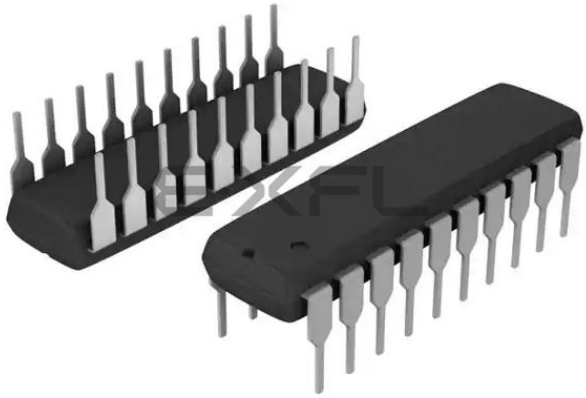


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

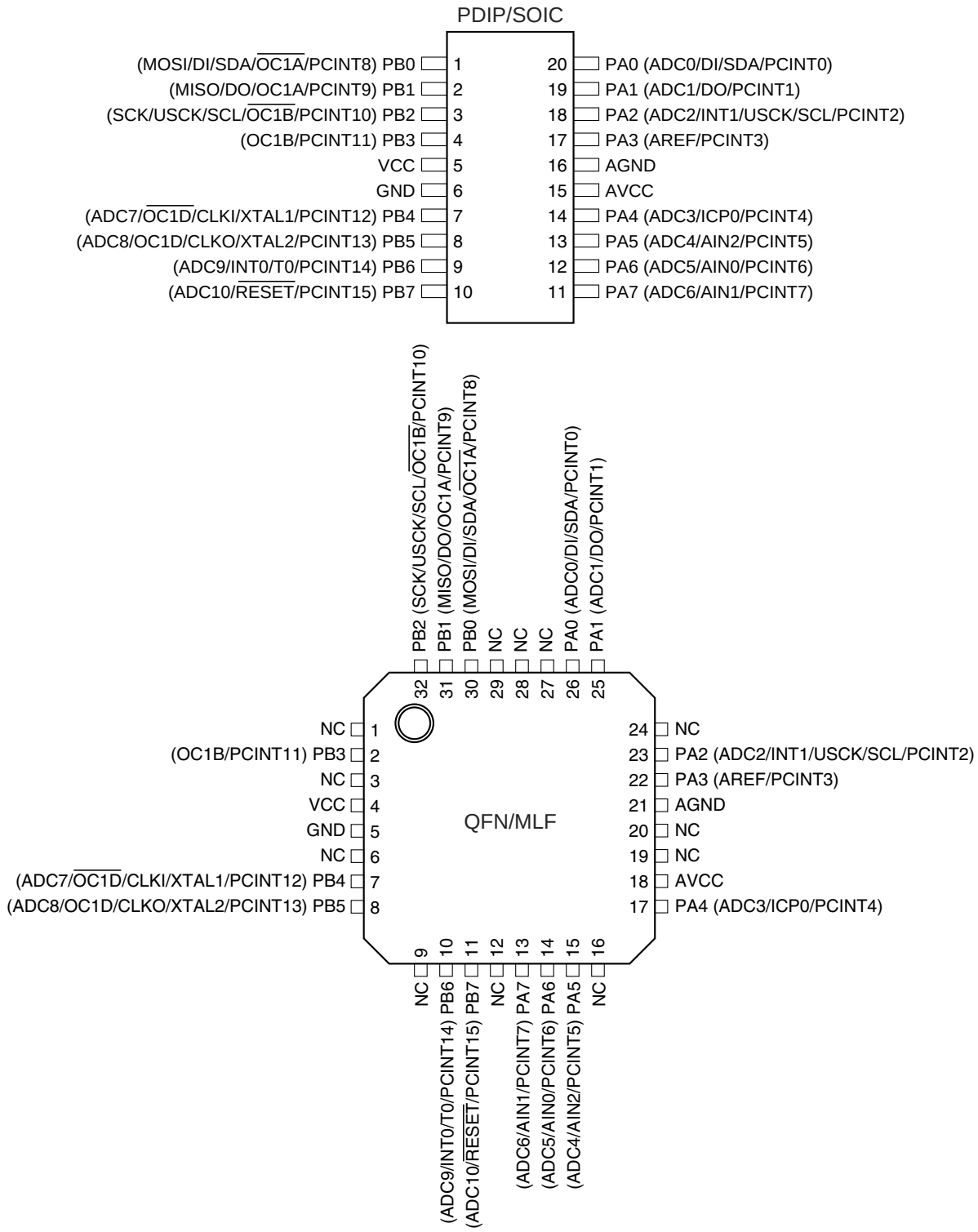
"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" 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	10MHz
Connectivity	USI
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	16
Program Memory Size	8KB (4K x 16)
Program Memory Type	FLASH
EEPROM Size	512 x 8
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 5.5V
Data Converters	A/D 11x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Through Hole
Package / Case	20-DIP (0.300", 7.62mm)
Supplier Device Package	20-PDIP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/attiny861v-10pu

1. Pin Configurations

Figure 1-1. Pinout ATtiny261/461/861 and ATtiny261V/461V/861V



Note: To ensure mechanical stability the center pad underneath the QFN/MLF package should be soldered to ground on the board.

1.1 Pin Descriptions

1.1.1 VCC

Supply voltage.

1.1.2 GND

Ground.

1.1.3 AVCC

Analog supply voltage. This is the supply voltage pin for the Analog-to-digital Converter (ADC), the analog comparator, the Brown-Out Detector (BOD), the internal voltage reference and Port A. It should be externally connected to VCC, even if some peripherals such as the ADC are not used. If the ADC is used AVCC should be connected to VCC through a low-pass filter.

1.1.4 AGND

Analog ground.

1.1.5 Port A (PA7:PA0)

An 8-bit, bi-directional I/O port with internal pull-up resistors, individually selectable for each bit. Output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, port pins that are externally pulled low will source current if pull-up resistors have been activated. Port pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port A also serves the functions of various special features of the device, as listed on page 63.

1.1.6 Port B (PB7:PB0)

An 8-bit, bi-directional I/O port with internal pull-up resistors, individually selectable for each bit. Output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, port pins that are externally pulled low will source current if pull-up resistors have been activated. Port pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port B also serves the functions of various special features of the device, as listed on page 66.

1.1.7 $\overline{\text{RESET}}$

Reset input. A low level on this pin for longer than the minimum pulse length will generate a reset, even if the clock is not running and provided the reset pin has not been disabled. The minimum pulse length is given in Table 19-4 on page 190. Shorter pulses are not guaranteed to generate a reset.

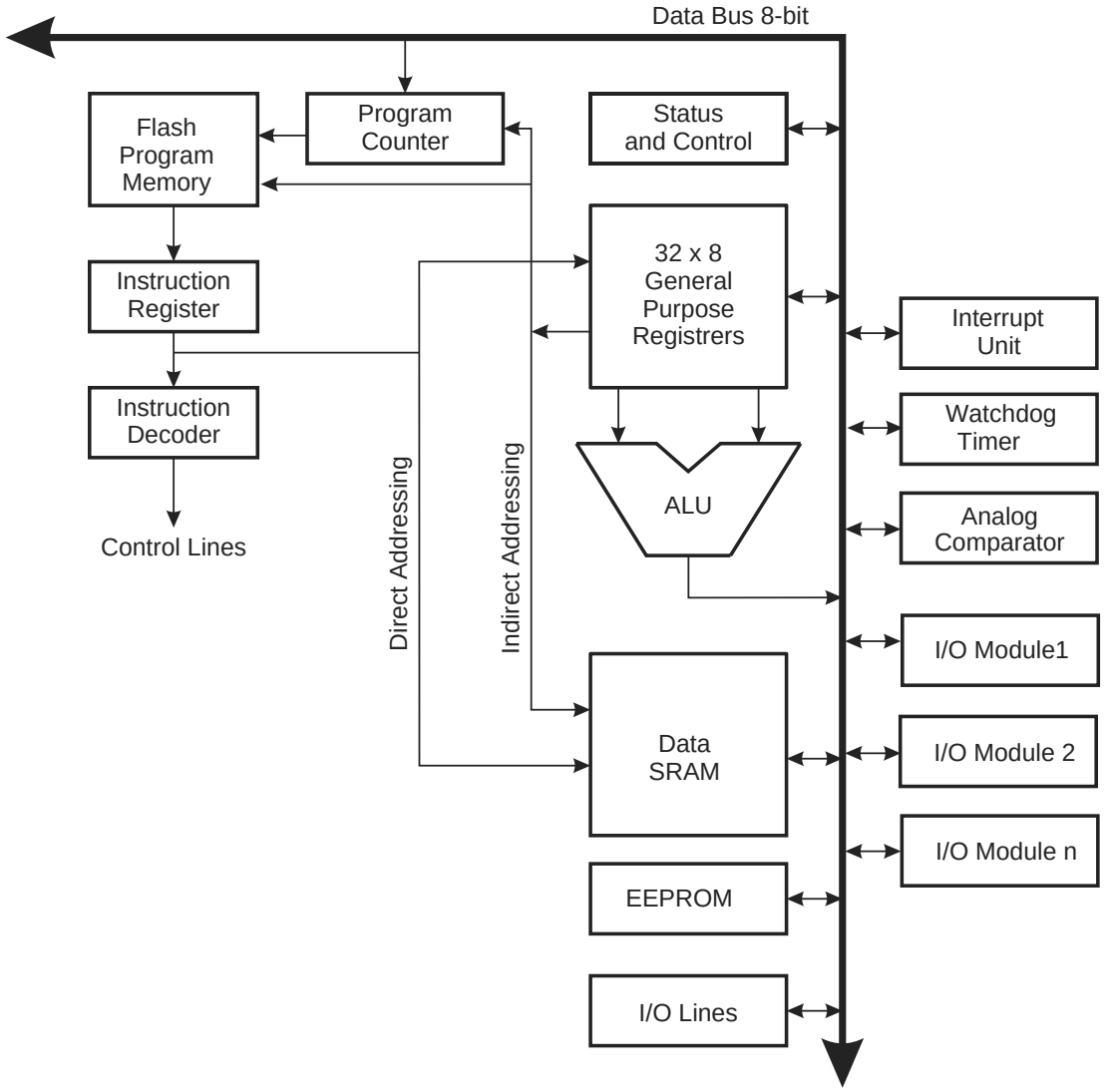
The reset pin can also be used as a (weak) I/O pin.

4. CPU Core

This section discusses the AVR core architecture in general. The main function of the CPU core is to ensure correct program execution. The CPU must therefore be able to access memories, perform calculations, control peripherals, and handle interrupts.

4.1 Architectural Overview

Figure 4-1. Block Diagram of the AVR Architecture



In order to maximize performance and parallelism, the AVR uses a Harvard architecture – with separate memories and buses for program and data. Instructions in the Program memory are executed with a single level pipelining. While one instruction is being executed, the next instruction is pre-fetched from the Program memory. This concept enables instructions to be executed in every clock cycle. The Program memory is In-System Reprogrammable Flash memory.

The fast-access Register File contains 32 x 8-bit general purpose working registers with a single clock cycle access time. This allows single-cycle Arithmetic Logic Unit (ALU) operation. In a typical ALU operation, two operands are output from the Register File, the operation is executed, and the result is stored back in the Register File – in one clock cycle.

Six of the 32 registers can be used as three 16-bit indirect address register pointers for Data Space addressing – enabling efficient address calculations. One of the these address pointers can also be used as an address pointer for look up tables in Flash Program memory. These added function registers are the 16-bit X-, Y-, and Z-register, described later in this section.

The ALU supports arithmetic and logic operations between registers or between a constant and a register. Single register operations can also be executed in the ALU. After an arithmetic operation, the Status Register is updated to reflect information about the result of the operation.

Program flow is provided by conditional and unconditional jump and call instructions, capable of directly addressing the whole address space. Most AVR instructions have a single 16-bit word format but 32-bit wide instructions also exist. The actual instruction set varies, as some devices only implement a part of the instruction set.

During interrupts and subroutine calls, the return address Program Counter (PC) is stored on the Stack. The Stack is effectively allocated in the general data SRAM, and consequently the Stack size is only limited by the total SRAM size and the usage of the SRAM. All user programs must initialize the SP in the Reset routine (before subroutines or interrupts are executed). The Stack Pointer (SP) is read/write accessible in the I/O space. The data SRAM can easily be accessed through the five different addressing modes supported in the AVR architecture.

The memory spaces in the AVR architecture are all linear and regular memory maps.

A flexible interrupt module has its control registers in the I/O space with an additional Global Interrupt Enable bit in the Status Register. All interrupts have a separate Interrupt Vector in the Interrupt Vector table. The interrupts have priority in accordance with their Interrupt Vector position. The lower the Interrupt Vector address, the higher the priority.

The I/O memory space contains 64 addresses for CPU peripheral functions as Control Registers, SPI, and other I/O functions. The I/O memory can be accessed directly, or as the Data Space locations following those of the Register File, 0x20 - 0x5F.

4.2 ALU – Arithmetic Logic Unit

The high-performance AVR ALU operates in direct connection with all the 32 general purpose working registers. Within a single clock cycle, arithmetic operations between general purpose registers or between a register and an immediate are executed. The ALU operations are divided into three main categories – arithmetic, logical, and bit-functions. Some implementations of the architecture also provide a powerful multiplier supporting both signed/unsigned multiplication and fractional format. See the “Instruction Set” section for a detailed description.

4.3 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.

C Code Example

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

Note: See “Code Examples” on page 6.

4.7.1 Interrupt Response Time

The interrupt execution response for all the enabled AVR interrupts is four clock cycles minimum. After four clock cycles the Program Vector address for the actual interrupt handling routine is executed. During this four clock cycle period, the Program Counter is pushed onto the Stack. The vector is normally a jump to the interrupt routine, and this jump takes three clock cycles. If an interrupt occurs during execution of a multi-cycle instruction, this instruction is completed before the interrupt is served. If an interrupt occurs when the MCU is in sleep mode, the interrupt execution response time is increased by four clock cycles. This increase comes in addition to the start-up time from the selected sleep mode.

A return from an interrupt handling routine takes four clock cycles. During these four clock cycles, the Program Counter (two bytes) is popped back from the Stack, the Stack Pointer is incremented by two, and the I-bit in SREG is set.

5.5.6 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	

5.5.7 GPIOR0 – General Purpose I/O Register 0

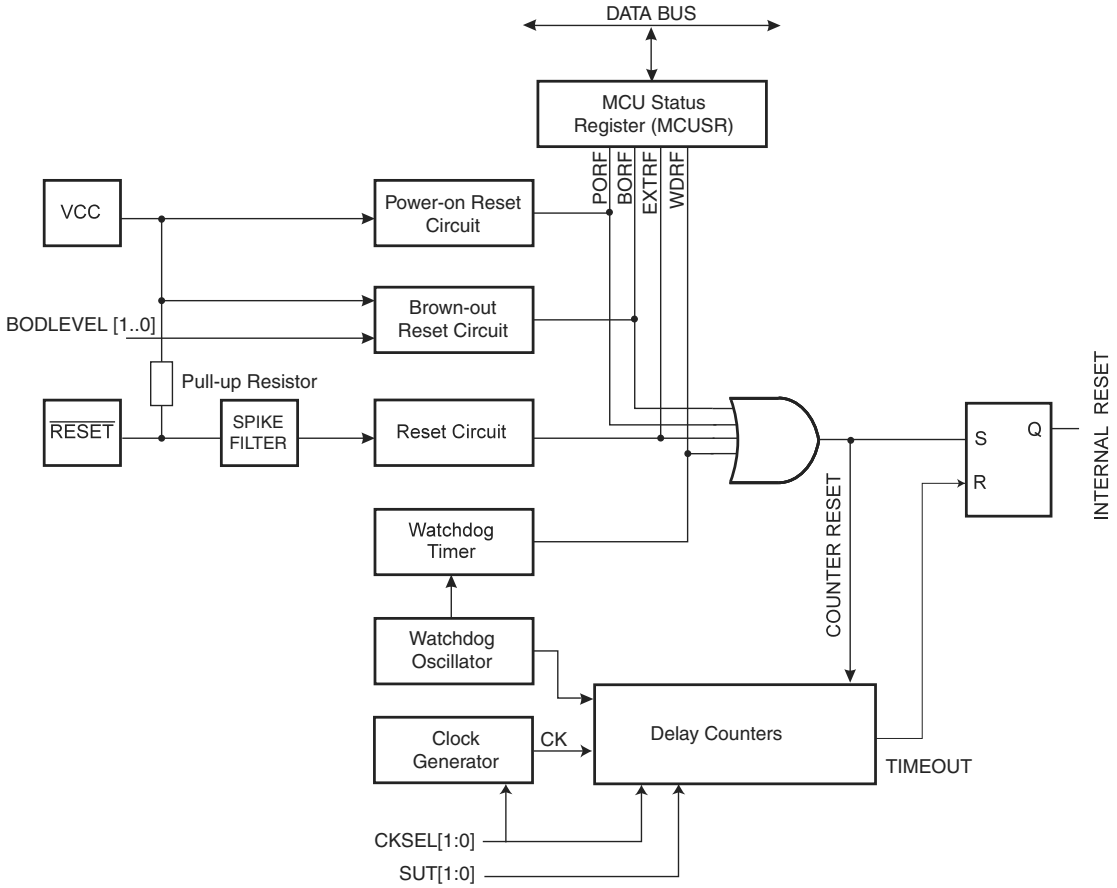
Bit	7	6	5	4	3	2	1	0	
0x0A (0x2A)	MSB							LSB	GPIOR0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

8. System Control and Reset

8.0.1 Resetting the AVR

During reset, all I/O Registers are set to their initial values, and the program starts execution from the Reset Vector. The instruction placed at the Reset Vector must be a RJMP – Relative Jump – instruction to the reset handling routine. If the program never enables an interrupt source, the Interrupt Vectors are not used, and regular program code can be placed at these locations. The circuit diagram in Figure 8-1 shows the reset logic. Electrical parameters of the reset circuitry are given in Table 19-4 on page 190.

Figure 8-1. Reset Logic



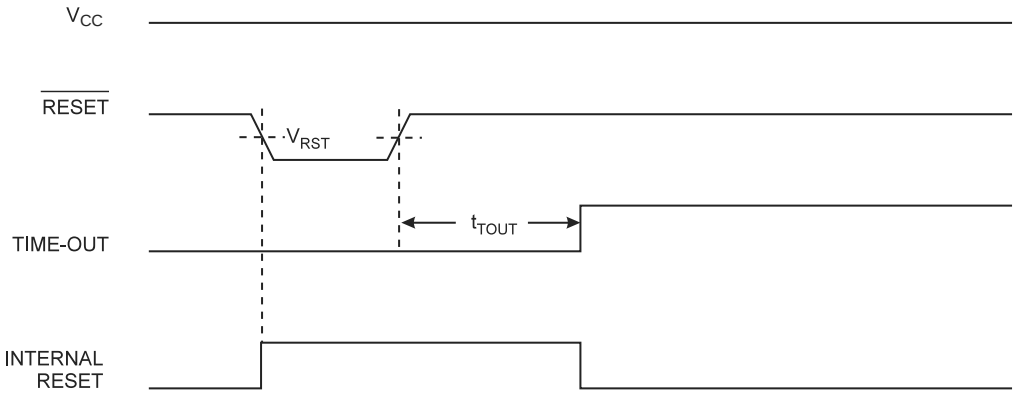
The I/O ports of the AVR are immediately reset to their initial state when a reset source goes active. This does not require any clock source to be running.

After all reset sources have gone inactive, a delay counter is invoked, stretching the internal reset. This allows the power to reach a stable level before normal operation starts. The time-out period of the delay counter is defined by the user through the SUT and CKSEL Fuses. The different selections for the delay period are presented in “Clock Sources” on page 25.

8.1.2 External Reset

An External Reset is generated by a low level on the $\overline{\text{RESET}}$ pin if enabled. Reset pulses longer than the minimum pulse width (see “System and Reset Characteristics” on page 190) will generate a reset, even if the clock is not running. Shorter pulses are not guaranteed to generate a reset. When the applied signal reaches the Reset Threshold Voltage – V_{RST} – on its positive edge, the delay counter starts the MCU after the Time-out period – t_{TOUT} – has expired.

Figure 8-4. External Reset During Operation



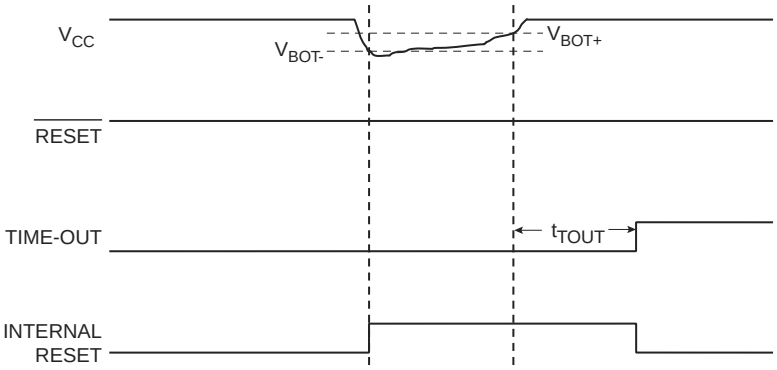
8.1.3 Brown-out Detection

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

When the BOD is enabled, and V_{CC} decreases to a value below the trigger level (V_{BOT-} in Figure 8-5), the Brown-out Reset is immediately activated. When V_{CC} increases above the trigger level (V_{BOT+} in Figure 8-5), the delay counter starts the MCU after the Time-out period t_{TOUT} has expired.

The BOD circuit will only detect a drop in V_{CC} if the voltage stays below the trigger level for longer than t_{BOD} given in “System and Reset Characteristics” on page 190.

Figure 8-5. Brown-out Reset During Operation



Control Register (MCUCR) define whether the external interrupt is activated on rising and/or falling edge of the INT0 pin or level sensed. Activity on the pin will cause an interrupt request even if INT0 is configured as an output. The corresponding interrupt of External Interrupt Request 0 is executed from the INT0 Interrupt Vector.

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

When the PCIE1 bit is set (one) and the I-bit in the Status Register (SREG) is set (one), pin change interrupt is enabled. Any change on any enabled PCINT7:0 or PCINT15:12 pin will cause an interrupt. The corresponding interrupt of Pin Change Interrupt Request is executed from the PCI Interrupt Vector. PCINT7:0 and PCINT15:12 pins are enabled individually by the PCMSK0 and PCMSK1 Register.

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

When the PCIE0 bit is set (one) and the I-bit in the Status Register (SREG) is set (one), pin change interrupt is enabled. Any change on any enabled PCINT11:8 pin will cause an interrupt. The corresponding interrupt of Pin Change Interrupt Request is executed from the PCI Interrupt Vector. PCINT11:8 pins are enabled individually by the PCMSK1 Register.

- **Bits 3:0 – Res: Reserved Bits**

These bits are reserved and will always read as zero.

9.3.3 GIFR – General Interrupt Flag Register

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

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

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

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

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

- **Bit 5 – PCIF: Pin Change Interrupt Flag**

When a logic change on any PCINT15 pin triggers an interrupt request, PCIF becomes set (one). If the I-bit in SREG and the PCIE bit in GIMSK are set (one), the MCU will jump to the corresponding Interrupt Vector. The flag is cleared when the interrupt routine is executed. Alternatively, the flag can be cleared by writing a logical one to it.

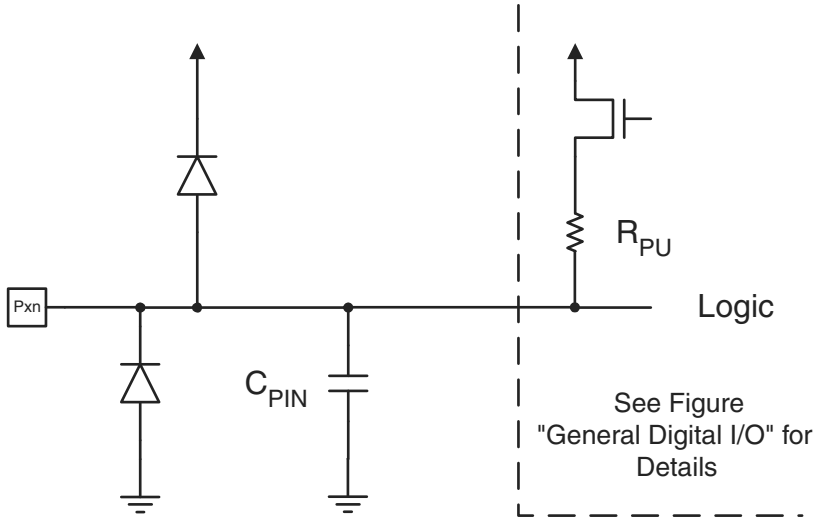
- **Bits 4:0 – Res: Reserved Bits**

These bits are reserved and will always read as zero.

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.

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	

- **Bit 1 – TOV0: Timer/Counter0 Overflow Flag**

The bit TOV0 is set when an overflow occurs in Timer/Counter0. TOV0 is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, TOV0 is cleared by writing a logic one to the flag. When the SREG I-bit, TOIE0 (Timer/Counter0 Overflow Interrupt Enable), and TOV0 are set, the Timer/Counter0 Overflow interrupt is executed.

- **Bit 0 – ICF0: Timer/Counter0, Input Capture Flag**

This flag is set when a capture event occurs on the ICP0 pin. When the Input Capture Register (ICR0) is set to be used as the TOP value, the ICF0 flag is set when the counter reaches the TOP value.

ICF0 is automatically cleared when the Input Capture Interrupt Vector is executed. Alternatively, ICF0 can be cleared by writing a logic one to its bit location.

Table 12-20 shows the COM1D1:0 bit functionality when the PWM1D and WGM11:10 bits are set to Phase and Frequency Correct PWM Mode.

Table 12-20. Compare Output Mode, Phase and Frequency Correct PWM Mode

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

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

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

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

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

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

12.12.4 TCCR1D – Timer/Counter1 Control Register D

Bit	7	6	5	4	3	2	1	0	
0x26 (0x46)	FPIE1	FPEN1	FPNC1	FPES1	FPAC1	FPF1	WGM11	WGM10	TCCR1D
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 – FPIE1: Fault Protection Interrupt Enable**

Setting this bit (to one) enables the Fault Protection Interrupt.

- **Bit 6 – FPEN1: Fault Protection Mode Enable**

Setting this bit (to one) activates the Fault Protection Mode.

- **Bit 5 – FPNC1: Fault Protection Noise Canceler**

Setting this bit activates the Fault Protection Noise Canceler. When the noise canceler is activated, the input from the Fault Protection Pin (INT0) is filtered. The filter function requires four successive equal valued samples of the INT0 pin for changing its output. The Fault Protection is therefore delayed by four Oscillator cycles when the noise canceler is enabled.

- **Bit 4 – FPES1: Fault Protection Edge Select**

This bit selects which edge on the Fault Protection pin (INT0) is used to trigger a fault event. When the FPES1 bit is written to zero, a falling (negative) edge is used as trigger, and when the FPES1 bit is written to one, a rising (positive) edge will trigger the fault.

```
SPITransfer_loop:
    sts    USICR,r16
    lds    r16, USISR
    sbrs   r16, USIOIF
    rjmp   SPITransfer_loop
    lds    r16,USIDR
    ret
```

The code is size optimized using only eight instructions (plus return). The code example assumes that the DO and USCK pins have been enabled as outputs in DDRA. The value stored in register r16 prior to the function is called is transferred to the slave device, and when the transfer is completed the data received from the slave is stored back into the register r16.

The second and third instructions clear the USI Counter Overflow Flag and the USI counter value. The fourth and fifth instructions set three-wire mode, positive edge clock, count at USITC strobe, and toggle USCK. The loop is repeated 16 times.

The following code demonstrates how to use the USI as an SPI master with maximum speed ($f_{SCK} = f_{CK}/2$):

```
SPITransfer_Fast:
    out    USIDR,r16
    ldi    r16,(1<<USIWM0)|(0<<USICS0)|(1<<USITC)
    ldi    r17,(1<<USIWM0)|(0<<USICS0)|(1<<USITC)|(1<<USICLK)

    out    USICR,r16 ; MSB
    out    USICR,r17
    out    USICR,r16
    out    USICR,r17
    out    USICR,r16
    out    USICR,r17
    out    USICR,r16
    out    USICR,r17
    out    USICR,r16
    out    USICR,r17
    out    USICR,r16
    out    USICR,r17
    out    USICR,r16
    out    USICR,r17
    out    USICR,r16
    out    USICR,r17
    out    USICR,r16 ; LSB
    out    USICR,r17

    in     r16,USIDR
ret
```

Table 13-2. Relations between the USICS1:0 and USICLK Setting (Continued)

USICS1	USICS0	USICLK	USI Data Register Clock Source	4-bit Counter Clock Source
1	1	0	External, negative edge	External, both edges
1	0	1	External, positive edge	Software clock strobe (USITC)
1	1	1	External, negative edge	Software clock strobe (USITC)

- **Bit 1 – USICLK: Clock Strobe**

Writing a one to this bit location strobes the USI Data Register to shift one step and the counter to increment by one, provided that the USICS1:0 bits are set to zero and by doing so the software clock strobe option is selected. The output will change immediately when the clock strobe is executed, i.e., in the same instruction cycle. The value shifted into the USI Data Register is sampled the previous instruction cycle. The bit will be read as zero.

When an external clock source is selected (USICS1 = 1), the USICLK function is changed from a clock strobe to a Clock Select Register. Setting the USICLK bit in this case will select the USITC strobe bit as clock source for the 4-bit counter (see Table 13-2).

- **Bit 0 – USITC: Toggle Clock Port Pin**

Writing a one to this bit location toggles the USCK/SCL value either from 0 to 1, or from 1 to 0. The toggling is independent of the setting in the Data Direction Register, but if the PORT value is to be shown on the pin the DDB2 must be set as output (to one). This feature allows easy clock generation when implementing master devices. The bit will be read as zero.

When an external clock source is selected (USICS1 = 1) and the USICLK bit is set to one, writing to the USITC strobe bit will directly clock the 4-bit counter. This allows an early detection of when the transfer is done when operating as a master device.

13.5.5 USIPP – USI Pin Position

Bit	7	6	5	4	3	2	1	0	
0x11 (0x31)	-	-	-	-	-	-	-	USIPOS	USIPP
Read/Write	R	R	R	R	R	R	R	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bits 7:1 – Res: Reserved Bits**

These bits are reserved and will always read as zero.

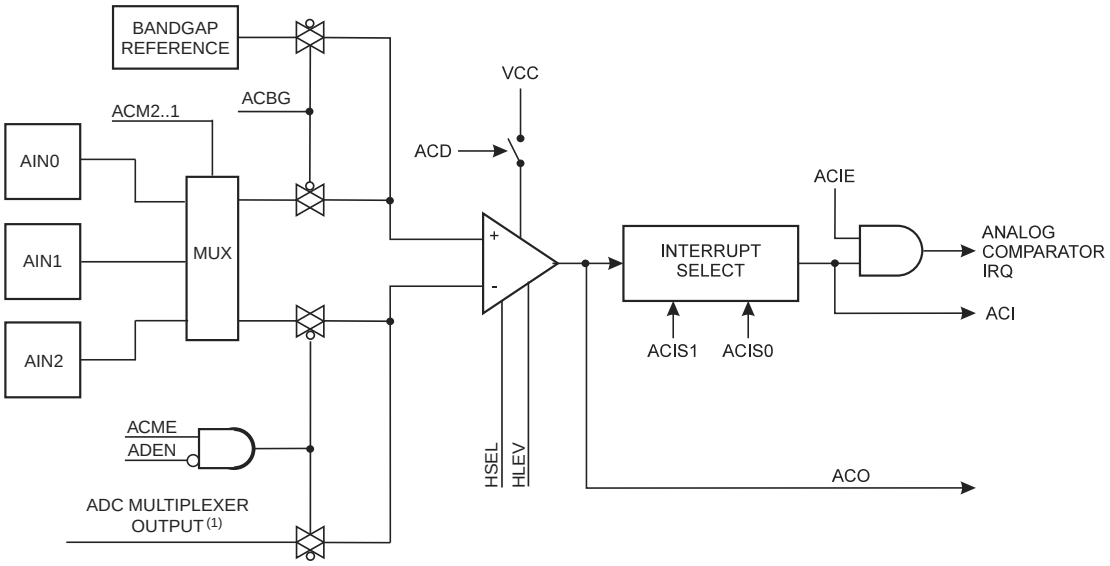
- **Bit 0 – USIPOS: USI Pin Position**

Setting this bit to one changes the USI pin position. As default pins PB2:PB0 are used for the USI pin functions, but when writing this bit to one the USIPOS bit is set the USI pin functions are on pins PA2:PA0.

14. AC – Analog Comparator

The analog comparator compares the input values on the selectable positive pin (AIN0, AIN1 or AIN2) and selectable negative pin (AIN0, AIN1 or AIN2). When the voltage on the positive pin is higher than the voltage on the negative pin, the Analog Comparator Output, ACO, is set. The comparator can trigger a separate interrupt, exclusive to the analog comparator. The user can select Interrupt triggering on comparator output rise, fall or toggle. A block diagram of the comparator and its surrounding logic is shown in Figure 14-1.

Figure 14-1. Analog Comparator Block Diagram



Notes: 1. See Table 14-1 on page 137.

See Figure 1-1 on page 2 and Table 10-3 on page 63 for Analog Comparator pin placement.

14.1 Analog Comparator Multiplexed Input

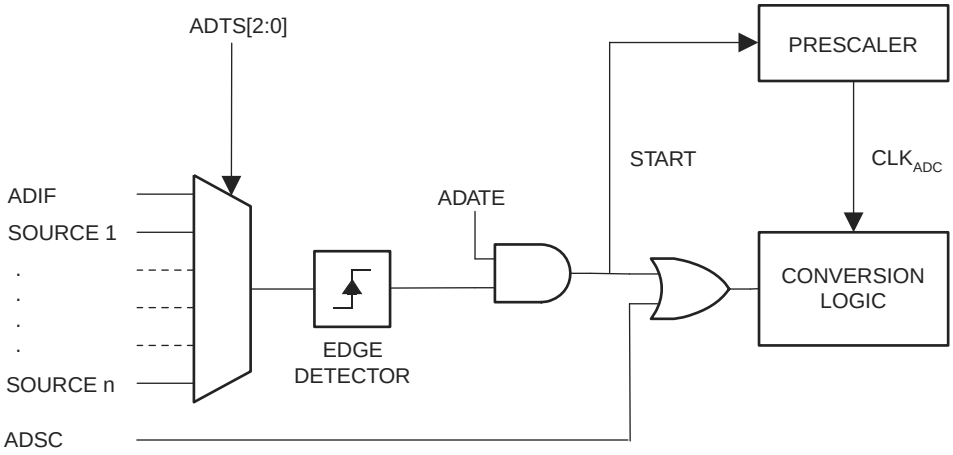
When the Analog to Digital Converter (ADC) is configurated as single ended input channel, it is possible to select any of the ADC10:0 pins to replace the negative input to the analog comparator. The ADC multiplexer is used to select this input, and consequently, the ADC must be switched off to utilize this feature. If the Analog Comparator Multiplexer Enable bit (ACME in ADCSRB) is set and the ADC is switched off (ADEN in ADCSRA is zero), MUX5:0 in ADMUX select the input pin to replace the negative input to the analog comparator, as shown in Table 14-1. If ACME is cleared or ADEN is set, either AIN0, AIN1 or AIN2 is applied to the negative input to the analog comparator.

Table 14-1. Analog Comparator Multiplexed Input

ACME	ADEN	MUX5:0	ACM2:0	Positive Input	Negative Input
0	x	xxxxxx	000	AIN0	AIN1
0	x	xxxxxx	001	AIN0	AIN2
0	x	xxxxxx	010	AIN1	AIN0
0	x	xxxxxx	011	AIN1	AIN2

conversion will not be started. If another positive edge occurs on the trigger signal during conversion, the edge will be ignored. Note that an Interrupt Flag will be set even if the specific interrupt is disabled or the Global Interrupt Enable bit in SREG is cleared. A conversion can thus be triggered without causing an interrupt. However, the Interrupt Flag must be cleared in order to trigger a new conversion at the next interrupt event.

Figure 15-2. ADC Auto Trigger Logic



Using the ADC Interrupt Flag as a trigger source makes the ADC start a new conversion as soon as the ongoing conversion has finished. The ADC then operates in Free Running mode, constantly sampling and updating the ADC Data Register. The first conversion must be started by writing a logical one to the ADSC bit in ADCSRA. In this mode the ADC will perform successive conversions independently of whether the ADC Interrupt Flag, ADIF is cleared or not.

If Auto Triggering is enabled, single conversions can be started by writing ADSC in ADCSRA to one. ADSC can also be used to determine if a conversion is in progress. The ADSC bit will be read as one during a conversion, independently of how the conversion was started.

15.5 Prescaling and Conversion Timing

By default, the successive approximation circuitry requires an input clock frequency between 50 kHz and 200 kHz to get maximum resolution. If a lower resolution than 10 bits is needed, the input clock frequency to the ADC can be higher than 200 kHz to get a higher sample rate.

The ADC module contains a prescaler, which generates an acceptable ADC clock frequency from any CPU frequency above 100 kHz. The prescaling is set by the ADPS bits in ADCSRA. The prescaler starts counting from the moment the ADC is switched on by setting the ADEN bit in ADCSRA. The prescaler keeps running for as long as the ADEN bit is set, and is continuously reset when ADEN is low.

See Figure 15-3 on page 146.

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

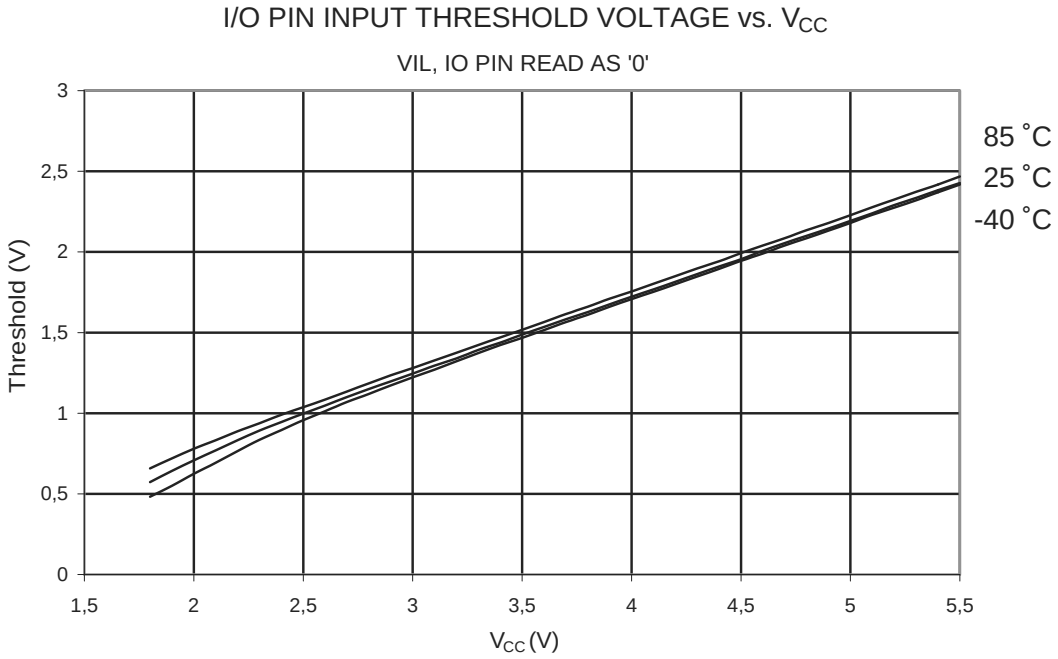
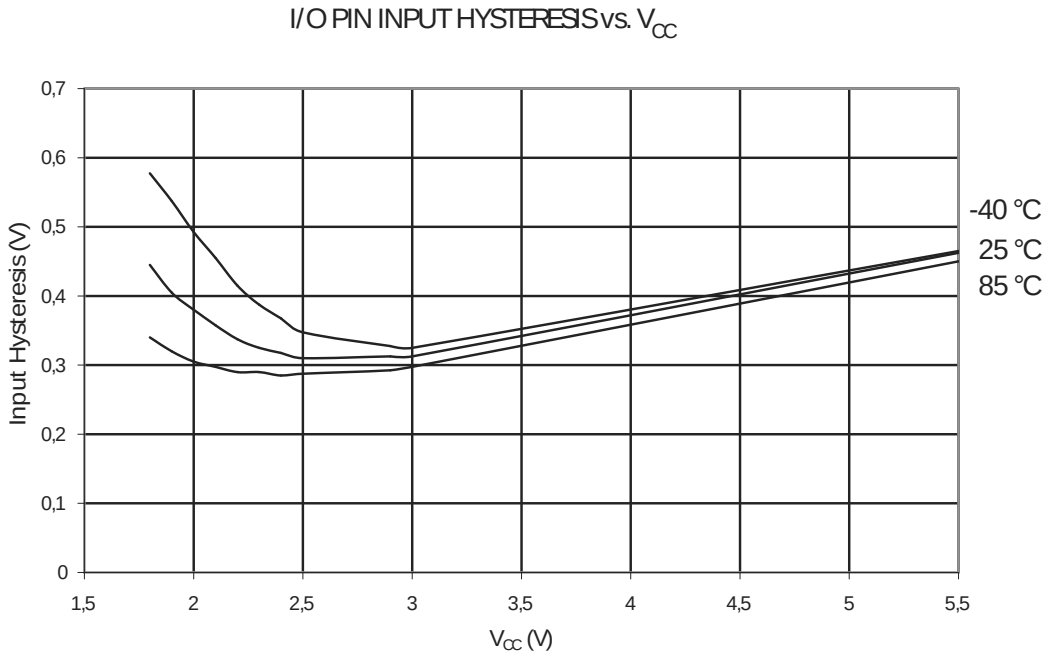
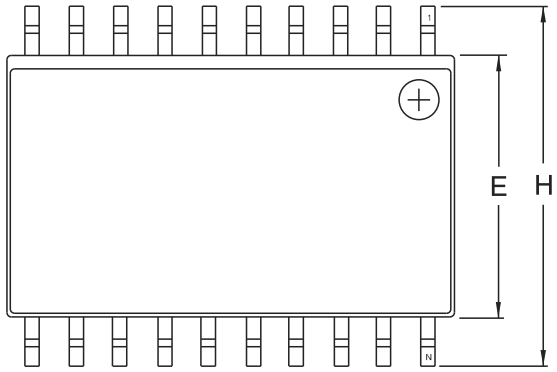


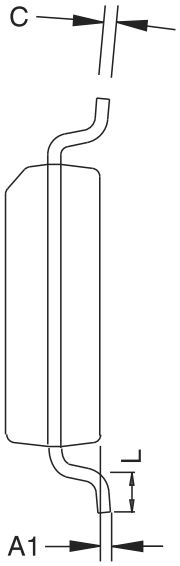
Figure 20-29. I/O Pin Input Hysteresis vs. V_{CC}



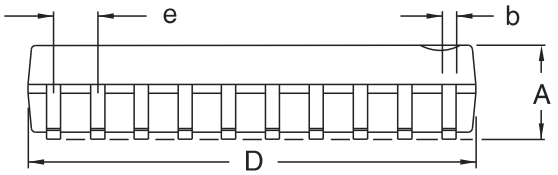
24.3 20S2



Top View



End View



Side View

COMMON DIMENSIONS
(Unit of Measure – mm)

SYMBOL	MIN	NOM	MAX	NOTE
A	2.35		2.65	
A1	0.10		0.30	
b	0.33		0.51	4
C	0.23		0.32	
D	12.60		13.00	1
E	7.40		7.60	2
H	10.00		10.65	
L	0.40		1.27	3
e	1.27 BSC			

Notes.
1. This drawing is for general information only; refer to JEDEC Drawing MS-013, Variation AC for additional information.
2. Dimension 'D' does not include mold Flash, protrusions or gate burrs. Mold Flash, protrusions and gate burrs shall not exceed 0.15 mm (0.006') per side.
3. Dimension 'E' does not include inter-lead Flash or protrusion. Inter-lead Flash and protrusions shall not exceed 0.25 mm (0.010') per side.
4. 'L' is the length of the terminal for soldering to a substrate.
5. The lead width 'b', as measured 0.36 mm (0.014') or greater above the seating plane, shall not exceed a maximum value of 0.61 mm (0.024') per side.

11/6/06

2325 Orchard Parkway San Jose, CA 95131	**TITLE** **20S2**, 20-lead, 0.300' Wide Body, Plastic Gull Wing Small Outline Package (SOIC)	**DRAWING NO.** 20S2	**REV.** B

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