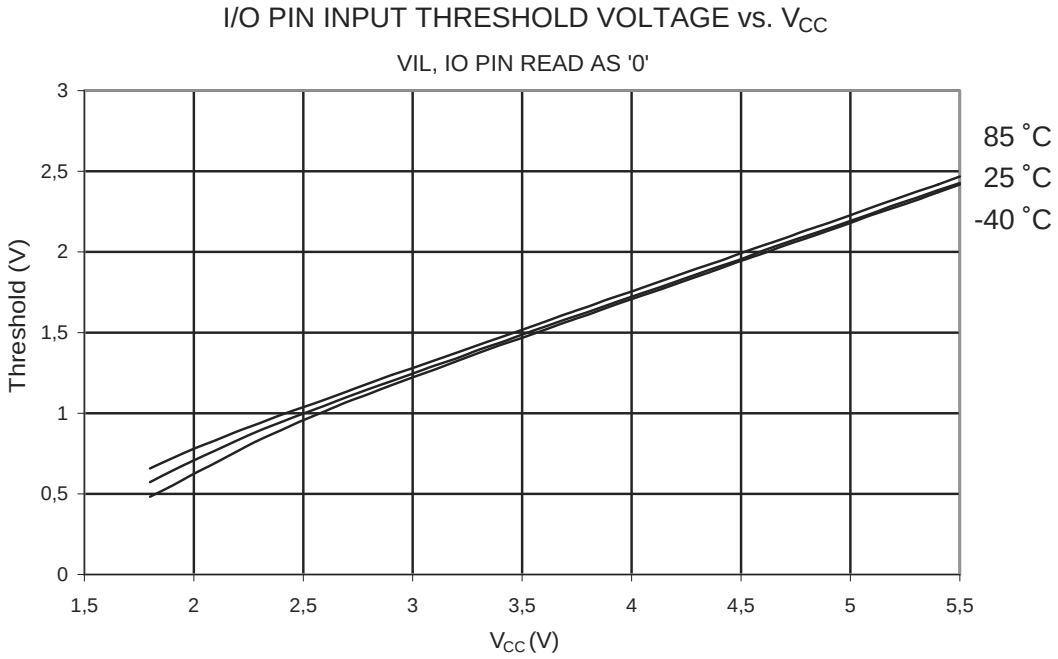


Figure 20-29. I/O Pin Input Hysteresis vs. V_{CC}