

## Features

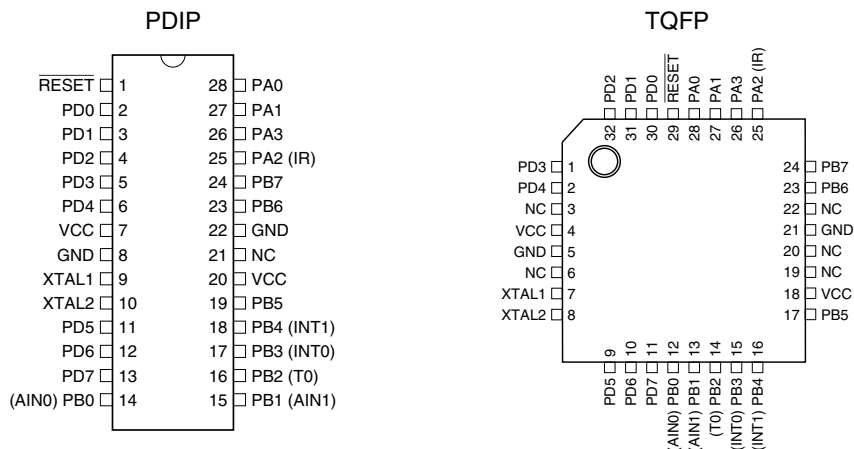
- Utilizes the AVR<sup>®</sup> RISC Architecture
- AVR - High-performance and Low-power RISC Architecture
  - 90 Powerful Instructions - Most Single Clock Cycle Execution
  - 32 x 8 General Purpose Working Registers
  - Up to 4 MIPS Throughput at 4 MHz
- Nonvolatile Program Memory
  - 2K Bytes of Flash Program Memory
  - Endurance: 1,000 Write/Erase Cycles
  - Programming Lock for Flash Program Data Security
- Peripheral Features
  - Interrupt and Wake-up on Low-level Input
  - One 8-bit Timer/Counter with Separate Prescaler
  - On-chip Analog Comparator
  - Programmable Watchdog Timer with On-chip Oscillator
  - Built-in High-current LED Driver with Programmable Modulation
- Special Microcontroller Features
  - Low-power, Idle and Power-down Modes
  - External and Internal Interrupt Sources
  - Power-on Reset Circuit with Programmable Start-up Time
  - Internal Calibrated RC Oscillator
- Power Consumption 2V at 1 MHz, 25°C
  - Active: TBD
  - Idle Mode: TBD
  - Power-down Mode: <1  $\mu$ A
- I/O and Packages
  - 11 Programmable I/O Lines, 8 Input Lines and a High-current LED Driver
  - 28-pin PDIP and 32-lead TQFP
- Operating Voltages
  - V<sub>CC</sub>: 1.8V - 5.5V (ATtiny28V)
  - V<sub>CC</sub>: 2.7V - 5.5V (ATtiny28L)
- Speed Grades
  - 0 - 1 MHz (ATtiny28V)
  - 0 - 4 MHz (ATtiny28L)

## Description

The ATtiny28 is a low-power CMOS 8-bit microcontroller based on the AVR RISC architecture. By executing powerful instructions in a single clock cycle, the ATtiny28 achieves throughputs approaching 1 MIPS per MHz allowing the system designer to

*(continued)*

## Pin Configuration



Rev. 1062C-05/00



8-bit AVR<sup>®</sup>  
Microcontroller  
with 2K Bytes  
Flash

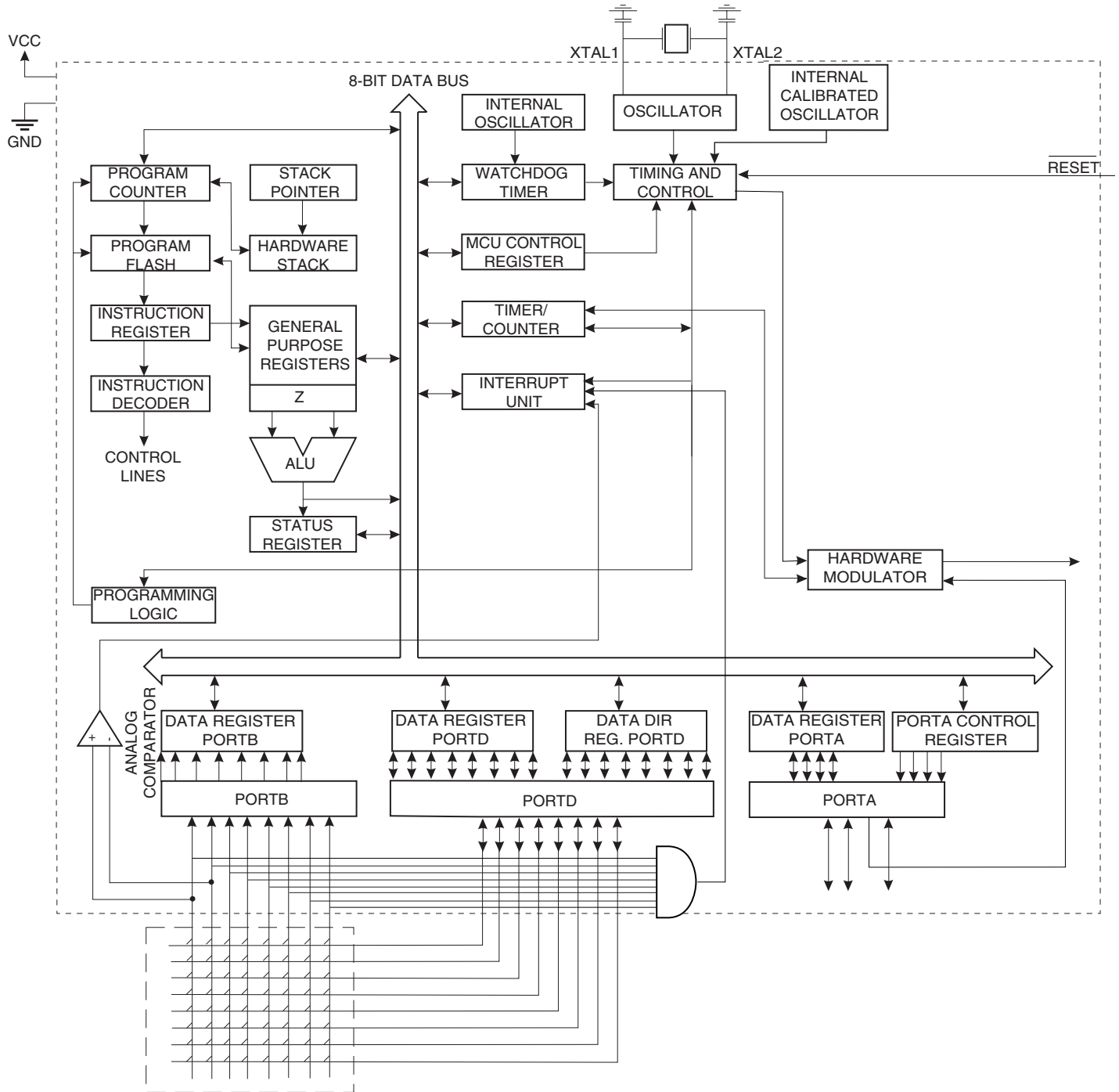
ATtiny28L  
ATtiny28V

Preliminary

optimize power consumption versus processing speed. The AVR core combines a rich instruction set with 32 general purpose working registers. All the 32 registers are directly connected to the Arithmetic Logic Unit (ALU), allowing two independent registers to be accessed in one single instruction executed in one clock cycle. The resulting architecture is more code efficient while achieving throughputs up to ten times faster than conventional CISC microcontrollers.

## Block Diagram

Figure 1. The ATtiny28 Block Diagram



The ATtiny28 provides the following features: 2K bytes of Flash, 11 general purpose I/O lines, 8 input lines, a high-current LED driver, 32 general purpose working registers, an 8-bit timer/counter, internal and external interrupts, programmable Watchdog Timer with internal oscillator, and two software selectable power saving modes. The Idle Mode stops the CPU while allowing the timer/counter, and interrupt system to continue functioning. The power-down mode saves the register contents but freezes the oscillator, disabling all other chip functions until the next interrupt or hardware reset. The wakeup or interrupt on low level input feature, enables the ATtiny28 to be highly responsive to external events, still featuring the lowest power consumption while in the power-down modes.

The device is manufactured using Atmel's high density nonvolatile memory technology. By combining an enhanced RISC 8-bit CPU with Flash on a monolithic chip, the Atmel ATtiny28 is a powerful microcontroller that provides a highly flexible and cost-effective solution to many embedded control applications. The ATtiny28 AVR is supported with a full suite of program and system development tools including: macro assemblers, program debugger/simulators, in-circuit emulators, and evaluation kits.

## Pin Descriptions

### VCC

Supply voltage pin

### GND

Ground pin

### Port A (PA3..PA0)

Port A is a 4-bit I/O port. PA2 is output only and can be used as a high-current LED driver. At  $V_{CC} = 1.8V$ , the PA2 output buffer can sink 25 mA. PA3, PA1 and PA0 are bidirectional I/O pins with internal pullups (selected for each bit). The port pins are tri-stated when a reset condition becomes active, even if the clock is not running.

### Port B (PB7..PB0)

Port B is an 8-bit input port with internal pullups (selected for all Port B pins). Port B pins that are externally pulled low will source current if the pullups are activated.

Port B also serves the functions of various special features of the ATtiny28 as listed on page 35. If any of the special features are enabled, the pullup(s) on the corresponding pin(s) is automatically disabled. The port pins are tri-stated when a reset condition becomes active, even if the clock is not running.

### Port D (PD7..PD0)

Port D is an 8-bit I/O port. Port pins can provide internal pullup resistors (selected for each bit). The port pins are tri-stated when a reset condition becomes active, even if the clock is not running.

### XTAL1

Input to the inverting oscillator amplifier and input to the internal clock operating circuit.

### XTAL2

Output from the inverting oscillator amplifier.

### RESET

Reset input. An external reset is generated by a low level on the  $\overline{\text{RESET}}$  pin. Reset pulses longer than 50 ns will generate a reset, even if the clock is not running. Shorter pulses are not guaranteed to generate a reset.

## Clock Options

The device has the following clock source options, selectable by flash fuse bits as shown:

**Table 1.** Device Clocking Option Select

| Clock Option                    | CKSEL3..0   |
|---------------------------------|-------------|
| External Crystal/Cer. Resonator | 1111 - 1010 |
| External Low-frequency Crystal  | 1001 - 1000 |
| External RC Oscillator          | 0111 - 0101 |
| Internal RC Oscillator          | 0100 - 0010 |
| External Clock                  | 0001 - 0000 |

Note: "1" means unprogrammed, "0" means programmed.

The various choices for each clocking option give different start-up times as shown in Table 5 on page 15.

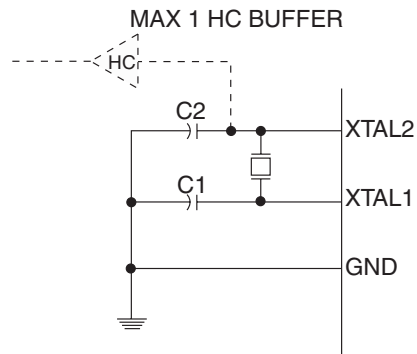
### Internal RC Oscillator

The internal RC oscillator option is an on-chip calibrated oscillator running at a nominal frequency of 1 MHz. If selected, the device can operate with no external components. The device is shipped with this option selected.

### Crystal Oscillator

XTAL1 and XTAL2 are input and output, respectively, of an inverting amplifier which can be configured for use as an on-chip oscillator, as shown in Figure 2. Either a quartz crystal or a ceramic resonator may be used. When the INTCAP fuse is programmed, internal load capacitors with typical values 50 pF are connected between XTAL1/XTAL2 and ground.

**Figure 2.** Oscillator Connections

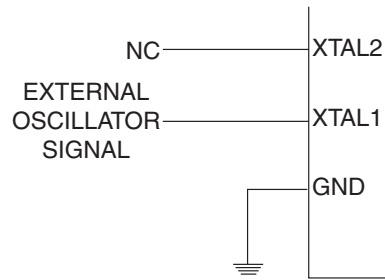


Note: When using the MCU Oscillator as a clock for an external device, an HC buffer should be connected as indicated in the figure.

### External Clock

To drive the device from an external clock source, XTAL2 should be left unconnected while XTAL1 is driven as shown in Figure 3.

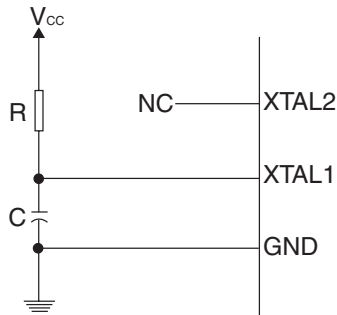
**Figure 3.** External Clock Drive Configuration



### External RC Oscillator

For timing insensitive applications, the external RC configuration shown in Figure 4 can be used. For details on how to choose R and C, see Table 26 on page 49.

**Figure 4.** External RC Configuration

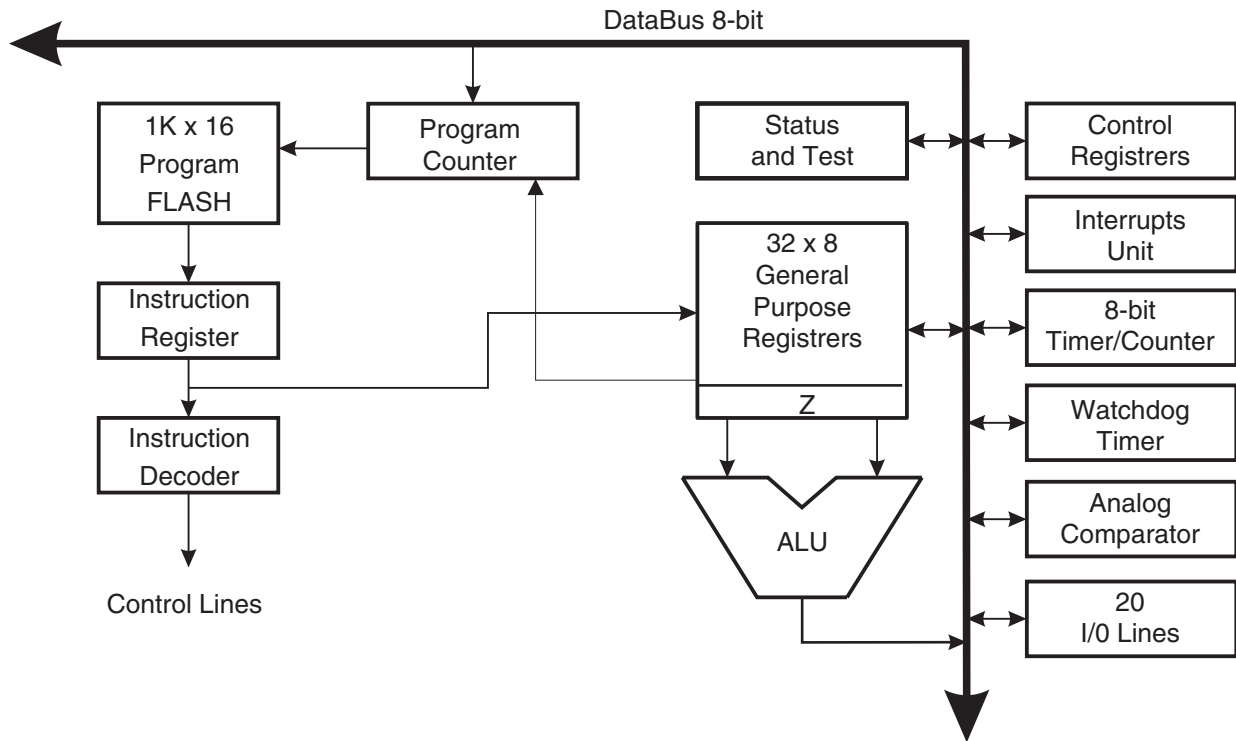


## Architectural Overview

The fast-access register file concept contains 32 x 8-bit general purpose working registers with a single clock cycle access time. This means that during one single clock cycle, one ALU (Arithmetic Logic Unit) operation is executed. 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.

Two of the 32 registers can be used as a 16-bit pointer for indirect memory access. This pointer is called the Z-pointer, and can address the register file and the Flash program memory.

**Figure 5.** The ATtiny28 AVR RISC Architecture



The ALU supports arithmetic and logic functions between registers or between a constant and a register. Single register operations are also executed in the ALU. Figure 5 shows the ATtiny28 AVR RISC microcontroller architecture. The AVR uses a Harvard architecture concept – with separate memories and buses for program and data memories. The program memory is accessed with a two stage pipeline. While one instruction is being executed, the next instruction is pre-fetched from the program memory. This concept enables instructions to be executed every clock cycle. The program memory is reprogrammable Flash memory.

With the relative jump and relative call instructions, the whole 1K address space is directly accessed. All AVR instructions have a single 16-bit word format, meaning that every program memory address contains a single 16-bit instruction.

During interrupts and subroutine calls, the return address program counter (PC) is stored on the stack. The stack is a 3 level deep hardware stack dedicated for subroutines and interrupts.

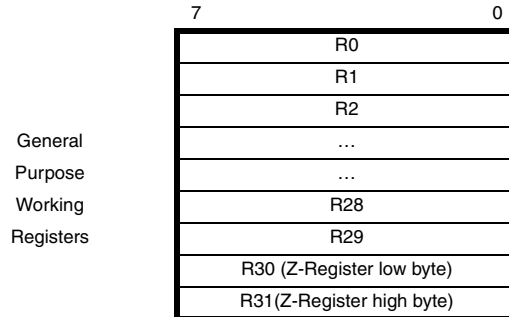
The I/O memory space contains 64 addresses for CPU peripheral functions as Control Registers, Timer/Counters, and other I/O functions. The memory spaces in the AVR architecture are all linear and regular memory maps.

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 the different interrupts have a separate interrupt vector in the interrupt vector table at the beginning of the program memory. The different interrupts have priority in accordance with their interrupt vector position. The lower the interrupt vector address, the higher the priority.

## General Purpose Register File

Figure 6 shows the structure of the 32 general purpose registers in the CPU.

**Figure 6.** AVR CPU General Purpose Working Registers



All the register operating instructions in the instruction set have direct and single cycle access to all registers. The only exception is the five constant arithmetic and logic instructions SBCI, SUBI, CPI, ANDI, and ORI between a constant and a register and the LDI instruction for load immediate constant data. These instructions apply to the second half of the registers in the register file – R16..R31. The general SBC, SUB, CP, AND, OR and all other operations between two registers or on a single register apply to the entire register file.

Registers 30 and 31 form a 16-bit pointer (the Z-pointer) which is used for indirect Flash memory and register file access. When the register file is accessed, the contents of R31 are discarded by the CPU.

## 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, ALU operations between registers in the register file are executed. The ALU operations are divided into three main categories – arithmetic, logic and bit-functions. Some microcontrollers in the AVR product family feature a hardware multiplier in the arithmetic part of the ALU.

## Downloadable Flash Program Memory

The ATtiny28 contains 2K bytes of on-chip Flash memory for program storage. Since all instructions are single 16-bit words, the Flash is organized as 1K x 16 words. The Flash memory has an endurance of at least 1000 write/erase cycles.

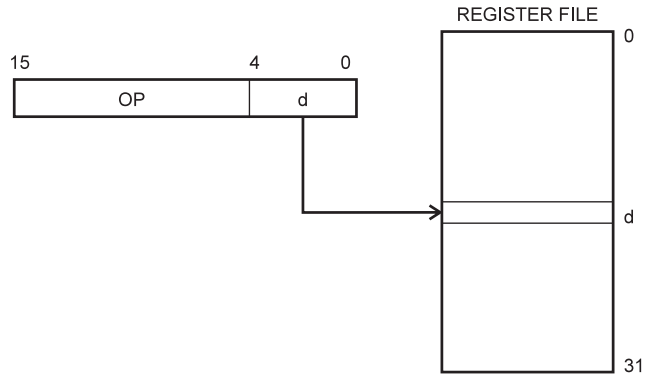
The ATtiny28 Program Counter is 10-bit wide, thus addressing the 1K words Flash program memory. See page 40 for a detailed description on Flash data downloading.

## Program and Data Addressing Modes

The ATtiny28 AVR RISC Microcontroller supports powerful and efficient addressing modes. This section describes the different addressing modes supported in the ATtiny28. In the figures, OP means the operation code part of the instruction word. To simplify, not all figures show the exact location of the addressing bits.

### Register Direct, Single Register Rd

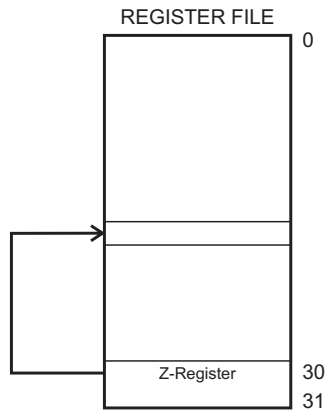
**Figure 7.** Direct Single Register Addressing



The operand is contained in register d (Rd).

### Register Indirect

**Figure 8.** Indirect Register Addressing

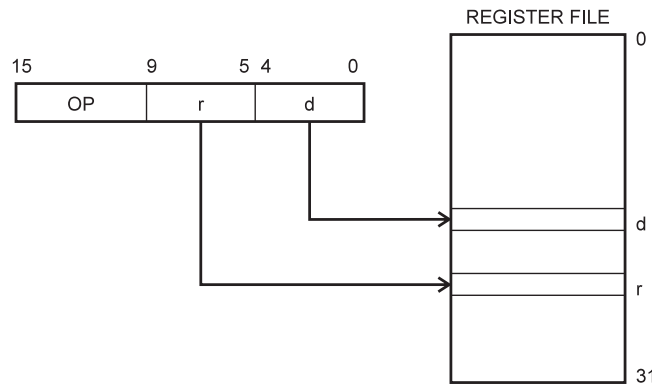


The register accessed is the one pointed to by the Z-register (R31, R30).



## Register Direct, Two Registers Rd and Rr

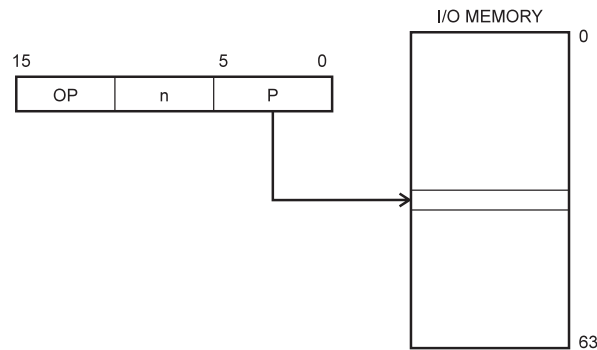
**Figure 9.** Direct Register Addressing, Two Registers



Operands are contained in register r (Rr) and d (Rd). The result is stored in register d (Rd).

## I/O Direct

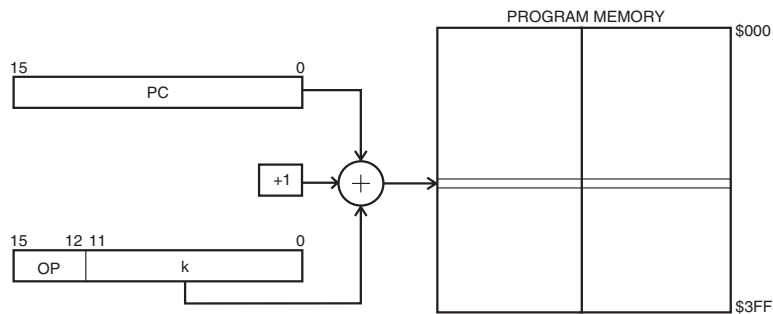
**Figure 10.** I/O Direct Addressing



Operand address is contained in 6 bits of the instruction word. n is the destination or source register address.

## Relative Program Addressing, RJMP and RCALL

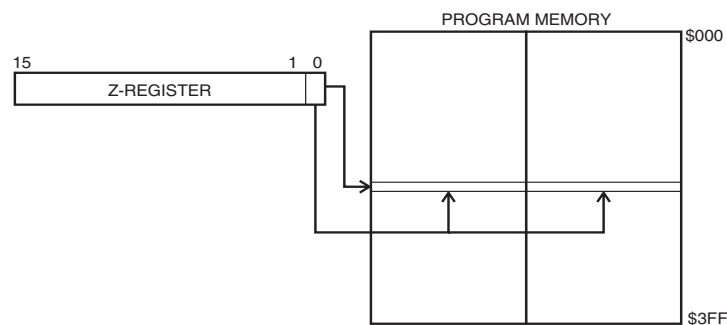
**Figure 11.** Relative Program Memory Addressing



Program execution continues at address  $PC + k + 1$ . The relative address  $k$  is  $-2048$  to  $2047$ .

## Constant Addressing Using the LPM Instruction

**Figure 12.** Code Memory Constant Addressing



Constant byte address is specified by the Z-register contents. The 15 MSBs select word address (0 - 1K), and LSB selects low byte if cleared (LSB = 0) or high byte if set (LSB = 1).

## Subroutine and Interrupt Hardware Stack

The ATtiny28 uses a 3 level deep hardware stack for subroutines and interrupts. The hardware stack is 10 bit wide and stores the Program Counter – PC – return address while subroutines and interrupts are executed.

RCALL instructions and interrupts push the PC return address onto stack level 0, and the data in the other stack levels 1-2 are pushed one level deeper in the stack. When a RET or RETI instruction is executed the returning PC is fetched from stack level 0, and the data in the other stack levels 1-2 are popped one level in the stack.

If more than 3 subsequent subroutine calls or interrupts are executed, the first values written to the stack are overwritten.

## Memory Access and Instruction Execution Timing

This section describes the general access timing concepts for instruction execution and internal memory access.

The AVR CPU is driven by the System Clock, directly generated from the external clock crystal for the chip. No internal clock division is used.

Figure 13 shows the parallel instruction fetches and instruction executions enabled by the Harvard architecture and the fast-access register file concept. This is the basic pipelining concept to obtain up to 1 MIPS per MHz with the corresponding unique results for functions per cost, functions per clocks, and functions per power-unit.

**Figure 13.** The Parallel Instruction Fetches and Instruction Executions

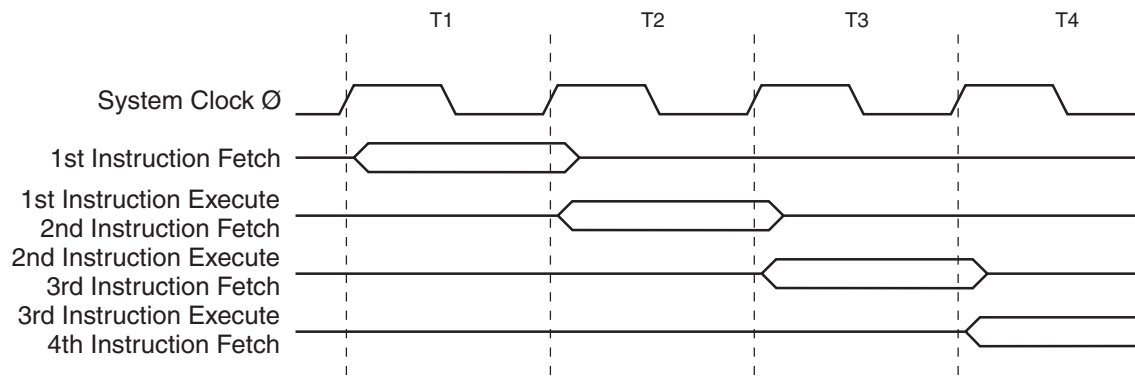
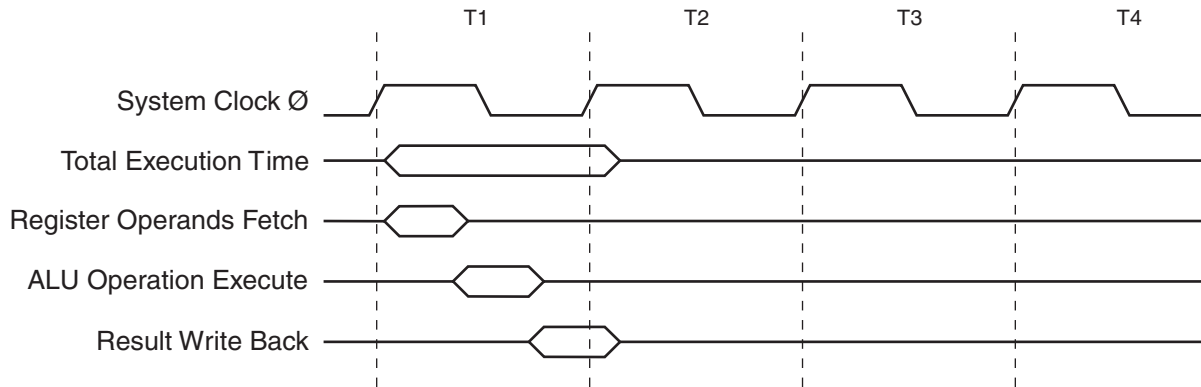


Figure 14 shows the internal timing concept for the register file. In a single clock cycle an ALU operation using two register operands is executed, and the result is stored back to the destination register.

**Figure 14.** Single Cycle ALU Operation





## I/O Memory

The I/O space definition of the ATtiny28 is shown in the following table:

**Table 2.** ATtiny28 I/O Space

| Address Hex | Name   | Function                                      |
|-------------|--------|---|
| \$3F        | SREG   | Status Register                               |
| \$1B        | PORTA  | Data Register, Port A                         |
| \$1A        | PACR   | Port A Control Register                       |
| \$19        | PINA   | Input Pins, Port A                            |
| \$16        | PINB   | Input Pins, Port B                            |
| \$12        | PORTD  | Data Register, Port D                         |
| \$11        | DDRD   | Data Direction Register, Port D               |
| \$10        | PIND   | Input Pins, Port D                            |
| \$08        | ACSR   | Analog Comparator control and Status Register |
| \$07        | MCUCS  | MCU Control and Status Register               |
| \$06        | ICR    | Interrupt Control Register                    |
| \$05        | IFR    | Interrupt Flag Register                       |
| \$04        | TCCR0  | Timer/Counter0 Control Register               |
| \$03        | TCNT0  | Timer/Counter0 (8-bit)                        |
| \$02        | MODCR  | Modulation Control Register                   |
| \$01        | WDTCSR | Watchdog Timer Control Register               |
| \$00        | OSCCAL | Oscillator Calibration Register               |

Note: Reserved and unused locations are not shown in the table.

All ATtiny28 I/O and peripherals are placed in the I/O space. The I/O locations are accessed by the IN and OUT instructions transferring data between the 32 general purpose working registers and the I/O space. I/O registers within the address range \$00 - \$1F are directly bit-accessible using the SBI and CBI instructions. In these registers, the value of single bits can be checked by using the SBIS and SBIC instructions. Refer to the instruction set chapter for more details.

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

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

### Status Register – SREG

The AVR status register – SREG – at I/O space location \$3F is defined as:

|               |     |     |     |     |     |     |     |     |      |
|---------------|-----|-----|-----|-----|-----|-----|-----|-----|------|
| Bit           | 7   | 6   | 5   | 4   | 3   | 2   | 1   | 0   |      |
| \$3F          | 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 (one) 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 (zero), 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.

- **Bit 6 - T: Bit Copy Storage**

The bit copy instructions BLD (Bit Load) and BST (Bit Store) use the T bit as source and 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. See the Instruction Set Description for detailed information.

- **Bit 4 - S: Sign Bit,  $S = N \oplus V$**

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 arithmetic. See the Instruction Set Description for detailed information.

- **Bit 2 - N: Negative Flag**

The negative flag N indicates a negative result from an arithmetical or logical operation. See the Instruction Set Description for detailed information.

- **Bit 1 - Z: Zero Flag**

The zero flag Z indicates a zero result from an arithmetical or logical operation. See the Instruction Set Description for detailed information.

- **Bit 0 - C: Carry Flag**

The carry flag C indicates a carry in an arithmetical or logical operation. See the Instruction Set Description for detailed information.

Note that the status register is not automatically stored when entering an interrupt routine and restored when returning from an interrupt routine. This must be handled by software.

## Reset and Interrupt Handling

The ATtiny28 provides 5 different interrupt sources. These interrupts and the reset vector, each have a separate program vector in the program memory space. All the interrupts are assigned to individual enable bits. In order to enable the interrupt, both the individual enable bit and the I-bit in the status register (SREG) must be set to one.

The lowest addresses in the program memory space are automatically defined as the Reset and Interrupt vectors. The complete list of vectors is shown in Table 3. The list also determines the priority levels of the different interrupts. The lower the address the higher is the priority level. RESET has the highest priority, and next is INT0 – the External Interrupt Request 0.

**Table 3.** Reset and Interrupt Vectors

| Vector No. | Program Address | Source       | Interrupt Definition                            |
|------------|-----------------|--------------|---|
| 1          | \$000           | RESET        | Hardware Pin, Power-on Reset and Watchdog Reset |
| 2          | \$001           | INT0         | External Interrupt Request 0                    |
| 3          | \$002           | INT1         | External Interrupt Request 1                    |
| 4          | \$003           | Input Pins   | Low Level Input on Port B                       |
| 5          | \$004           | TIMER0, OVFO | Timer/Counter0 Overflow                         |
| 6          | \$005           | ANA_COMP     | Analog Comparator                               |

The most typical and general program setup for the Reset and Interrupt Vector Addresses are:

| Address | Labels | Code           | Comments                   |
|---------|--------|----------------|----------------------------|
| \$000   |        | rjmp RESET     | ; Reset handler            |
| \$001   |        | rjmp EXT_INT0  | ; IRQ0 handler             |
| \$002   |        | rjmp EXT_INT1  | ; IRQ1 handler             |
| \$003   |        | rjmp LOW_LEVEL | ; Low level input handler  |
| \$004   |        | rjmp TIM0_OVF  | ; Timer0 overflow handle   |
| \$005   |        | rjmp ANA_COMP  | ; Analog Comparator handle |
| ;       |        |                |                            |
| \$006   | MAIN:  | <instr> xxx    | ; Main program start       |
| ...     | ...    | ...            | ...                        |

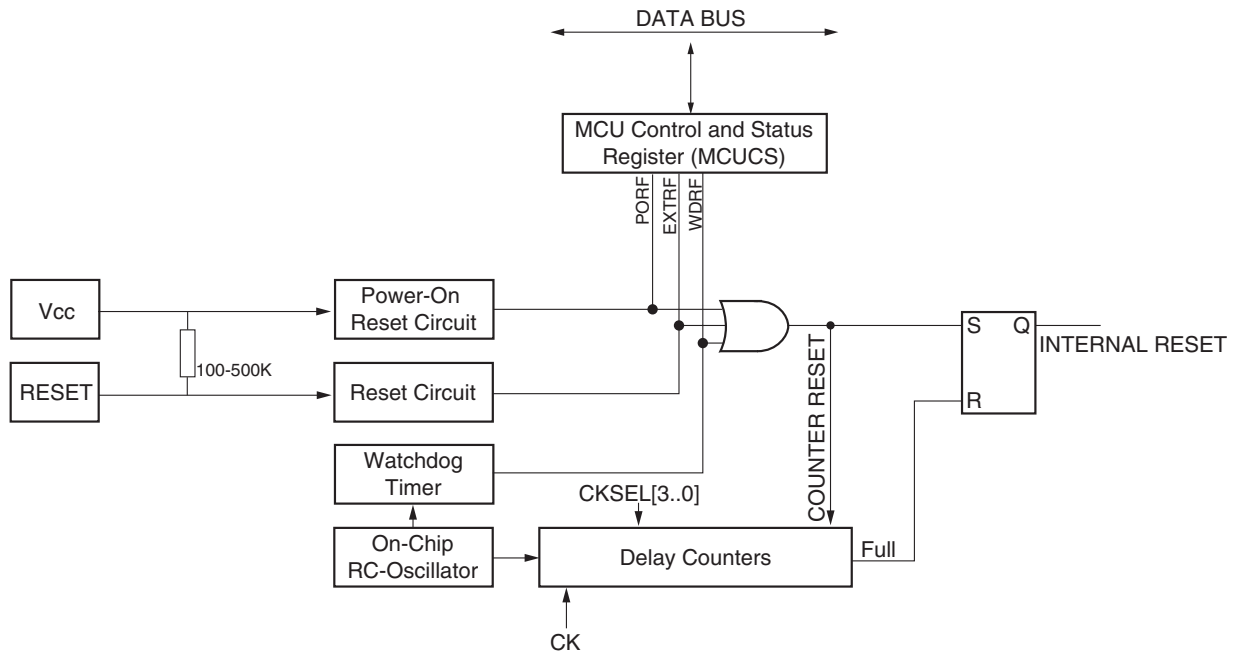
### Reset Sources

The ATtiny28 provides three sources of reset:

- Power-on Reset. The MCU is reset when the supply voltage is below the power-on reset threshold ( $V_{POT}$ ).
- External Reset. The MCU is reset when a low level is present on the **RESET** pin for more than 50 ns.
- Watchdog Reset. The MCU is reset when the Watchdog timer period expires and the Watchdog is enabled.

During reset, all I/O registers are then set to their initial values, and the program starts execution from address \$000. The instruction placed in address \$000 must be an 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 15 shows the reset logic. Table 4 defines the timing and electrical parameters of the reset circuitry.

Figure 15. Reset Logic



**Table 4.** Reset Characteristics

| Symbol                          | Parameter                                  | Min | Typ                 | Max | Units |
|---------------------------------|--|-----|---------------------|-----|-------|
| V <sub>POT</sub> <sup>(1)</sup> | Power-on Reset Threshold Voltage (rising)  | 1.0 | 1.4                 | 1.8 | V     |
|                                 | Power-on Reset Threshold Voltage (falling) | 0.4 | 0.6                 | 0.8 | V     |
| V <sub>RST</sub>                | RESET Pin Threshold Voltage                |     | 0.6 V <sub>CC</sub> |     | V     |

Notes: 1. The Power-on Reset will not work unless the supply voltage has been below V<sub>POT</sub> (falling).

**Table 5.** ATtiny28 Clock Options and Start-up Time

| CKSEL3..0 | Clock Source                                      | Start-up Time at 2.7V |
|-----------|---|-----------------------|
| 1111      | External Crystal/Ceramic Resonator <sup>(1)</sup> | 1K CK                 |
| 1110      | External Crystal/Ceramic Resonator <sup>(1)</sup> | 4.2 ms + 1K CK        |
| 1101      | External Crystal/Ceramic Resonator <sup>(1)</sup> | 67 ms + 1K CK         |
| 1100      | External Crystal/Ceramic Resonator                | 16K CK                |
| 1011      | External Crystal/Ceramic Resonator                | 4.2 ms + 16K CK       |
| 1010      | External Crystal/Ceramic Resonator                | 67 ms + 16K CK        |
| 1001      | External Low-frequency Crystal                    | 67 ms + 1K CK         |
| 1000      | External Low-frequency Crystal                    | 67 ms + 32K CK        |
| 0111      | External RC Oscillator                            | 6 CK                  |
| 0110      | External RC Oscillator                            | 4.2 ms + 6 CK         |
| 0101      | External RC Oscillator                            | 67 ms + 6 CK          |
| 0100      | Internal RC Oscillator                            | 6 CK                  |
| 0011      | Internal RC Oscillator                            | 4.2 ms + 6 CK         |
| 0010      | Internal RC Oscillator                            | 67 ms + 6 CK          |
| 0001      | External Clock                                    | 6 CK                  |
| 0000      | External Clock                                    | 4.2 ms + 6 CK         |

Note: 1. Due to limited no. of clock cycles in the start-up period, it is recommended to use Cer. Resonator

This table shows the start-up times from reset. From power-down mode, only the clock counting part of the start-up time is used. The watchdog oscillator is used for timing the real-time part of the start-up time. The number WDT oscillator cycles used for each time-out is shown in Table 6.

**Table 6.** Number of Watchdog Oscillator Cycles

| Time-out | Number of Cycles |
|----------|------------------|
| 4.2 ms   | 1K               |
| 67 ms    | 16K              |

The frequency of the watchdog oscillator is voltage depend as shown in the section “Typical Characteristics – Preliminary Data” on page 50.

The device is shipped with CKSEL = 0010.

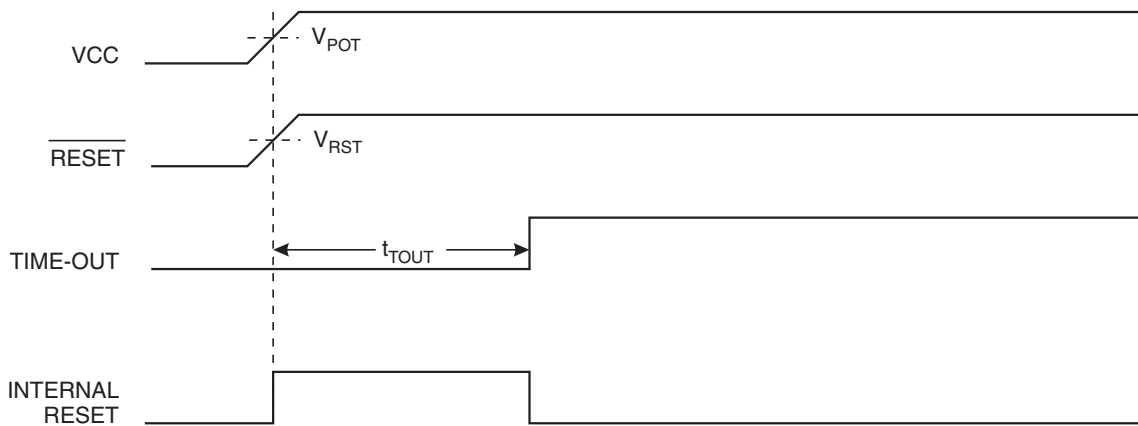
### Power-on Reset

A Power-on Reset (POR) pulse is generated by an on-chip detection circuit. The detection level is nominally 1.4V. The POR is activated whenever  $V_{CC}$  is below the detection level. The POR circuit can be used to trigger the start-up reset, as well as detect a failure in supply voltage.

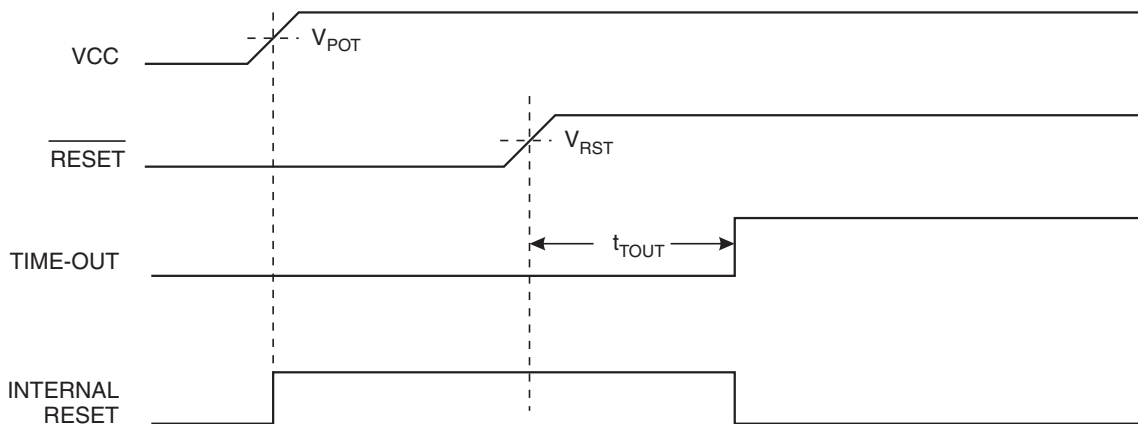
The Power-on Reset (POR) circuit ensures that the device is reset from power-on. Reaching the power-on reset threshold voltage invokes a delay counter, which determines the delay, for which the device is kept in RESET after  $V_{CC}$  rise. The time-out period of the delay counter can be defined by the user through the CKSEL fuses. The different selections for the delay period are presented in Table 5. The RESET signal is activated again, without any delay, when the  $V_{CC}$  decreases below detection level. See Figure 16.

If the built-in start-up delay is sufficient,  $\overline{\text{RESET}}$  can be connected to  $V_{CC}$  directly or via an external pull-up resistor. By holding the  $\overline{\text{RESET}}$  pin low for a period after  $V_{CC}$  has been applied, the Power-on Reset period can be extended. Refer to Figure 17 for a timing example on this.

**Figure 16.** MCU Start-up,  $\overline{\text{RESET}}$  Tied to  $V_{CC}$ .



**Figure 17.** MCU Start-up,  $\overline{\text{RESET}}$  Controlled Externally

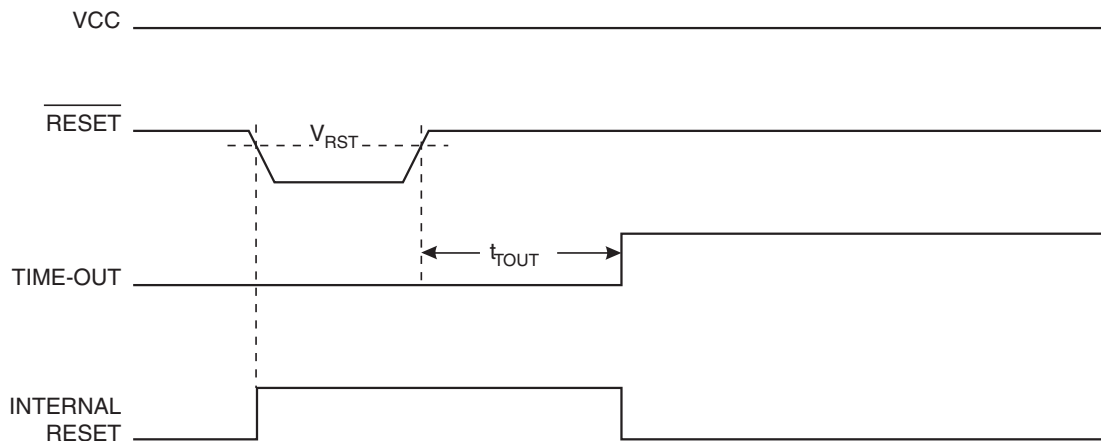




## External Reset

An external reset is generated by a low level on the  $\overline{\text{RESET}}$  pin. Reset pulses longer than 50ns will generate a reset, even if the clock is not running. Shorter pulses are not guaranteed to generate a reset. When the applied voltage reaches the Reset Threshold Voltage –  $V_{\text{RST}}$  on its positive edge, the delay timer starts the MCU after the Time-out period  $t_{\text{TOUT}}$  has expired.

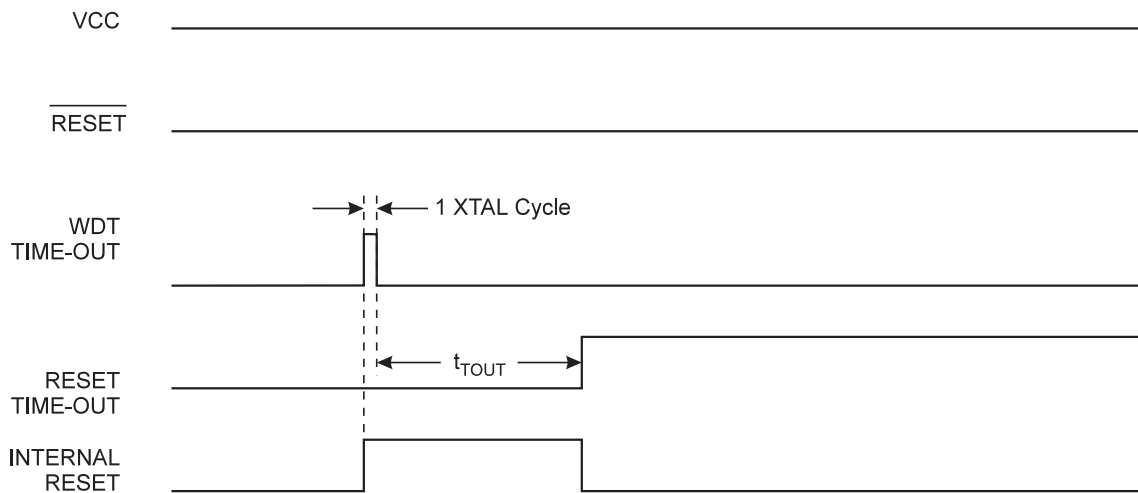
**Figure 18.** External Reset During Operation



## Watchdog Reset

When the Watchdog times out, it will generate a short reset pulse of 1 XTAL cycle duration. On the falling edge of this pulse, the delay timer starts counting the Time-out period  $t_{\text{TOUT}}$ . Refer to page 25 for details on operation of the Watchdog.

**Figure 19.** Watchdog Reset During Operation



## MCU Control and Status Register – MCUCS

The MCU Control and Status Register contains control and status bits for general MCU functions.

| Bit           | 7     | 6 | 5   | 4   | 3             | 2 | 1                   | 0    |       |
|---------------|-------|---|-----|-----|---------------|---|---------------------|------|-------|
| \$07          | PLUPB | - | SE  | SM  | WDRF          | - | EXTRF               | PORF | MCUCS |
| Read/Write    | R/W   | R | R/W | R/W | R/W           | R | R/W                 | R/W  |       |
| Initial value | 0     | 0 | 0   | 0   | See bit desc. | 0 | See bit description |      |       |

- **Bit 7 - PLUPB: Pull Up Enable Port B**

When the PLUPB bit is set (one), pullup resistors are enabled on all port B input pins. When PLUPB is cleared, the pullups are disabled. If any of the special functions of port B is enabled, the corresponding pullup(s) is disabled, independent of the value of PLUPB.

- **Bit 6 - Res: Reserved Bit**

This bit is a reserved bit in the ATtiny28 and always reads as zero.

- **Bit 5 - SE: Sleep Enable**

The SE bit must be set (one) to make the MCU enter the sleep mode when the SLEEP instruction is executed. To avoid the MCU entering the sleep mode unless it is the programmers purpose, it is recommended to set the Sleep Enable SE bit just before the execution of the SLEEP instruction.

- **Bit 4 - SM: Sleep Mode**

This bit selects between the two available sleep modes. When SM is cleared (zero), Idle Mode is selected as Sleep Mode. When SM is set (one), Power-down mode is selected as sleep mode. For details, refer to the paragraph “Sleep Modes” below.

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

This bit is set if a watchdog reset occurs. The bit is cleared by a power-on reset, or by writing a logic zero to the flag.

- **Bit 2 - Res: Reserved Bit**

This bit is a reserved bit in the ATtiny28 and always reads as zero.

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

This bit is set if an external reset occurs. The bit is cleared 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 cleared 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 clear the flag bits in MCUCS 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.

## Interrupt Handling

The ATtiny28 has one 8-bit Interrupt Control Register – ICR

When an interrupt occurs, the Global Interrupt Enable I-bit is cleared (zero) and all interrupts are disabled. The user software can set (one) the I-bit to enable nested interrupts. The I-bit is set (one) when a Return from Interrupt instruction – RETI – is executed.

When the Program Counter is vectored to the actual interrupt vector in order to execute the interrupt handling routine, hardware clears the corresponding flag that generated the interrupt. Some of the interrupt flags can also be cleared by writing a logic one to the flag bit position(s) to be cleared.

If an interrupt condition occurs when the corresponding interrupt enable bit is cleared (zero), the interrupt flag will be set and remembered until the interrupt is enabled, or the flag is cleared by software.

If one or more interrupt conditions occur when the global interrupt enable bit is cleared (zero), the corresponding interrupt flag(s) will be set and remembered until the global interrupt enable bit is set (one), and will be executed by order of priority.

Note that external level interrupt does not have a flag, and will only be remembered for as long as the interrupt condition is active.

Note that the status register is not automatically stored when entering an interrupt routine and restored when returning from an interrupt routine. This must be handled by software.

## Interrupt Response Time

The interrupt execution response for all the enabled AVR interrupts is 4 clock cycles minimum. After 4 clock cycles the program vector address for the actual interrupt handling routine is executed. During this 4 clock cycle period, the Program Counter (10 bits) is pushed onto the Stack. The vector is normally a relative jump to the interrupt routine, and this jump takes 2 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 4 clock cycles.

A return from an interrupt handling routine takes 4 clock cycles. During these 4 clock cycles, the Program Counter (10 bits) is popped back from the Stack, and the I flag in SREG is set. When AVR exits from an interrupt, it will always return to the main program and execute one more instruction before any pending interrupt is served.

## Interrupt Control Register – ICR

| Bit           | 7    | 6    | 5    | 4     | 3     | 2     | 1     | 0     |     |
|---------------|------|------|------|-------|-------|-------|-------|-------|-----|
| \$06          | INT1 | INT0 | LLIE | TOIE0 | ISC11 | ISC10 | ISC01 | ISC00 | ICR |
| 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 - INT1: External Interrupt Request 1 Enable**

When the INT1 bit is set (one) and I-bit in the Status Register (SREG) is set (one), the external pin interrupt 1 is enabled. The interrupt Sense Control1 bits 1/0 (ISC11 and ISC10) defines whether the external interrupt is activated on rising or falling edge, on pin change, or low level of the INT1 pin. The corresponding interrupt of External Interrupt Request 1 is executed from program memory address \$002. See also “External Interrupts”.

- **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 0 is enabled. The interrupt Sense Control 0 bits 1/0 (ISC01 and ISC00) defines whether the external interrupt is activated on rising or falling edge, on pin change, or low level of the INT0 pin. The corresponding interrupt of External Interrupt Request 0 is executed from program memory address \$001. See also “External Interrupts”.

- **Bit 5 - LLIE: Low Level Input Interrupt Enable**

When the LLIE is set (one) and the I-bit in the status register (SREG) is set (one), the interrupt on low level input is activated. Any of the port B pins pulled low will then cause an interrupt. However, if any port B pins are used for other special features, these pins will not trigger the interrupt. The corresponding interrupt of Low Level Input Interrupt Request is executed from program memory address \$003. See also “Low Level Input Interrupt”.

- **Bit 4 - TOIE0: Timer/Counter0 Overflow Interrupt Enable**

When the TOIE0 bit is set (one) and the I-bit in the Status Register is set (one), the Timer/Counter0 Overflow Interrupt is enabled. The corresponding interrupt (at vector \$004) is executed if an overflow in Timer/Counter0 occurs, i.e., when the TOV0 bit is set in the Interrupt Flag Register – IFR.

- **Bit 3,2 - ISC11, ISC10: Interrupt Sense Control 1 bit 1 and bit 0**

The External Interrupt 1 is activated by the external pin INT1 if the SREG I-flag and the corresponding interrupt enable is set. The level and edges on the external INT1 pin that activate the interrupt are defined in the following table:

**Table 7.** Interrupt 1 Sense Control

| ISC11 | ISC10 | Description  |
|-------|-------|--|
| 0     | 0     | The low level of INT1 generates an interrupt request.    |
| 0     | 1     | Any change on INT1 generates an interrupt request.       |
| 1     | 0     | The falling edge of INT1 generates an interrupt request. |
| 1     | 1     | The rising edge of INT1 generates an interrupt request.  |

Note: When changing the ISC11/ISC10 bits, INT1 must be disabled by clearing its Interrupt Enable bit. Otherwise an interrupt can occur when the bits are changed.

• **Bit 1,0 - ISC01, ISC00: Interrupt Sense Control 0 bit 1 and bit 0**

The External Interrupt 0 is activated by the external pin INT0 if the SREG I-flag and the corresponding interrupt enable is set. The level and edges on the external INT0 pin that activate the interrupt are defined in the following table:

**Table 8.** Interrupt 0 Sense Control

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

Note: When changing the ISC01/ISC00 bits, INT0 must be disabled by clearing its Interrupt Enable bit. Otherwise an interrupt can occur when the bits are changed.

The value on the INT pins are sampled before detecting edges. If edge interrupt is selected, pulses that last longer than one CPU 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. If enabled, a level triggered interrupt will generate an interrupt request as long as the pin is held low.

**Interrupt Flag Register – IFR**

| Bit           | 7     | 6     | 5 | 4    | 3 | 2 | 1 | 0 |     |
|---------------|-------|-------|---|------|---|---|---|---|-----|
| \$05          | INTF1 | INTF0 | - | TOV0 | - | - | - | - | IFR |
| Read/Write    | R/W   | R/W   | R | R/W  | R | R | R | R |     |
| Initial Value | 0     | 0     | 0 | 0    | 0 | 0 | 0 | 0 |     |

• **Bit 7 - INTF1: External Interrupt Flag1**

When an event on the INT1 pin triggers an interrupt request, INTF1 becomes set (one). If the I-bit in SREG and the INT1 bit in ICR are set (one), the MCU will jump to the interrupt vector at address \$002. The flag is cleared when the interrupt routine is executed. Alternatively, the flag can be cleared by writing a logical one to it.

• **Bit 6 - INTF0: External Interrupt Flag0**

When an event on the INT0 pin triggers an interrupt request, INTF0 becomes set (one). If the I-bit in SREG and the INT0 bit in ICR are set (one), the MCU will jump to the interrupt vector at address \$001. The flag is cleared when the interrupt routine is executed. Alternatively, the flag can be cleared by writing a logical one to it.

• **Bit 5 - Res: Reserved bit**

This bit is a reserved bit in the ATtiny28 and always reads as zero

• **Bit 4 - TOV0: Timer/Counter0 Overflow Flag**

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

• **Bit 3..0 - Res: Reserved bits**

These bits are reserved bits in the ATtiny28, and always read as zero.

Note: One should not try to use the SBI (set Bit in I/O Register) instruction to clear individual flags in the IFR Register. This will result in clearing all the flags in the register, because the register is first read, then modified and finally written, thus writing ones to all set flags. Using the CBI (Clear Bit in I/O Register) instruction on IFR will result in clearing all bits apart from the specified bit.

**External Interrupt**

The external interrupt is triggered by the INT pins. Observe that, if enabled, the interrupt will trigger even if the INT pin is configured as an output. This feature provides a way of generating a software interrupt. The external interrupt can be triggered by a falling or rising edge, a pin change, or a low level. This is set up as indicated in the specification for the Interrupt Control Register – ICR. When the external interrupt is enabled and is configured as level triggered, the interrupt will trigger as long as the pin is held low.

The external interrupt is set up as described in the specification for the Interrupt Control Register – ICR.

### **Low Level Input Interrupt**

The low level interrupt is triggered by setting any of the port B pins low. However, if any port B pins are used for other special features, these pins will not trigger the interrupt. For example, if the analog comparator is enabled, a low level on PB0 or PB1 will not cause an interrupt. This is also the case for the special functions T0, INT0 and INT1. If low level interrupt is selected, the low level must be held until the completion of the currently executing instruction to generate an interrupt. When this interrupt is enabled, the interrupt will trigger as long as any of the port B pins are held low.

### **Sleep Modes**

To enter the sleep modes, the SE bit in MCUCS must be set (one) and a SLEEP instruction must be executed. The SM bit in the MCUCS register selects which sleep mode (Idle or Power-down) will be activated by the SLEEP instruction. If an enabled interrupt occurs while the MCU is in a sleep mode, the MCU awakes. The CPU is then halted for 4 cycles, it executes the interrupt routine, and resumes execution from the instruction following SLEEP. The contents of the register file and I/O memory are unaltered. If a reset occurs during sleep mode, the MCU wakes up and executes from the Reset vector.

#### **Idle Mode**

When the SM bit is cleared (zero), the SLEEP instruction forces the MCU into the Idle Mode stopping the CPU but allowing Timer/Counters, Watchdog and the interrupt system to continue operating. This enables the MCU to wake up from external triggered interrupts as well as internal ones like Timer Overflow interrupt and watchdog reset. 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. Note that the ACD-bit is set by default.

#### **Power-down Mode**

When the SM bit is set (one), the SLEEP instruction forces the MCU into the Power-down Mode. In this mode, the external oscillator is stopped, while the external interrupts and the Watchdog (if enabled) continue operating. Only an external reset, a watchdog reset (if enabled), or an external level interrupt can wake up the MCU.

Note that if a level triggered interrupt is used for wake-up from Power-down Mode, the changed level must be held for some time to wake up the MCU. This makes the MCU less sensitive to noise. The wake-up period is equal to the clock-counting part of the reset period (see Table 5). The MCU will wake up from power-down if the input has the required level for two watchdog oscillator cycles. If the wake-up period is shorter than two watchdog oscillator cycles, the MCU will wake up if the input has the required level for the duration of the wake-up period. If the wake-up condition disappears before the wake-up period has expired, the MCU will wake up from power-down without executing the corresponding interrupt. The period of the watchdog oscillator is 2.7 $\mu$ s (nominal) at 3.0V and 25C. The frequency of the watchdog oscillator is voltage dependent as shown in the section “Typical Characteristics – Preliminary Data” on page 50.

When waking up from the Power-down Mode, there is a delay from the wake-up condition occurs until the wake-up becomes effective. This allows the clock to restart and become stable after having been stopped.

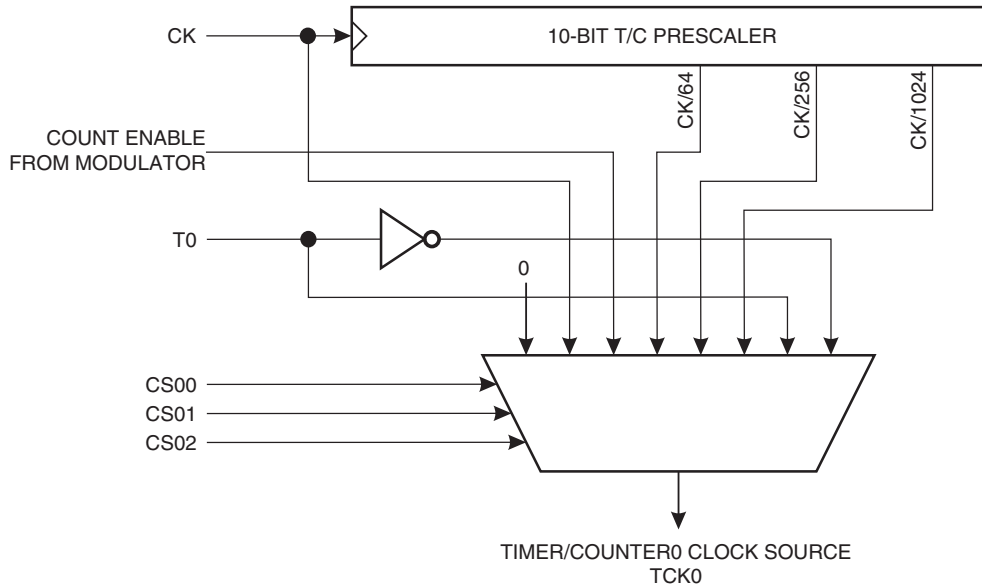
### **Timer/Counter0**

The ATtiny28 provides one general purpose 8-bit Timer/Counter – Timer/Counter0. The Timer/Counter0 has prescaling selection from the 10-bit prescaling timer. The Timer/Counter0 can either be used as a timer with an internal clock time-base or as a counter with an external pin connection that triggers the counting.

## Timer/Counter Prescaler

Figure 20 shows the Timer/Counter prescaler.

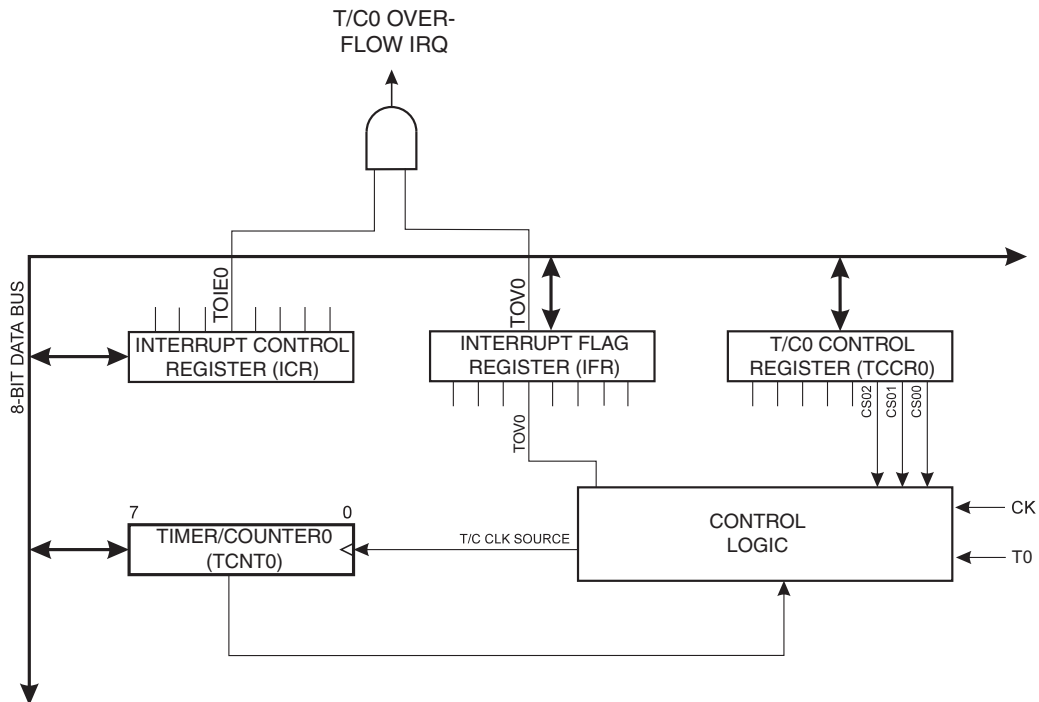
**Figure 20.** Timer/Counter0 Prescaler



The four different prescaled selections are: The hardware modulator period, CK/64, CK/256 and CK/1028 where CK is the oscillator clock. CK, external source and stop, can also be selected as clock sources.

Figure 21 shows the block diagram for Timer/Counter0.

**Figure 21.** Timer/Counter0 Block Diagram



The 8-bit Timer/Counter0 can select clock source from CK, prescaled CK, or an external pin. In addition, it can be stopped as described in the specification for the Timer/Counter0 Control Register – TCCR0. The overflow status flag is found in the Interrupt Flag Register – IFR. Control signals are found in the Timer/Counter0 Control Register – TCCR0. The interrupt enable/disable setting for Timer/Counter0 is found in the Interrupt Control Register – ICR.

When Timer/Counter0 is externally clocked, the external signal is synchronized with the oscillator frequency of the CPU. To ensure proper sampling of the external clock, the minimum time between two external clock transitions must be at least one internal CPU clock period. The external clock signal is sampled on the rising edge of the internal CPU clock.

The 8-bit Timer/Counter0 features both a high resolution and a high accuracy usage with the lower prescaling opportunities. Similarly, the high prescaling opportunities make the Timer/Counter0 useful for lower speed functions or exact timing functions with infrequent actions.

### Timer/Counter0 Control Register – TCCR0

|               |      |   |   |       |       |      |      |      |       |
|---------------|------|---|---|-------|-------|------|------|------|-------|
| Bit           | 7    | 6 | 5 | 4     | 3     | 2    | 1    | 0    |       |
| \$04          | FOV0 | - | - | OOM01 | OOM00 | CS02 | CS01 | CS00 | TCCR0 |
| Read/Write    | R/W  | R | R | R/W   | R/W   | R/W  | R/W  | R/W  |       |
| Initial value | 0    | 0 | 0 | 0     | 0     | 0    | 0    | 0    |       |

#### Bit 7 - FOV0: Force Overflow

Writing a logical one to this bit, forces a change on the overflow output pin PA2 according to the values already set in OOM01 and OOM00. If the OOM01 and OOM00 bits are written in the same cycle as FOV0, the new settings will not take effect until the next overflow or forced overflow occurs. The Force Overflow bit can be used to change the output pin without waiting for an overflow in the timer. The automatic action programmed in OOM01 and OOM00 happens as if an overflow had occurred, but no interrupt is generated. The FOV0 bit will always read as zero, and writing a zero to this bit has no effect.

#### Bit 6, 5 - Res: Reserved bits

These bits are reserved bits in the ATtiny28 and always read as zero

#### Bit 4, 3 - OOM01, OOM00: Overflow Output Mode, Bits 1 and 0

The OOM01 and OOM00 control bits determine any output pin action following an overflow or a forced overflow in Timer/Counter0. Any output pin actions affect pin PA2. The control configuration is shown in Table 9.

**Table 9.** Overflow Output Mode Select

| OOM01 | OOM00 | Description                                     |
|-------|-------|---|
| 0     | 0     | Timer/Counter0 disconnected form output pin PA2 |
| 0     | 1     | Toggle the PA2 output line                      |
| 1     | 0     | Clear the PA2 output line to zero               |
| 1     | 1     | Set the PA2 output line to one                  |

**Bits 2,1,0 - CS02, CS01, CS00: Clock Select0, bit 2,1 and 0**

The Clock Select0 bits 2,1 and 0 define the prescaling source of Timer0.

**Table 10.** Clock 0 Prescale Select

| CS02 | CS01 | CS00 | Description                          |
|------|------|------|--------------------------------------|
| 0    | 0    | 0    | Stop, the Timer/Counter0 is stopped. |
| 0    | 0    | 1    | CK                                   |
| 0    | 1    | 0    | Modulator Period                     |
| 0    | 1    | 1    | CK/64                                |
| 1    | 0    | 0    | CK/256                               |
| 1    | 0    | 1    | CK/1024                              |
| 1    | 1    | 0    | External Pin T0, falling edge        |
| 1    | 1    | 1    | External Pin T0, rising edge         |

The Stop condition provides a Timer Enable/Disable function. The CK down divided modes are scaled directly from the CK oscillator clock. If the external pin modes are used for Timer/Counter0, transitions on PB2/(T0) will clock the counter even if the pin is configured as an output. This feature can give the user SW control of the counting.

**Timer Counter 0 – TCNT0**

|               |     |     |     |     |     |     |     |     |       |
|---------------|-----|-----|-----|-----|-----|-----|-----|-----|-------|
| Bit           | 7   | 6   | 5   | 4   | 3   | 2   | 1   | 0   |       |
| \$03          | MSB |     |     |     |     |     |     | LSB | TCNT0 |
| 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 Timer/Counter0 is realized as an up-counter with read and write access. If the Timer/Counter0 is written and a clock source is present, the Timer/Counter0 continues counting in the timer clock cycle following the write operation.

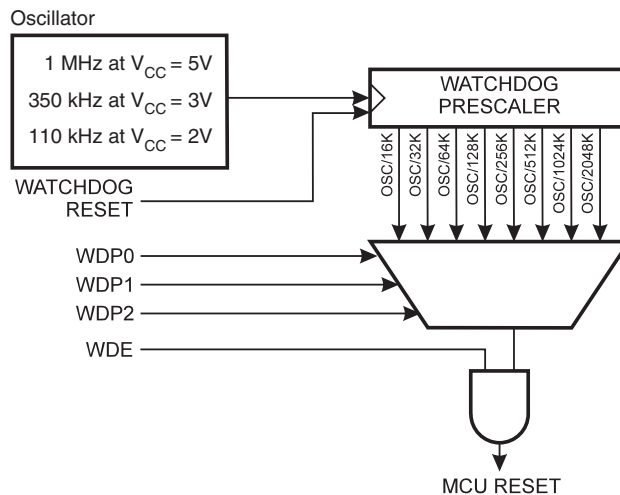


## Watchdog Timer

The Watchdog Timer is clocked from a separate on-chip oscillator. By controlling the Watchdog Timer prescaler, the Watchdog reset interval can be adjusted as shown in Table 11. See characterization data for typical values at other  $V_{CC}$  levels. The WDR – Watchdog Reset – instruction resets the Watchdog Timer. Eight different clock cycle periods can be selected to determine the reset period. If the reset period expires without another Watchdog reset, the ATtiny28 resets and executes from the reset vector. For timing details on the Watchdog reset, refer to page 17.

To prevent unintentional disabling of the watchdog, a special turn-off sequence must be followed when the watchdog is disabled. Refer to the description of the Watchdog Timer Control Register for details.

**Figure 22.** Watchdog Timer



### Watchdog Timer Control Register – WDTCR

| Bit           | 7 | 6 | 5 | 4     | 3   | 2    | 1    | 0    |       |
|---------------|---|---|---|-------|-----|------|------|------|-------|
| \$01          | - | - | - | WDTOE | WDE | WDP2 | WDP1 | WDP0 | WDTCR |
| Read/Write    | R | R | R | R/W   | R/W | R/W  | R/W  | R/W  |       |
| Initial value | 0 | 0 | 0 | 0     | 0   | 0    | 0    | 0    |       |

- **Bits 7..5 - Res: Reserved bits**

These bits are reserved bits in the ATtiny28 and will always read as zero.

- **Bit 4 - WDTOE: Watch Dog Turn-off Enable**

This bit must be set (one) when the WDE bit is cleared. Otherwise, the watchdog will not be disabled. Once set, hardware will clear this bit to zero after four clock cycles. Refer to the description of the WDE bit for a watchdog disable procedure.

- **Bit 3 - WDE: Watch Dog Enable**

When the WDE is set (one) the Watchdog Timer is enabled, and if the WDE is cleared (zero) the Watchdog Timer function is disabled. WDE can only be cleared if the WDTOE bit is set(one). To disable an enabled watchdog timer, the following procedure must be followed:

1. In the same operation, write a logical one to WDTOE and WDE. A logical one must be written to WDE even though it is set to one before the disable operation starts.
2. Within the next four clock cycles, write a logical 0 to WDE. This disables the watchdog.

- **Bits 2..0 - WDP2, WDP1, WDP0: Watchdog Timer Prescaler 2, 1 and 0**

The WDP2, WDP1 and WDP0 bits determine the Watchdog Timer prescaling when the Watchdog Timer is enabled. The different prescaling values and their corresponding time-out periods are shown in Table 11.

**Table 11.** Watch Dog Timer Prescale Select

| WDP2 | WDP1 | WDP0 | Number of WDT Oscillator cycles | Typical time-out at V <sub>CC</sub> = 2.0V | Typical time-out at V <sub>CC</sub> = 3.0V | Typical time-out at V <sub>CC</sub> = 5.0V |
|------|------|------|---------------------------------|--|--|--|
| 0    | 0    | 0    | 16K cycles                      | 0.15s                                      | 47 ms                                      | 15 ms                                      |
| 0    | 0    | 1    | 32K cycles                      | 0.30s                                      | 94 ms                                      | 30 ms                                      |
| 0    | 1    | 0    | 64K cycles                      | 0.60s                                      | 0.19 s                                     | 60 ms                                      |
| 0    | 1    | 1    | 128K cycles                     | 1.2s                                       | 0.38 s                                     | 0.12 s                                     |
| 1    | 0    | 0    | 256K cycles                     | 2.4s                                       | 0.75 s                                     | 0.24 s                                     |
| 1    | 0    | 1    | 512K cycles                     | 4.8s                                       | 1.5 s                                      | 0.49 s                                     |
| 1    | 1    | 0    | 1,024K cycles                   | 9.6s                                       | 3.0 s                                      | 0.97 s                                     |
| 1    | 1    | 1    | 2,048K cycles                   | 19s  | 6.0 s                                      | 1.9 s                                      |

Note: The frequency of the watchdog oscillator is voltage dependent as shown in the section “Typical Characteristics – Preliminary Data” on page 50.  
 The WDR – Watchdog Reset – instruction should always be executed before the Watchdog Timer is enabled. This ensures that the reset period will be in accordance with the Watchdog Timer prescale settings. If the Watchdog Timer is enabled without reset, the watchdog timer may not start counting from zero.

## Calibrated Internal RC Oscillator

The calibrated internal oscillator provides a fixed 1MHz (nominal) clock at 3V and 25°C. This clock may be used as the system clock. See the section “Clock Options” on page 4 for information on how to select this clock as the system clock. This oscillator can be calibrated by writing the calibration byte to the OSCCAL register. When this oscillator is used as the chip clock, the Watchdog Oscillator will still be used for the Watchdog Timer and for the reset time-out. For details on how to use the pre-programmed calibration value, see the section “Calibration Byte” on page 40.

### Oscillator Calibration Register – OSCCAL

|               |      |      |      |      |      |      |      |      |        |
|---------------|------|------|------|------|------|------|------|------|--------|
| Bit           | 7    | 6    | 5    | 4    | 3    | 2    | 1    | 0    |        |
| \$00          | CAL7 | CAL6 | CAL5 | CAL4 | CAL3 | CAL2 | CAL1 | CAL0 | OSCCAL |
| 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..0 - CAL7..CAL0: Oscillator Calibration Value**

Writing the calibration byte to this address will trim the internal oscillator to remove process variation from the oscillator frequency. When OSCCAL is zero, the lowest available frequency is chosen. Writing non-zero values to the register will increase the frequency to the internal oscillator. Writing \$FF to the register gives the highest available frequency. Table 12 shows the range for OSCCAL. Note that the oscillator is intended for calibration to 1.0 MHz, thus tuning to other values is not guaranteed.

**Table 12.** Internal RC Oscillator Range

| OSCCAL Value | Min Frequency | Max Frequency |
|--------------|---------------|---------------|
| 0x00         | 0.5 MHz       | 1.0 MHz       |
| 0x7F         | 0.7 MHz       | 1.5 MHz       |
| 0xFF         | 1.0 MHz       | 2.0 MHz       |

## Hardware Modulator

ATtiny28 features a built-in hardware modulator connected to a high-current output pad, PA2. The hardware modulator generates a configurable pulse train. The on-time of a pulse can be set to a number of chip clock cycles. This is done by configuring the Modulation Control Register – MODCR.

### Modulation Control Register – MODCR

|               |  |     |     |     |     |     |     |     |       |
|---------------|--|-----|-----|-----|-----|-----|-----|-----|-------|
| Bit           | 7  | 6   | 5   | 4   | 3   | 2   | 1   | 0   |       |
| \$02          | <b>ONTIM4   ONTIM3   ONTIM2   ONTIM1   ONTIM0   MCONF2   MCONF1   MCONF0</b> |     |     |     |     |     |     |     | MODCR |
| 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..3 - ONTIM4..0: Modulation On-time**

This 5 bit value +1 determines the number of clock cycles the output pin PA2 is active (low).

- **Bit 2..0 - MCONF2..0: Modulation Configuration bit 2, 1 and 0**

These 3 bits determine the relationship between the on- and off-times of the modulator, and hereby the duty-cycle. The various setting is shown in Table 13. The minimum and maximum modulation period is also shown in the table. The minimum modulation period is obtained by setting ONTIM to zero, while the maximum period is obtained by setting ONTIM to 31. The configuration values for some common oscillator and carrier frequencies are listed in Table 15. The relationship between oscillator frequency and carrier frequency is:

$$f_{carrier} = \frac{f_{osc}}{(On - time + Off - time)}$$

If the MCONF register is set to 111, the carrier frequency will be equal to the oscillator frequency.

**Table 13.** MCONF2..0 Effect on Duty-cycle and Modulation Period

| MCONF2..0 | On-time       | Off-time      | Duty-cycle | Min Period | Max Period | Comment               |
|-----------|---------------|---------------|------------|------------|------------|-----------------------|
| 000       | X             | X             | 100%       | X          | X          | Unmodulated output    |
| 001       | ONTIM+1       | ONTIM+1       | 50%        | 2 CK       | 64 CK      |                       |
| 010       | ONTIM+1       | 2 x (ONTIM+1) | 33%        | 3 CK       | 96 CK      |                       |
| 011       | ONTIM+1       | 3 x (ONTIM+1) | 25%        | 4 CK       | 128 CK     |                       |
| 100       | 2 x (ONTIM+1) | ONTIM+1       | 67%        | 3 CK       | 96 CK      |                       |
| 101       | 3 x (ONTIM+1) | ONTIM+1       | 75%        | 4 CK       | 128 CK     |                       |
| 110       | Reserved      |               |            |            |            |                       |
| 111       | X             | X             | Note 1     | 1 CK       | 1 CK       | High frequency output |

Note: 1. In the high frequency mode, the output is gated with the clock signal. Thus, the on- and off-times will be dependent on the clock input to the MCU. Also note that when changing from this mode directly to another modulation mode, the output will have a small glitch. Thus, PA2 should be set to stop the modulated output before changing from this mode.

PA2 is the built-in high current LED driver, and it is always an output pin. The output buffer can sink 25 mA at 1.8V. When MCONF is zero, modulation is switched off, and the pin acts as a normal high current output pin. The following truth table shows the effect of various PORTA2 and MCONF settings.

**Table 14. PA2 Output**

| PORTA2 | MCONF | PA2 Output |
|--------|-------|------------|
| 0      | 0     | 0          |
| 0      | 1-7   | Modulated  |
| 1      | X     | 1          |

The modulation period is available as a prescale to Timer/Counter0, and thus this timer should be used to time the length of each burst. If the number of pulses to be sent is N, the number 255 – N should be loaded to the timer. When an overflow occurs, the transmission is complete. The OOM01 and OOM00 bits in TCCR0 can be configured to automatically change the value on PA2 when a Timer/Counter0 overflow occurs. See the section “Timer/Counter0” on page 21 for details on how to configure the OOM01 and OOM00 bits.

The modulation period is available as a prescale even when PORTA2 is high and modulation is stopped. Thus, this prescale can also be used to time the intervals between bursts.

To get a glitch free output, the user should first configure the MODCR register to enable modulation. Then there are two ways to start the modulation:

1. Clear the PORTA2 bit in Port A Data Register (PORTA).
2. Configure OOM00 and OOM01 bits in the Timer/Counter0 Control Register (TCCR0) to clear PA2 on the next overflow. Either an overflow or a forced overflow can then be used to start modulation.

The PA2 output will then be set low at the start of the next cycle. To stop the modulated output, the user should set the PORTA2 bit, or configure OOM00 and OOM01 to set PA2 on the next overflow. If the MODCR register is changed during modulation, the changed value will take effect at the start of the next cycle, producing a glitch free output. See Figure 23 and Figure 30 on page 34.

**Figure 23. The Hardware Modulator**

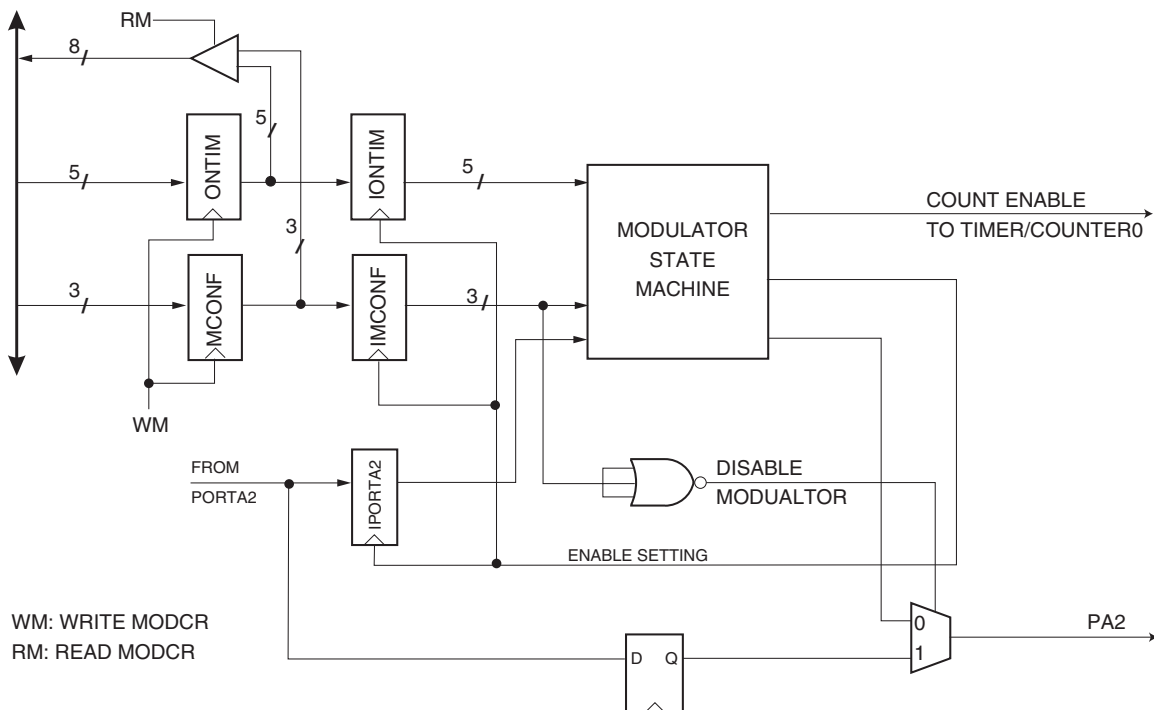
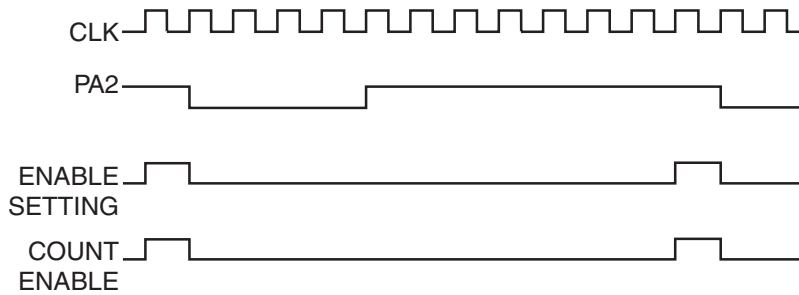


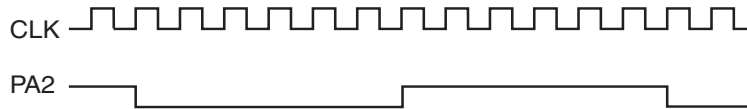
Figure 24 to Figure 27 show examples on output from the Modulator. Figure 24 also shows the timing for the enable setting signal and for the count enable signal to Timer/Counter0.

**Figure 24.** Modulation with  $ONTIM = 3$ ,  $MCONF = 010$ .



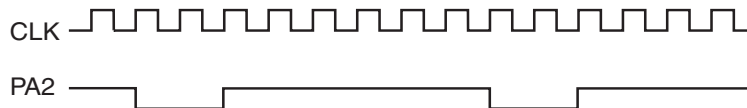
Clock frequency: 455 kHz, Modulation frequency: 38 kHz, Duty-cycle: 33%

**Figure 25.** Modulation with  $ONTIM = 5$ ,  $MCONF = 001$



Clock frequency: 455 kHz, Modulation frequency: 38 kHz, Duty-cycle: 50%

**Figure 26.** Modulation with  $ONTIM = 1$ ,  $MCONF = 011$



Clock frequency: 3.64 MHz, Modulation frequency: 455 kHz, Duty-cycle: 25%

**Figure 27.** Modulation with  $ONTIM = 3$ ,  $MCONF = 001$



Clock frequency: 3.64 MHz, Modulation frequency: 455 kHz, Duty-cycle: 50%

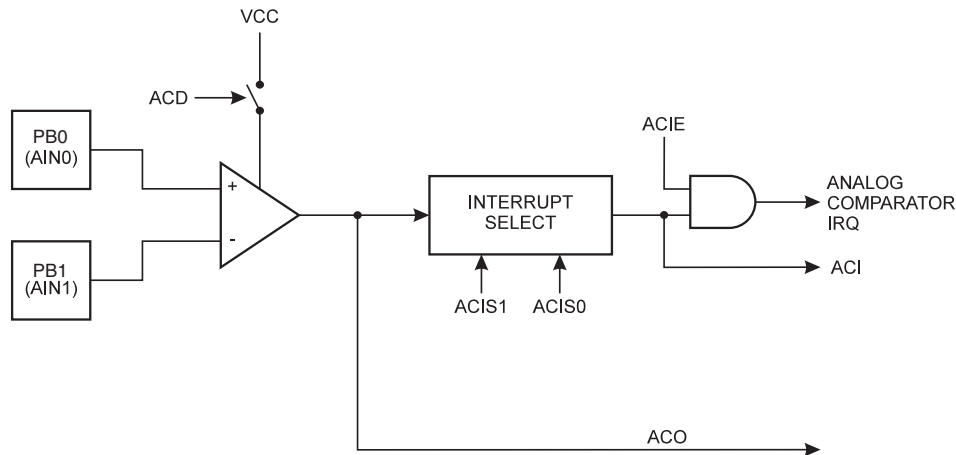
**Table 15.** Some Common Modulator Configurations

| Crystal/Resonator Frequency | Carrier Frequency | % Error in Frequency | Duty Cycle  | ONTIM Value | MCONF Value |
|-----------------------------|-------------------|----------------------|-------------|-------------|-------------|
| 455 kHz                     | 38 kHz            | 0.2                  | 25%         | 2           | 011         |
| 455 kHz                     | 38 kHz            | 0.2                  | 33%         | 3           | 010         |
| 455 kHz                     | 38 kHz            | 0.2                  | 50%         | 5           | 001         |
| 455 kHz                     | 38 kHz            | 0.2                  | 67%         | 3           | 100         |
| 455 kHz                     | 38 kHz            | 0.2                  | 75%         | 2           | 101         |
| 1 MHz                       | 38 kHz            | 1.2                  | 50%         | 12          | 001         |
| 1.8432 MHz                  | 38 kHz            | 1.1                  | 25%         | 11          | 011         |
| 1.8432 MHz                  | 38 kHz            | 1.1                  | 33%         | 15          | 010         |
| 1.8432 MHz                  | 38 kHz            | 1.1                  | 50%         | 23          | 001         |
| 2 MHz                       | 38 kHz            | 1.2                  | 25%         | 12          | 011         |
| 2 MHz                       | 38 kHz            | 1.2                  | 50%         | 25          | 001         |
| 2.4576 MHz                  | 38 kHz            | 1.1                  | 50%         | 31          | 001         |
| 3.2768 MHz                  | 38 kHz            | 2.0                  | 25%         | 21          | 011         |
| 4 MHz                       | 38 kHz            | 1.2                  | 25%         | 25          | 011         |
| 455 kHz                     | 455 kHz           | 0.0                  | approx. 50% | X           | 111         |
| 1 MHz                       | 455 kHz           | 9.9                  | 50%         | 0           | 001         |
| 1.82 MHz                    | 455 kHz           | 0.0                  | 25%         | 0           | 011         |
| 1.82 MHz                    | 455 kHz           | 0.0                  | 50%         | 1           | 001         |
| 1.8432 MHz                  | 455 kHz           | 1.3                  | 25%         | 0           | 011         |
| 1.8432 MHz                  | 455 kHz           | 1.3                  | 50%         | 1           | 001         |
| 2 MHz                       | 455 kHz           | 9.9                  | 25%         | 0           | 011         |
| 2 MHz                       | 455 kHz           | 9.9                  | 50%         | 1           | 001         |
| 2.4576 MHz                  | 455 kHz           | 10.0                 | 33%         | 1           | 010         |
| 2.4576 MHz                  | 455 kHz           | 10.0                 | 50%         | 2           | 001         |
| 3.2768 MHz                  | 455 kHz           | 10.0                 | 25%         | 1           | 011         |
| 3.2768 MHz                  | 455 kHz           | 10.0                 | 50%         | 3           | 001         |
| 3.64 MHz                    | 455 kHz           | 0.0                  | 25%         | 1           | 011         |
| 3.64 MHz                    | 455 kHz           | 0.0                  | 50%         | 3           | 001         |
| 4 MHz                       | 455 kHz           | 9.9                  | 25%         | 1           | 011         |
| 4 MHz                       | 455 kHz           | 9.9                  | 50%         | 3           | 001         |

## Analog Comparator

The analog comparator compares the input values on the positive input PB0 (AIN0) and negative input PB1 (AIN1). When the voltage on the positive input PB0 (AIN0) is higher than the voltage on the negative input PB1 (AIN1), the Analog Comparator Output, ACO is set (one). 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 28.

**Figure 28.** Analog Comparator Block Diagram



### Analog Comparator Control And Status Register – ACSR

| Bit           | 7   | 6 | 5   | 4   | 3    | 2 | 1     | 0     |      |
|---------------|-----|---|-----|-----|------|---|-------|-------|------|
| \$08          | ACD | - | ACO | ACI | ACIE | - | ACIS1 | ACIS0 | ACSR |
| Read/Write    | R/W | R | R   | R/W | R/W  | R | R/W   | R/W   |      |
| Initial value | 1   | 0 | X   | 0   | 0    | 0 | 0     | 0     |      |

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

When this bit is set(one), the power to the analog comparator is switched off. This bit can be set at any time to turn off the analog comparator. When changing the ACD bit, the Analog Comparator Interrupt must be disabled by clearing the ACIE bit in ACSR. Otherwise an interrupt can occur when the bit is changed. To use the analog comparator, the user must clear this bit.

- **Bit 6 - Res: Reserved bit**

This bit is a reserved bit in the ATtiny28 and will always read as zero.

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

ACO is directly connected to the comparator output.

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

This bit is set (one) when a comparator output event triggers the interrupt mode defined by ACI1 and ACI0. The Analog Comparator Interrupt routine is executed if the ACIE bit is set (one) and the I-bit in SREG is set (one). 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 set (one) and the I-bit in the Status Register is set (one), the analog comparator interrupt is activated. When cleared (zero), the interrupt is disabled.

- **Bit 2 - RES: Reserved bit**

This bit is a reserved bit in the ATtiny28 and will always read as zero.

• **Bits 1,0 - ACIS1, ACIS0: Analog Comparator Interrupt Mode Select**

These bits determine which comparator events that trigger the Analog Comparator interrupt. The different settings are shown in Table 16.

**Table 16.** ACIS1/ACIS0 Settings

| ACIS1 | ACIS0 | Interrupt Mode                              |
|-------|-------|---|
| 0     | 0     | Comparator Interrupt on Output Toggle       |
| 0     | 1     | Reserved                                    |
| 1     | 0     | Comparator Interrupt on Falling Output Edge |
| 1     | 1     | Comparator Interrupt on Rising Output Edge  |

Note: When changing the ACIS1/ACIS0 bits, The Analog Comparator Interrupt must be disabled by clearing its Interrupt Enable bit in the ACSR register. Otherwise an interrupt can occur when the bits are changed.

Caution: Using the SBI or CBI instruction on other bits than ACI in this register, will write a one back into ACI if it is read as set, thus clearing the flag.

## 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 for changing drive value (if configured as output) or enabling/disabling of pull-up resistors (if configured as input).

### Port A

Port A is a 4-bit I/O port. PA2 is output only, while PA3, PA1 and PA0 are bi-directional.

Three I/O memory address locations are allocated for Port A, one each for the Data Register – PORTA, \$1B, Port A Control Register – PACR, \$1A and the Port A Input Pins – PINA, \$19. The Port A Input Pins address is read only, while the Data Register and the Control Register are read/write. Compared to other output ports, the Port A output is delayed one extra clock cycle.

Port pins PA0, PA1 and PA3 have individually selectable pull-up resistors. When pins PA0, PA1 or PA3 are used as inputs and are externally pulled low, they will source current if the internal pull-up resistors are activated. PA2 is output only. The PA2 output buffer can sink 25 mA and thus drive a high current LED directly. This output can also be modulated, see the section “Hardware Modulator” on page 27 for details.

#### Port A Data Register – PORTA

| Bit           | 7 | 6 | 5 | 4 | 3      | 2      | 1      | 0      |       |
|---------------|---|---|---|---|--------|--------|--------|--------|-------|
| \$1B          | - | - | - | - | PORTA3 | PORTA2 | PORTA1 | PORTA0 | PORTA |
| Read/Write    | R | R | R | R | R/W    | R/W    | R/W    | R/W    |       |
| Initial value | 0 | 0 | 0 | 0 | 0      | 1      | 0      | 0      |       |

#### Port A Control Register – PACR

| Bit           | 7 | 6 | 5 | 4 | 3    | 2     | 1    | 0    |      |
|---------------|---|---|---|---|------|-------|------|------|------|
| \$1A          | - | - | - | - | DDA3 | PA2HC | DDA1 | DDA0 | PACR |
| Read/Write    | R | R | R | R | R/W  | R/W   | R/W  | R/W  |      |
| Initial value | 0 | 0 | 0 | 0 | 0    | 0     | 0    | 0    |      |

• **Bit 7..4 - Res: Reserved bits**

These bits are reserved bits in the ATtiny28 and always read as zero.

• **Bit 3 - DDA3: Data Direction PA3**

When DDA3 is set (one), the corresponding pin is an output pin. Otherwise it is an input pin.



- **Bit 2 - PA2HC: PORTA2 High Current Enable**

When the PA2HC bit is set (one), an additional driver at the output pin PA2 is enabled. This makes it possible to sink 25 mA at  $V_{CC} = 1.8V$  ( $V_{OL} = 0.8V$ ). When the PA2HC bit is cleared (zero), PA2 can sink 15 mA at  $V_{CC} = 1.8V$  ( $V_{OL} = 0.8V$ ).

- **Bit 1,0 - DDA1, DDA0: Data Direction PA1 and PA0**

When DDAn is set (one), the corresponding pin is an output pin. Otherwise it is an input pin.

### Port A as General Digital I/O

PA3, PA1 and PA0 are general I/O pins. The DDAn (n: 3,1,0) bits in PACR select the direction of these pins. If DDAn is set (one), PAn is configured as an output pin. If DDAn is cleared (zero), PAn is configured as an input pin. If PORTAn is set (one) when the pin is configured as an input pin, the MOS pull up resistor is activated. To switch the pull up resistor off, the PORTAn bit has to be cleared switch the pull up resistor off, the PORTAn bit has to be cleared (zero) or the pin has to be configured as an output pin. The effects of the DDAn and PORTAn bits on PA3, PA1 and PA0 are shown in Table 17. The port pins are tri-stated when a reset condition becomes active, even if the clock is not running.

**Table 17.** DDAn Effects on Port A Pins

| DDAn | PORTAn | I/O    | Pull Up | Comment                                     |
|------|--------|--------|---------|---|
| 0    | 0      | Input  | No      | Tri-state (Hi-Z)                            |
| 0    | 1      | Input  | Yes     | PAn will source current if ext. pulled low. |
| 1    | 0      | Output | No      | Push-pull Zero Output                       |
| 1    | 1      | Output | No      | Push-pull One Output                        |

n: 3,1,0, pin number

### Port A Input Pins Address – PINA

| Bit           | 7 | 6 | 5 | 4 | 3     | 2 | 1     | 0     |      |
|---------------|---|---|---|---|-------|---|-------|-------|------|
| \$19          | - | - | - | - | PINA3 | - | PINA1 | PINA0 | PINA |
| Read/Write    | R | R | R | R | R     | R | R     | R     |      |
| Initial value | 0 | 0 | 0 | 0 | N/A   | 0 | N/A   | N/A   |      |

The Port A Input Pins address – PINA – is not a register, and this address enables access to the physical value on each Port A pin. When reading PORTA the Port A Data Latch is read, and when reading PINA, the logical values present on the pins are read.

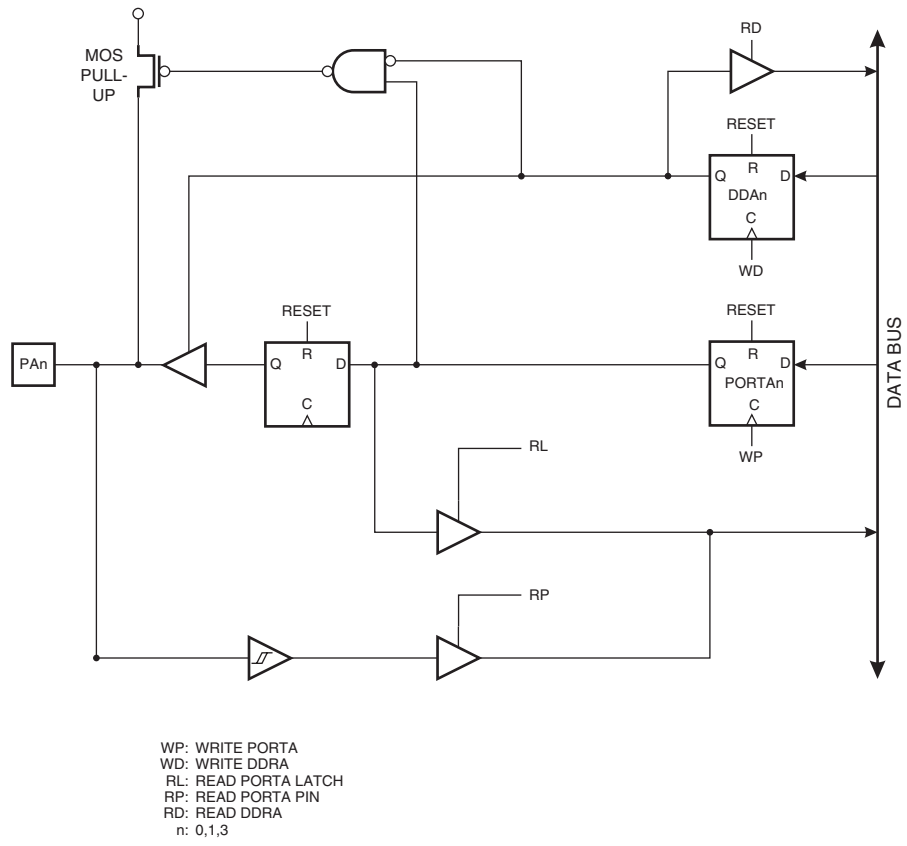
### Alternate Function of PA2

PA2 is the built-in high current LED driver, and it is always an output pin. The output signal can be modulated with a software programmable frequency. See the section “Hardware Modulator” on page 27 for further details

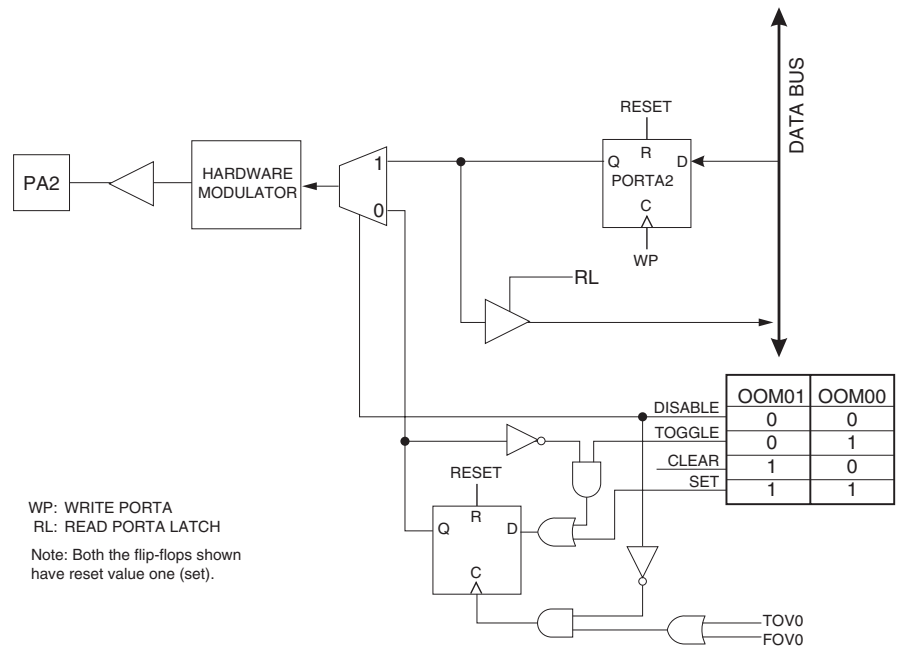
### Port A Schematics

Note that all port pins are synchronized. The synchronization latches are however, not shown in the figures.

**Figure 29.** Port A Schematic Diagram (Pins PA0, PA1 and PA3)



**Figure 30.** Port A Schematic Diagram (Pin PA2)



## Port B

Port B is an 8-bit input port.

One I/O address location is allocated for the Port B, the Port B Input Pins – PINB, \$16. The Port B Input Pins address is read only.

All port pins have pullups that can be switched on for all Port B pins simultaneously. If any of the Port B special functions is enabled, the corresponding pullup(s) is disabled. When pins PB0 to PB7 are externally pulled low, they will source current ( $I_{IL}$ ) if the internal pullup resistors are activated.

The Port B pins with alternate functions are shown in the following table:

**Table 18.** Port B Pins Alternate Functions

| Port Pin | Alternate Functions                         |
|----------|---|
| PB0      | AIN0 (Analog comparator positive input)     |
| PB1      | AIN1 (Analog comparator negative input)     |
| PB2      | T0 (Timer/Counter 0 external counter input) |
| PB3      | INT0 (External interrupt 0 input)           |
| PB4      | INT1 (External interrupt 1 input)           |

## Port B Input Pins Address – PINB

| Bit           | 7  | 6   | 5   | 4   | 3   | 2   | 1   | 0   |      |
|---------------|--|-----|-----|-----|-----|-----|-----|-----|------|
| \$16          | <b>PINB7 PINB6 PINB5 PINB4 PINB3 PINB2 PINB1 PINB0</b> |     |     |     |     |     |     |     | PINB |
| Read/Write    | R  | R   | R   | R   | R   | R   | R   | R   |      |
| Initial value | N/A  | N/A | N/A | N/A | N/A | N/A | N/A | N/A |      |

The Port B Input Pins address – PINB – is not a register, and this address enables access to the physical value on each Port B pin. When reading PINB, the logical values present on the pins are read.

## Port B As General Digital Input

All 8 pins in Port B have equal functionality when used as digital input pins.

PBn, General input pin: To switch the pullup resistors on, the PLUPB bit in the MCUCS register must be set (one). This bit controls the pullup on all Port B pins. To turn the pullups off, this bit has to be cleared (zero). Note that if any Port B pins are used for alternate functions, the pullup on the corresponding pins are disabled. The port pins are tri-stated when a reset condition becomes active, even if the clock is not running.

## Alternate Functions of Port B

All Port B pins are connected to a low level detector that can trigger the low level input interrupt. See “Low Level Input Interrupt” on page 21 for details. In addition, Port B has the following alternate functions:

- **INT1 - Port B, Bit 4**

INT1, External interrupt source 1. The PB4 pin can serve as an external interrupt source to the MCU. See the interrupt description for details on how to enable and configure this interrupt. If the interrupt is enabled, the pull up resistor on PB4 is disabled and PB4 will not give low level interrupts.

- **INT0 - Port B, Bit 3**

INT0, External Interrupt source 0. The PB3 pin can serve as an external interrupt source to the MCU. See the interrupt description for details on how to enable and configure this interrupt. If the interrupt is enabled, the pull up resistor on PB3 is disabled and PB3 will not give low level interrupts.

- **T0 - Port B, Bit 2**

T0, Timer/Counter0 counter source. See the timer description for further details. If T0 is used as the counter source, the pull up resistor on PB2 is disabled and PB2 will not give low level interrupts.

- **AIN1 – Port B, Bit 1**

AIN1, Analog Comparator Negative input. When the on-chip analog comparator is enabled, this pin also serves as the negative input of the comparator. If the analog comparator is enabled, the pull up resistors on PB1 and PB0 are disabled and these pins will not give low level interrupts.

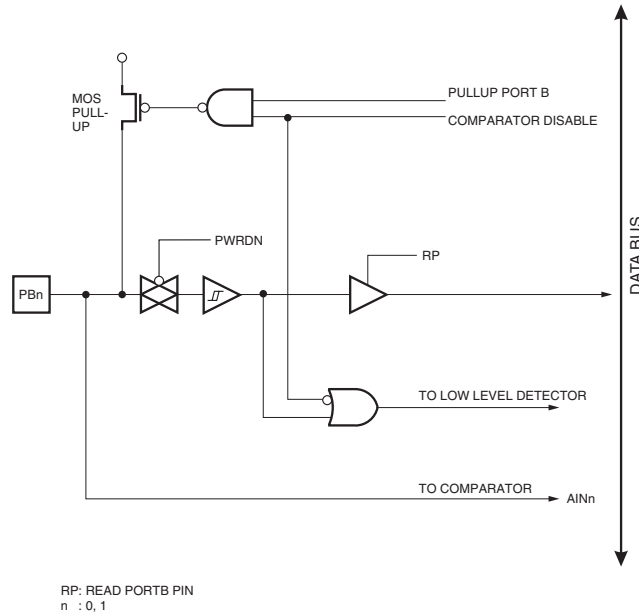
- **AIN0 – Port B, Bit 0**

AIN0, Analog Comparator Positive input. When the on-chip analog comparator is enabled, this pin also serves as the positive input of the comparator. If the analog comparator is enabled, the pull up resistors on PB1 and PB0 are disabled and these pins will not give low level interrupts.

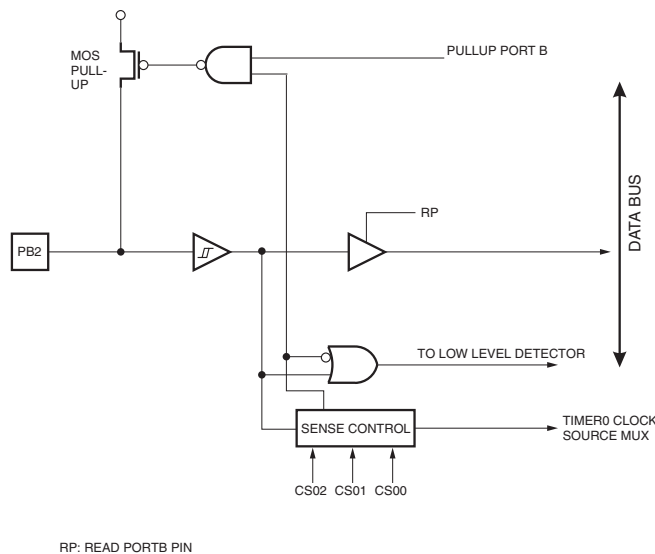
### Port B Schematics

Note that all port pins are synchronized. The synchronization latches are however, not shown in the figures.

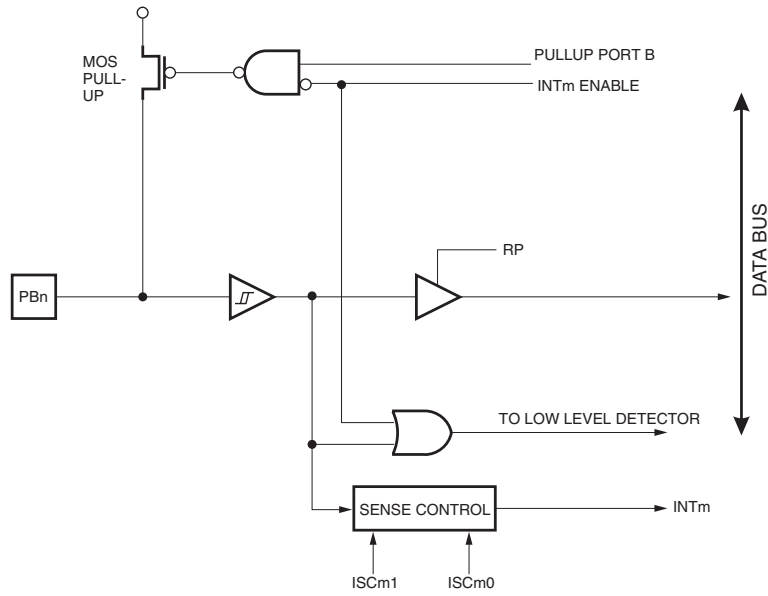
**Figure 31.** Port B Schematic Diagram (pins PB0 and PB1)



**Figure 32.** Port B Schematic Diagram (pin PB2)

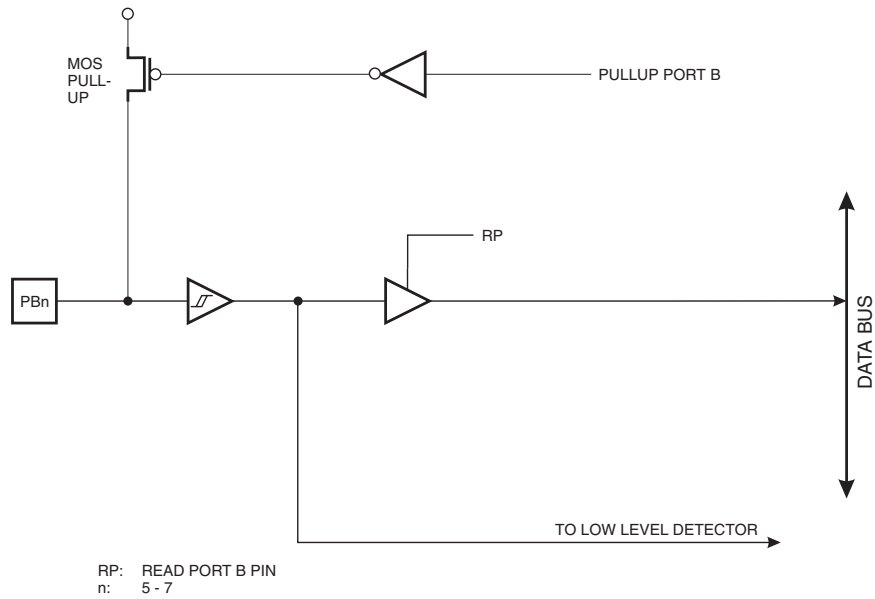


**Figure 33.** PORT B Schematic Diagram (pins PB3 and PB4)



RP: READ PORTB PIN  
 n : 3, 4  
 m : 0, 1

**Figure 34.** PORT B Schematic Diagram (Pins PB7 - PB5)



RP: READ PORT B PIN  
 n: 5 - 7

## Port D

Port D is an 8-bit bi-directional I/O port with internal pull-up resistors.

Three I/O memory address locations are allocated for Port D, one each for the Data Register – PORTD, \$12, Data Direction Register – DDRD, \$11 and the Port D Input Pins – PIND, \$10. The Port D Input Pins address is read only, while the Data Register and the Data Direction Register are Read/Write.

The Port D output buffers can sink 10 mA. As inputs, Port D pins that are externally pulled low will source current if the pull-up resistors are activated.

### Port D Data Register – PORTD

|               |  |     |     |     |     |     |     |     |              |
|---------------|--|-----|-----|-----|-----|-----|-----|-----|--------------|
| Bit           | 7  | 6   | 5   | 4   | 3   | 2   | 1   | 0   |              |
| \$12          | <b>PORTD7   PORTD6   PORTD5   PORTD4   PORTD3   PORTD2   PORTD1   PORTD0</b> |     |     |     |     |     |     |     | <b>PORTD</b> |
| 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   |              |

### Port D Data Direction Register – DDRD

|               |  |     |     |     |     |     |     |     |             |
|---------------|--|-----|-----|-----|-----|-----|-----|-----|-------------|
| Bit           | 7  | 6   | 5   | 4   | 3   | 2   | 1   | 0   |             |
| \$11          | <b>DDD7   DDD6   DDD5   DDD4   DDD3   DDD2   DDD1   DDD0</b> |     |     |     |     |     |     |     | <b>DDRD</b> |
| 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   |             |

### Port D Input Pins Address – PIND

|               |  |     |     |     |     |     |     |     |             |
|---------------|--|-----|-----|-----|-----|-----|-----|-----|-------------|
| Bit           | 7  | 6   | 5   | 4   | 3   | 2   | 1   | 0   |             |
| \$10          | <b>PIND7   PIND6   PIND5   PIND4   PIND3   PIND2   PIND1   PIND0</b> |     |     |     |     |     |     |     | <b>PIND</b> |
| Read/Write    | R  | R   | R   | R   | R   | R   | R   | R   |             |
| Initial value | N/A  | N/A | N/A | N/A | N/A | N/A | N/A | N/A |             |

The port D Input Pins Address – PIND – is not a register, and this address enables access to the physical value on each Port D pin. When reading PORTD, the Port D Data Latch is read, and when reading PIND, the logical values present on the pins are read.

### Port D as General Digital I/O

All 8 pins in Port D have equal functionality when used as digital I/O pins.

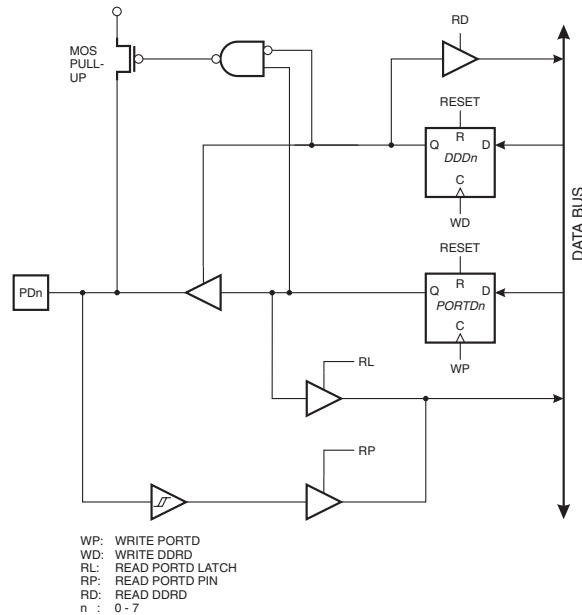
PD<sub>n</sub>, General I/O-pin: The DDD<sub>n</sub> bit in the DDRD register selects the direction of this pin. If DDD<sub>n</sub> is set (one), PD<sub>n</sub> is configured as an output pin. If DDD<sub>n</sub> is cleared (zero), PD<sub>n</sub> is configured as an input pin. If PD<sub>n</sub> is set (one) when configured as an input pin, the MOS pull up resistor is activated. To switch the pull up resistor off, the PD<sub>n</sub> has to be cleared (zero), or the pin has to be configured as an output pin. The port pins are tri-stated when a reset condition becomes active, even if the clock is not running.

**Table 19.** DDD<sub>n</sub> Bits on Port D Pins

| DDD <sub>n</sub> | PORTD <sub>n</sub> | I/O    | Pull Up | Comment  |
|------------------|--------------------|--------|---------|--|
| 0                | 0                  | Input  | No      | Tri-state (Hi-Z)                                       |
| 0                | 1                  | Input  | Yes     | PD <sub>n</sub> will source current if ext. pulled low |
| 1                | 0                  | Output | No      | Push-pull Zero Output                                  |
| 1                | 1                  | Output | NO      | Push-pull One Output                                   |

n: 7,6,...,0, pin number

**Figure 35.** Port D Schematic Diagram (Pins PD7 - PD0)



## Memory Programming

### Program Memory Lock Bits

The ATtiny28 MCU provides two Lock bits which can be left unprogrammed ('1') or can be programmed ('0') to obtain the additional features listed in Table 20. The Lock bits can only be erased with the Chip Erase command.

**Table 20.** Lock Bit Protection Modes

| Memory Lock Bits |     |     | Protection Type  |
|------------------|-----|-----|--|
| Mode             | LB1 | LB2 |  |
| 1                | 1   | 1   | No memory lock features enabled.                             |
| 2                | 0   | 1   | Further programming of the Flash is disabled. <sup>(1)</sup> |
| 3                | 0   | 0   | Same as mode 2, and verify is also disabled.                 |

Note: 1. Further programming of the Fuse bits are also disabled. Program the Fuse bits before programming the Lock bits.

### Fuse Bits

The ATtiny28 has five Fuse bits, INTCAP and CKSEL3..0.

- When the INTCAP Fuse is programmed ('0'), internal load capacitors are connected between XTAL1/XTAL2 and GND, similar to C1 and C2 in Figure 2. See "Crystal Oscillator" on page 4. Default value is unprogrammed ('1').
- CKSEL3..0 Fuses: See Table 1, "Device Clocking Option Select," on page 4 and Table 5, "ATtiny28 Clock Options and Start-up Time," on page 15, for which combination of CKSEL3..0 to use. Default value is '0010', Internal RC oscillator with long start-up time.

The status of the Fuse bits is not affected by Chip Erase.

## Signature Bytes

All Atmel microcontrollers have a three-byte signature code which identifies the device. The three bytes reside in a separate address space.

For the ATtiny28 they are:

1. \$000: \$1E (indicates manufactured by Atmel)
2. \$001: \$91 (indicates 2kB Flash memory)
3. \$002: \$07 (indicates ATtiny28 device when signature byte \$001 is \$91)

## Calibration Byte

The ATtiny28 has one byte calibration value for the internal RC Oscillator. This byte resides in the high byte of address \$000 in the signature address space. To make use of this byte, it should be read from this location and written into the normal flash program memory. At start-up, the user software must read this flash location and write the value to the OSCCAL register.

## Programming the Flash

Atmel's ATtiny28 offers 2K bytes of Flash Program memory.

The ATtiny28 is shipped with the on-chip Flash Program memory array in the erased state (i.e. contents = \$FF) and ready to be programmed. This device supports a High-voltage (12V) Parallel programming mode. Only minor currents (<1mA) are drawn from the +12V pin during programming.

The Program memory array in the ATtiny28 is programmed byte-by-byte. During programming, the supply voltage must be in accordance with Table 21.

**Table 21.** Supply Voltage During Programming

| Part     | Serial Programming | Parallel Programming |
|----------|--------------------|----------------------|
| ATtiny28 | Not applicable     | 4.5 - 5.5V           |

## Parallel Programming

This section describes how to parallel program and verify Flash Program memory, Lock bits and Fuse bits in the ATtiny28.

### Signal Names

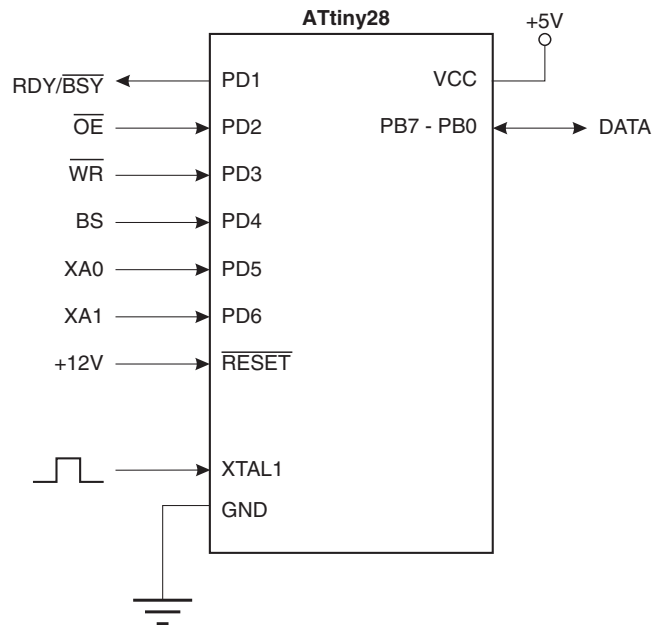
In this section, some pins of the ATtiny28 are referenced by signal names describing their function during parallel programming. See Figure 36 and Table 22. Pins not described in Table 22 are referenced by pin names.

The XA1/XA0 pins determines the action executed when the XTAL1 pin is given a positive pulse. The coding is shown in Table 23.

When pulsing  $\overline{WR}$  or  $\overline{OE}$ , the Command loaded determines the action executed. The Command is a byte where the different bits are assigned functions as shown in Table 24.



**Figure 36.** Parallel Programming



**Table 22.** Pin Name Mapping

| Signal Name in Programming Mode | Pin Name  | I/O | Function  |
|---------------------------------|-----------|-----|---|
| RDY/BSY                         | PD1       | O   | '0': Device is busy programming, '1': Device is ready for new command |
| OE                              | PD2       | I   | Output Enable (Active low)  |
| WR                              | PD3       | I   | Write Pulse (Active low)  |
| BS                              | PD4       | I   | Byte Select ('0' selects low byte, '1' selects high byte)             |
| XA0                             | PD5       | I   | XTAL1 Action Bit 0  |
| XA1                             | PD6       | I   | XTAL1 Action Bit 1  |
| DATA                            | PB7 - PB0 | I/O | Bidirectional Data bus (Output when OE is low)                        |

**Table 23.** XA1 and XA0 Coding

| XA1 | XA0 | Action when XTAL1 is Pulsed   |
|-----|-----|---|
| 0   | 0   | Load Flash/Signature byte Address (High or low address byte for Flash determined by BS) |
| 0   | 1   | Load Data (High or low data byte for Flash determined by BS)                            |
| 1   | 0   | Load Command  |
| 1   | 1   | No Action, Idle   |

**Table 24.** Command Byte Coding

| Command Byte | Command Executed                          |
|--------------|---|
| 1000 0000    | Chip Erase                                |
| 0100 0000    | Write Fuse Bits                           |
| 0010 0000    | Write Lock Bits                           |
| 0001 0000    | Write Flash                               |
| 0000 1000    | Read Signature Bytes and Calibration Byte |
| 0000 0100    | Read Fuse and Lock Bits                   |
| 0000 0010    | Read Flash                                |

### Enter Programming Mode

The following algorithm puts the device in parallel programming mode:

1. Apply 4.5 - 5.5V between VCC and GND.
2. Set  $\overline{\text{RESET}}$  and BS pins to '0' and wait at least 100 ns.
3. Apply 11.5 - 12.5V to  $\overline{\text{RESET}}$ . Any activity on BS within 100 ns after +12V has been applied to  $\overline{\text{RESET}}$ , will cause the device to fail entering programming mode.

### Chip Erase

The Chip Erase command will erase the Flash memory and the Lock bits. The Lock bits are not reset until the Flash has been completely erased. The Fuse bits are not changed. Chip Erase must be performed before the Flash is reprogrammed.

Load Command "Chip Erase"

1. Set XA1, XA0 to '10'. This enables command loading.
2. Set BS to '0'.
3. Set DATA to '1000 0000'. This is the command for Chip Erase.
4. Give XTAL1 a positive pulse. This loads the command.
5. Give  $\overline{\text{WR}}$  a negative pulse. This starts the Chip Erase. RDY/ $\overline{\text{BSY}}$  goes low.
6. Wait until RDY/ $\overline{\text{BSY}}$  goes high before loading a new command.

### Programming the Flash

A: Load Command "Write Flash"

1. Set XA1, XA0 to '10'. This enables command loading.
2. Set BS to '0'
3. Set DATA to '0001 0000'. This is the command for Write Flash.
4. Give XTAL1 a positive pulse. This loads the command.

B: Load Address High Byte

1. Set XA1, XA0 to '00'. This enables address loading.
2. Set BS to '1'. This selects high byte.
3. Set DATA = Address high byte (\$00 - \$03)
4. Give XTAL1 a positive pulse. This loads the address high byte.

C: Load Address Low Byte

1. Set XA1, XA0 to '00'. This enables address loading.
2. Set BS to '0'. This selects low byte.
3. Set DATA = Address low byte (\$00 - \$FF)

4. Give XTAL1 a positive pulse. This loads the address low byte.

D: Load Data Low Byte

1. Set XA1, XA0 to '01'. This enables data loading.
2. Set DATA = Data low byte (\$00 - \$FF)
3. Give XTAL1 a positive pulse. This loads the data low byte.

E: Write Data Low Byte

1. Set BS to '0'. This selects low data.
2. Give  $\overline{WR}$  a negative pulse. This starts programming of the data byte. RDY/ $\overline{BSY}$  goes low.
3. Wait until RDY/ $\overline{BSY}$  goes high to program the next byte.

(See Figure 37 for signal waveforms.)

F: Load Data High Byte

1. Set XA1, XA0 to '01'. This enables data loading.
2. Set DATA = Data high byte (\$00 - \$FF)
3. Give XTAL1 a positive pulse. This loads the data high byte.

G: Write Data High Byte

1. Set BS to '1'. This selects high data.
2. Give  $\overline{WR}$  a negative pulse. This starts programming of the data byte. RDY/ $\overline{BSY}$  goes low.
3. Wait until RDY/ $\overline{BSY}$  goes high to program the next byte.

(See Figure 38 for signal waveforms.)

The loaded command and address are retained in the device during programming. For efficient programming, the following should be considered.

- The command needs only be loaded once when writing or reading multiple memory locations.
- Address high byte needs only be loaded before programming a new 256 word page in the Flash.
- Skip writing the data value \$FF, that is the contents of the entire Flash after a Chip Erase.

These considerations also applies to Flash and Signature bytes reading.

## Reading the Flash

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

1. A: Load Command '0000 0010'.
2. B: Load Address High Byte (\$00 - \$03).
3. C: Load Address Low Byte (\$00 - \$FF).
4. Set  $\overline{OE}$  to '0', and BS 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 from DATA.
6. Set  $\overline{OE}$  to '1'.

### Programming the Fuse Bits

The algorithm for programming the Fuse bits is as follows (refer to Flash Programming for details on Command and Data loading):

1. A: Load Command '0100 0000'.
2. D: Load Data Low Byte. Bit n = '0' programs and bit n = '1' erases the Fuse bit.  
 Bit 4 = INTCAP Fuse  
 Bit 3 = CKSEL3 Fuse  
 Bit 2 = CKSEL2 Fuse  
 Bit 1 = CKSEL1 Fuse  
 Bit 0 = CKSEL0 Fuse  
 Bits 7 - 5 = '1'. These bits are reserved and should be left unprogrammed ('1').
3. E: Write Data Low Byte.

### Programming the Lock Bits

The algorithm for programming the Lock bits is as follows (refer to Programming the Flash for details on Command and Data loading):

1. A: Load Command '0010 0000'.
2. D: Load Data Low Byte. Bit n = '0' programs the Lock bit.  
 Bit 2 = Lock Bit2  
 Bit 1 = Lock Bit1  
 Bits 7 - 3,0 = '1'. These bits are reserved and should be left unprogrammed ('1').
3. E: Write Data Low Byte.

The Lock bits can only be cleared by executing Chip Erase.

### Reading the Fuse and Lock Bits

The algorithm for reading the Fuse and Lock bits is as follows (refer to Programming the Flash for details on Command loading):

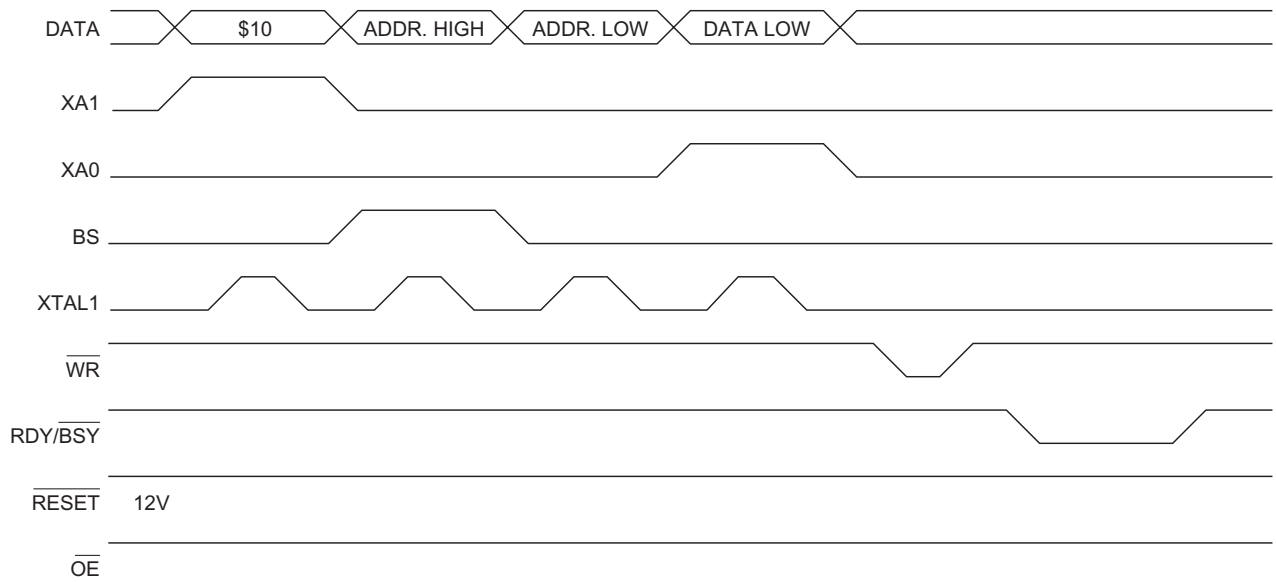
1. A: Load Command '0000 0100'.
2. Set  $\overline{OE}$  to '0', and BS to '0'. The status of the Fuse bits can now be read at DATA ('0' means programmed).  
 Bit 4 = INTCAP Fuse  
 Bit 3 = CKSEL3 Fuse  
 Bit 2 = CKSEL2 Fuse  
 Bit 1 = CKSEL1 Fuse  
 Bit 0 = CKSEL0 Fuse
3. Set BS to '1'. The status of the Lock bits can now be read at DATA ('0' means programmed).  
 Bit 2: Lock Bit2  
 Bit 1: Lock Bit1
4. Set  $\overline{OE}$  to '1'.

### Reading the Signature Bytes and Calibration Byte

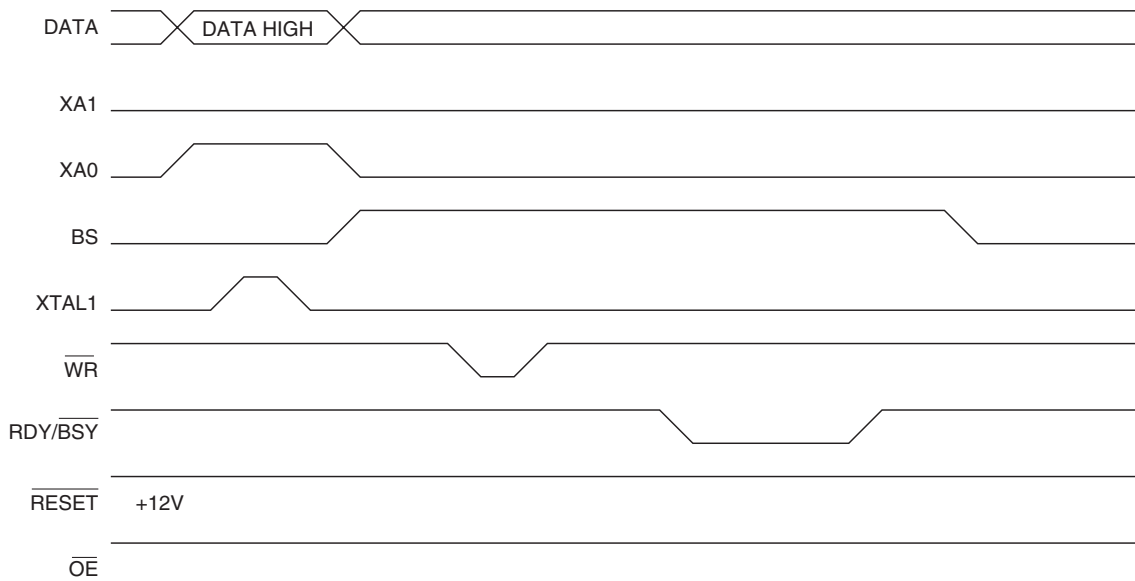
The algorithm for reading the Signature bytes and the Calibration byte is as follows (refer to Programming the Flash for details on Command and Address loading):

1. A: Load Command '0000 1000'.
2. C: Load Address Low Byte (\$00 - \$02).
3. Set  $\overline{OE}$  to '0', and BS to '0'. The selected Signature byte can now be read at DATA.
4. C: Load Address Low Byte (\$00).
5. Set  $\overline{OE}$  to '0', and BS to '1'. The Calibration byte can now be read at DATA.
6. Set  $\overline{OE}$  to '1'.

**Figure 37. Programming the Flash Waveforms**



**Figure 38. Programming the Flash Waveforms (continued)**



## Parallel Programming Characteristics – Preliminary Data

Figure 39. Parallel Programming Timing

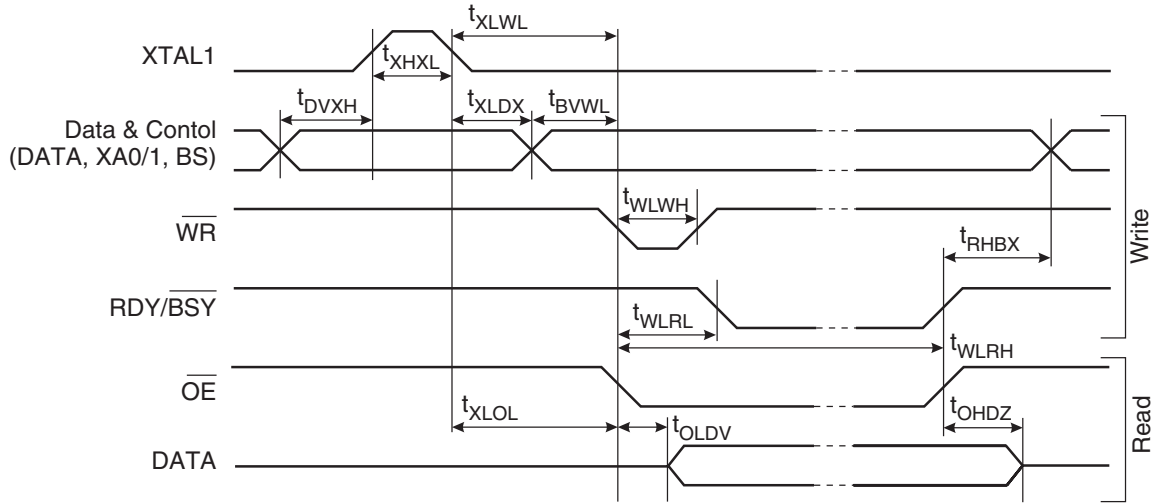


Table 25. Parallel Programming Characteristics

$T_A = 25^\circ\text{C} \pm 10\%$ ,  $V_{CC} = 5\text{V} \pm 10\%$

| Symbol     | Parameter   | Min  | Typ | Max  | Units         |
|------------|---|------|-----|------|---------------|
| $V_{PP}$   | Programming Enable Voltage                        | 11.5 |     | 12.5 | V             |
| $I_{PP}$   | Programming Enable Current                        |      |     | 250  | $\mu\text{A}$ |
| $t_{DVXH}$ | Data & Control Valid before XTAL1 High            | 67   |     |      | ns            |
| $t_{XHXL}$ | XTAL1 Pulse Width High                            | 67   |     |      | ns            |
| $t_{XLDX}$ | Data & Control Hold after XTAL1 Low               | 67   |     |      | ns            |
| $t_{XLWL}$ | XTAL1 Low to $\overline{WR}$ Low                  | 67   |     |      | ns            |
| $t_{BVWL}$ | BS Valid to $\overline{WR}$ Low                   | 67   |     |      | ns            |
| $t_{RHBX}$ | BS Hold after RDY/ $\overline{BSY}$ High          | 67   |     |      | ns            |
| $t_{WLWH}$ | $\overline{WR}$ Pulse Width Low                   | 67   |     |      | ns            |
| $t_{WLRL}$ | $\overline{WR}$ Low to RDY/ $\overline{BSY}$ Low  | 0    |     | 2.5  | $\mu\text{s}$ |
| $t_{WLRH}$ | $\overline{WR}$ Low to RDY/ $\overline{BSY}$ High | 0.5  | 0.7 | 0.9  | ms            |
| $t_{XLOL}$ | XTAL1 Low to $\overline{OE}$ Low                  | 67   |     |      | ns            |
| $t_{OLDV}$ | $\overline{OE}$ Low to DATA Valid                 |      | 20  |      | ns            |
| $t_{OHDZ}$ | $\overline{OE}$ High to DATA Tri-stated           |      |     | 20   | ns            |

## Electrical Characteristics

### Absolute Maximum Ratings

|  |                        |
|--|------------------------|
| Operating Temperature.....   | -40°C to +85/105°C     |
| Storage Temperature .....  | -65°C to +150°C        |
| Voltage on any Pin except $\overline{\text{RESET}}$ with respect to Ground ..... | -1.0V to $V_{CC}+0.5V$ |
| Maximum Operating Voltage .....  | 6.0V                   |
| Voltage on $\overline{\text{RESET}}$ with respect to Ground.....                 | -1.0V to +13.0V        |
| DC Current per I/O Pin, except PA2 .....   | 40.0 mA                |
| DC Current PA2 .....   | 60.0 mA                |
| DC Current VCC and GND Pin.....  | 300.0 mA               |

\*NOTICE: Stresses beyond those ratings listed under “Absolute Maximum Ratings” may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

### DC Characteristics – Preliminary Data

$T_A = -40^\circ\text{C}$  to  $85^\circ\text{C}$ ,  $V_{CC} = 1.8\text{V}$  to  $5.5\text{V}$  (unless otherwise noted)

| Symbol    | Parameter   | Condition  | Min                 | Typ | Max                | Units            |
|-----------|---|--|---------------------|-----|--------------------|------------------|
| $V_{IL}$  | Input Low Voltage   | (Except XTAL)  | -0.5                |     | $0.3 V_{CC}^{(1)}$ | V                |
| $V_{IL1}$ | Input Low Voltage   | XTAL   | -0.5                |     | $0.1 V_{CC}^{(1)}$ | V                |
| $V_{IH}$  | Input High Voltage  | (Except XTAL, $\overline{\text{RESET}}$ )                        | $0.6 V_{CC}^{(2)}$  |     | $V_{CC} + 0.5$     | V                |
| $V_{IH1}$ | Input High Voltage  | XTAL   | $0.7 V_{CC}^{(2)}$  |     | $V_{CC} + 0.5$     | V                |
| $V_{IH2}$ | Input High Voltage  | RESET  | $0.85 V_{CC}^{(2)}$ |     | $V_{CC} + 0.5$     | V                |
| $V_{OL}$  | Output Low Voltage <sup>(3)</sup><br>Ports A, B, D <sup>(1)</sup> | $I_{OL} = 20 \text{ mA}$ , $V_{CC} = 5\text{V}$                  |                     |     | 0.6                | V                |
|           |   | $I_{OL} = 10 \text{ mA}$ , $V_{CC} = 3\text{V}$                  |                     |     | 0.5                | V                |
| $V_{OL}$  | Output Low Voltage <sup>(3)</sup><br>Port A2 <sup>(1)</sup>       | $I_{OL} = \text{TBD}$ , $V_{CC} = 3\text{V}$                     |                     |     | TBD                | V                |
|           |   | $I_{OL} = 25 \text{ mA}$ , $V_{CC} = 1.8\text{V}$                |                     |     | TBD                | V                |
| $V_{OH}$  | Output High Voltage <sup>(4)</sup><br>Ports A, B, D               | $I_{OH} = -3 \text{ mA}$ , $V_{CC} = 5\text{V}$                  | 4.3                 |     |                    | V                |
|           |   | $I_{OH} = -1.5 \text{ mA}$ , $V_{CC} = 3\text{V}$                | 2.3                 |     |                    | V                |
| $I_{IL}$  | Input Leakage Current I/O pin                                     | $V_{CC} = 5.5\text{V}$ , pin low<br>(Absolute value)             |                     |     | 8.0                | $\mu\text{A}$    |
| $I_{IL}$  | Input Leakage Current I/O pin                                     | $V_{CC} = 5.5\text{V}$ , pin high<br>(Absolute value)            |                     |     | 8.0                | $\mu\text{A}$    |
| $R_{I/O}$ | I/O Pin Pull-up   |  | 35                  |     | 122                | $\text{k}\Omega$ |
| $I_{CC}$  | Power Supply Current  | Active Mode, $V_{CC} = 3\text{V}$ ,<br>4MHz                      |                     |     | 3.0                | mA               |
|           |   | Idle Mode $V_{CC} = 3\text{V}$ ,<br>4MHz                         |                     | 1.0 | 1.2                | mA               |
|           |   | Power-down <sup>(5)</sup> , $V_{CC} = 3\text{V}$<br>WDT enabled  |                     | 9.0 | 15.0               | $\mu\text{A}$    |
|           |   | Power-down <sup>(5)</sup> , $V_{CC} = 3\text{V}$<br>WDT disabled |                     | <1  | 2.0                | $\mu\text{A}$    |

$T_A = -40^{\circ}\text{C}$  to  $85^{\circ}\text{C}$ ,  $V_{CC} = 1.8\text{V}$  to  $5.5\text{V}$  (unless otherwise noted) (Continued)

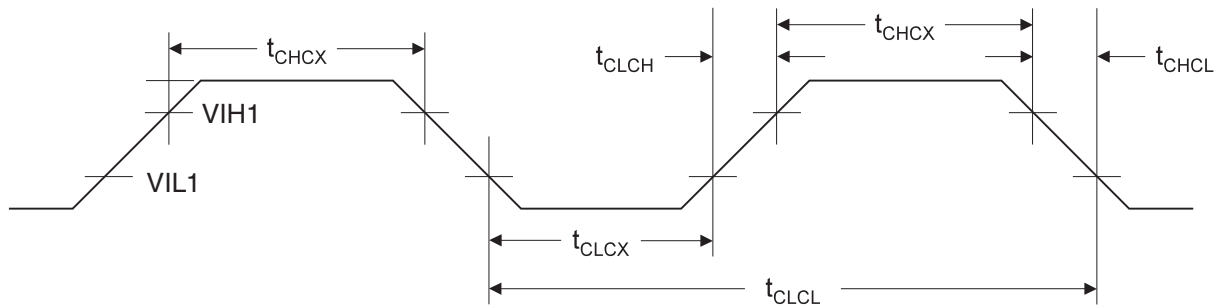
| Symbol     | Parameter                               | Condition  | Min | Typ        | Max | Units |
|------------|---|--|-----|------------|-----|-------|
| $V_{ACIO}$ | Analog Comparator Input Offset Voltage  | $V_{CC} = 5\text{V}$<br>$V_{IN} = V_{CC}/2$      |     |            | 40  | mV    |
| $I_{ACLK}$ | Analog Comparator Input Leakage Current | $V_{CC} = 5\text{V}$<br>$V_{IN} = V_{CC}/2$      | -50 |            | 50  | nA    |
| $T_{ACPD}$ | Analog Comparator Propagation Delay     | $V_{CC} = 2.7\text{V}$<br>$V_{CC} = 4.0\text{V}$ |     | 750<br>500 |     | ns    |

- Notes:
1. "Max" means the highest value where the pin is guaranteed to be read as low
  2. "Min" means the lowest value where the pin is guaranteed to be read as high
  3. Although each I/O port can sink more than the test conditions (20mA at  $V_{CC} = 5\text{V}$ , 10mA at  $V_{CC} = 3\text{V}$ ) under steady state conditions (non-transient), the following must be observed:
    - 1] The sum of all  $I_{OL}$ , for all ports, should not exceed 300 mA
    - 2] The sum of all  $I_{OL}$ , for port D0 - D7 and XTAL2 should not exceed 100 mA.
 If  $I_{OL}$  exceeds the test condition,  $V_{OL}$  may exceed the related specification.  
 Pins are not guaranteed to sink current greater than the listed test conditions.
  4. Although each I/O port can source more than the test conditions (3mA at  $V_{CC} = 5\text{V}$ , 1.5mA at  $V_{CC} = 3\text{V}$ ) under steady state conditions (non-transient), the following must be observed:
    - 1] The sum of all  $I_{OH}$ , for all ports, should not exceed 300 mA
    - 2] The sum of all  $I_{OH}$ , for port D0 - D7 and XTAL2 should not exceed 100 mA.
 If  $I_{OH}$  exceeds the test condition,  $V_{OH}$  may exceed the related specification.  
 Pins are not guaranteed to source current greater than the listed test conditions.
  5. Minimum  $V_{CC}$  for Power-down is 1.5V.



## External Clock Drive Waveforms

Figure 40. External Clock



## External Clock Drive

| Symbol       | Parameter            | $V_{CC} = 1.8V \text{ to } 2.7V$ |     | $V_{CC} = 2.7V \text{ to } 4.0V$ |     | $V_{CC} = 4.0V \text{ to } 5.5V$ |     | Units   |
|--------------|----------------------|----------------------------------|-----|----------------------------------|-----|----------------------------------|-----|---------|
|              |                      | Min                              | Max | Min                              | Max | Min                              | Max |         |
| $1/t_{CLCL}$ | Oscillator Frequency | 0                                | 1   | 0                                | 4   | 0                                | 4   | MHz     |
| $t_{CLCL}$   | Clock Period         | 1000                             |     | 250                              |     | 250                              |     | ns      |
| $t_{CHCX}$   | High Time            | 400                              |     | 100                              |     | 100                              |     | ns      |
| $t_{CLCX}$   | Low Time             | 400                              |     | 100                              |     | 100                              |     | ns      |
| $t_{CLCH}$   | Rise Time            |                                  | 1.6 |                                  | 1.6 |                                  | 0.5 | $\mu s$ |
| $t_{CHCL}$   | Fall Time            |                                  | 1.6 |                                  | 1.6 |                                  | 0.5 | $\mu s$ |

Table 26. External RC Oscillator, typical frequencies

| R [k $\Omega$ ] | C [pF] | f       |
|-----------------|--------|---------|
| 100             | 70     | 100 kHz |
| 31.5            | 20     | 1.0 MHz |
| 6.5             | 20     | 4.0 MHz |

R should be in the range 3 - 100 k $\Omega$ , and C should be at least 20pF.

## Typical Characteristics – Preliminary Data

The Following charts show typical behavior. These figures are not tested during manufacturing. All current consumption measurements are performed with all I/O pins configured as inputs and with internal pull-ups enabled. A sine wave generator with rail to rail output is used as clock source.

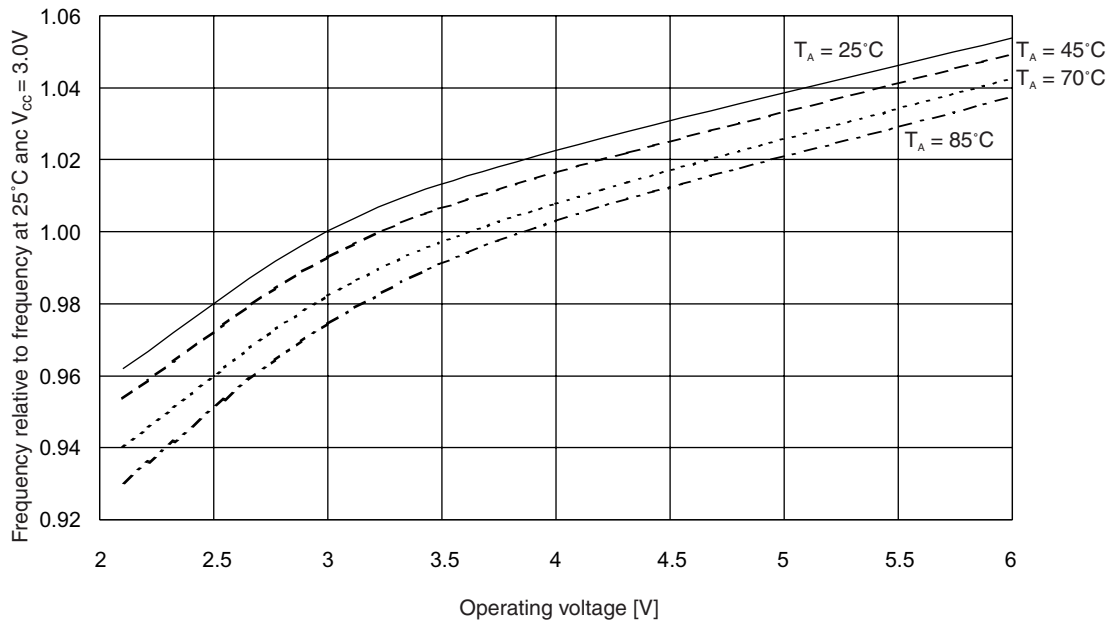
The power consumption in power-down mode is independent of clock selection.

The current consumption is a function of several factors such as: Operating voltage, Operating frequency, Loading of I/O pins, Switching rate of I/O pins, Code executed and Ambient temperature. The dominating factors are operating voltage and frequency. The current drawn from capacitive loaded pins may be estimated (for one pin) as  $C_L * V_{CC} * f$ , where  $C_L$  = Load Capacitance,  $V_{CC}$  = Operating Voltage and  $f$  = Average Switching Frequency of I/O pin.

The parts are characterized at frequencies and voltages higher than test limits. Parts are not guaranteed to function properly at frequencies and voltages higher than the ordering code indicates.

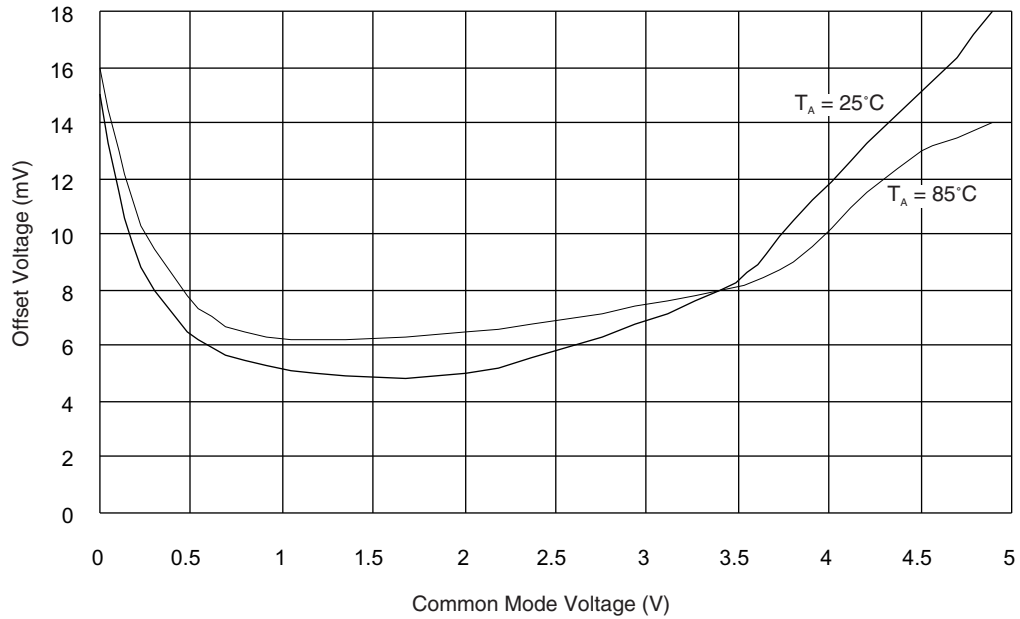
The difference between current consumption in Power-down mode with Watchdog timer enabled and Power-down mode with Watchdog timer disabled represents the differential current drawn by the Watchdog timer.

**Figure 41.** Calibrated Internal RC Oscillator Frequency vs.  $V_{CC}$

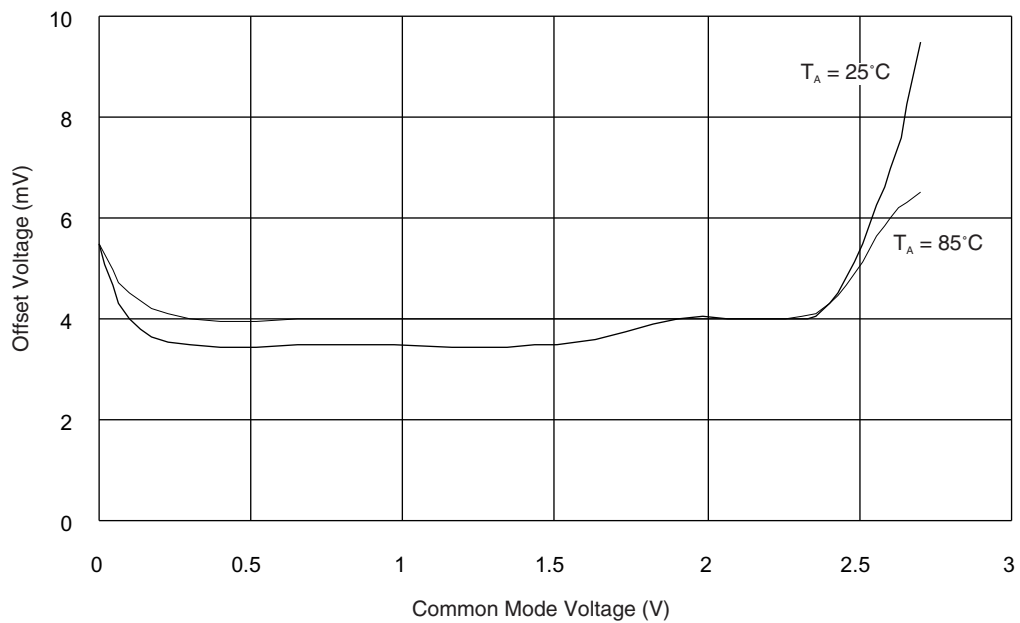


Analog comparator offset voltage is measured as absolute offset

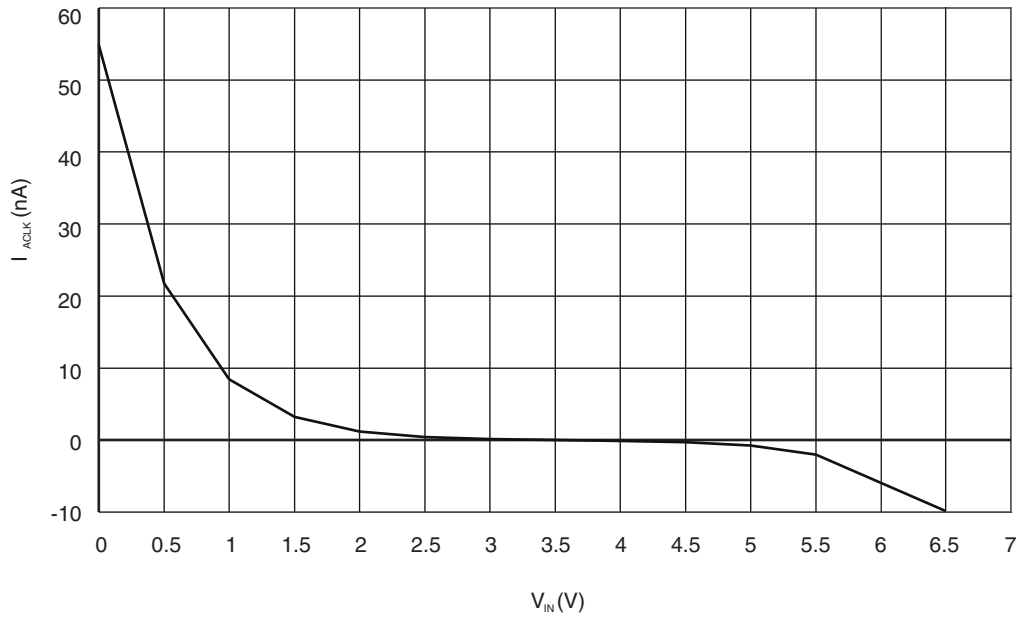
**Figure 42.** Analog Comparator Offset Voltage vs. Common Mode Voltage ( $V_{CC}=5V$ )



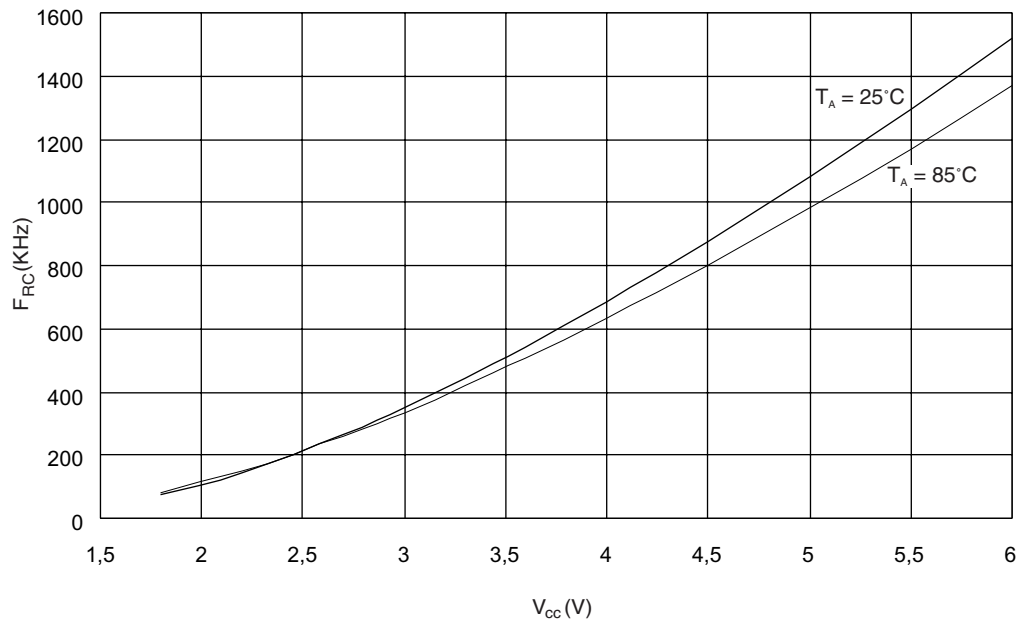
**Figure 43.** Analog Comparator Offset Voltage vs. Common Mode Voltage ( $V_{CC}=2.7V$ )



**Figure 44.** Analog Comparator Input Leakage Current ( $V_{CC} = 6V$ ;  $T_A = 25^\circ C$ )

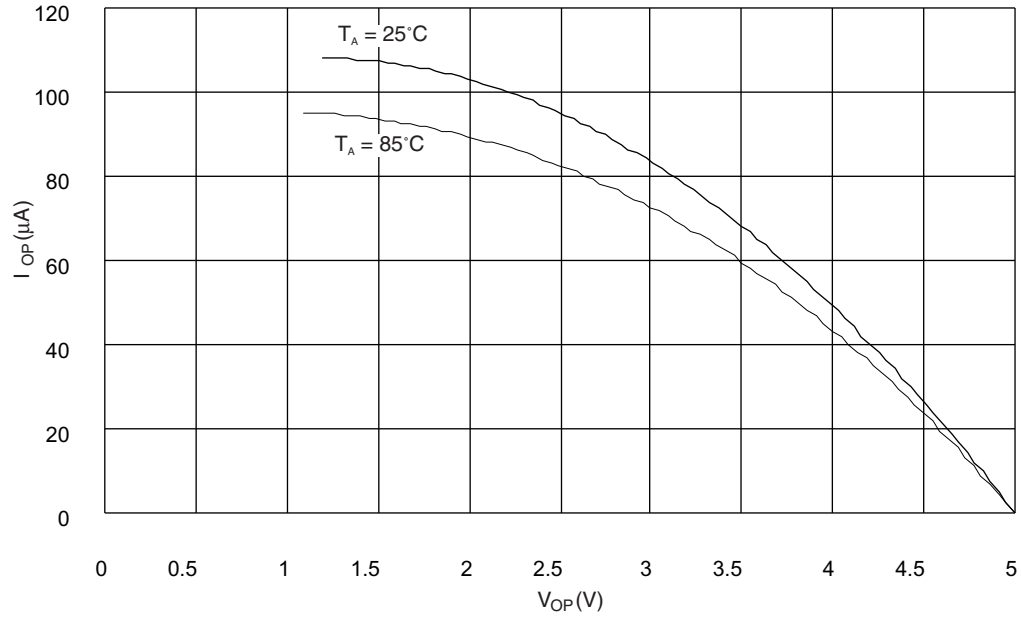


**Figure 45.** Watchdog Oscillator Frequency vs.  $V_{CC}$

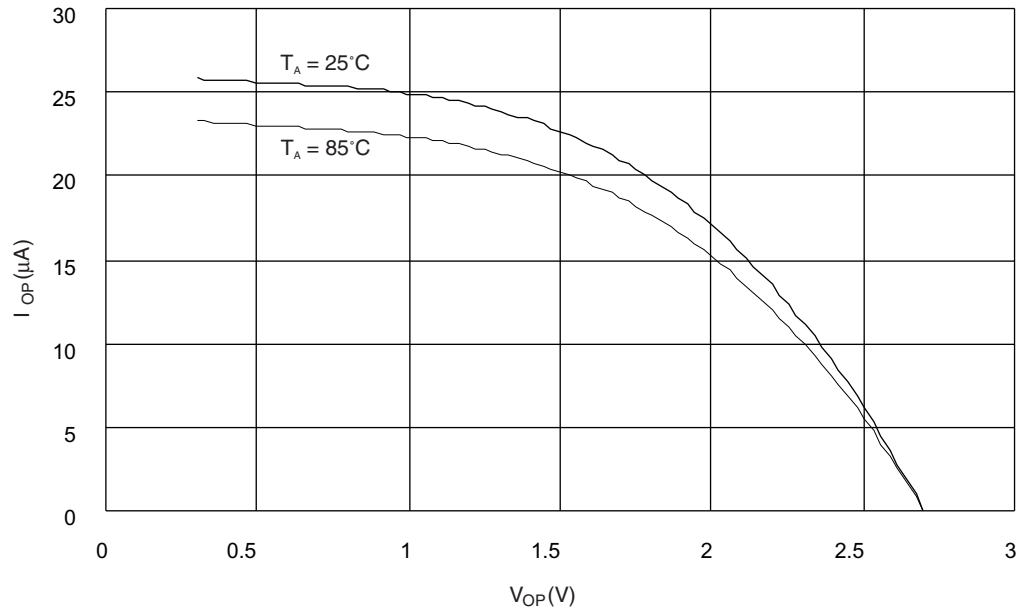


Sink and source capabilities of I/O ports are measured on one pin at a time.

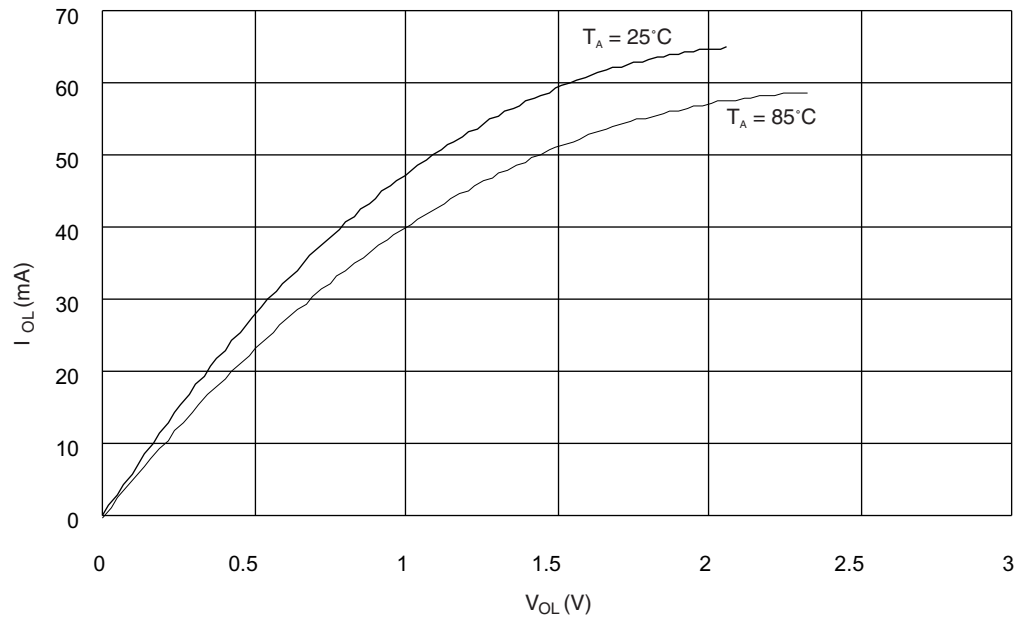
**Figure 46.** Pull-up Resistor Current vs. Input Voltage ( $V_{CC}=5V$ )



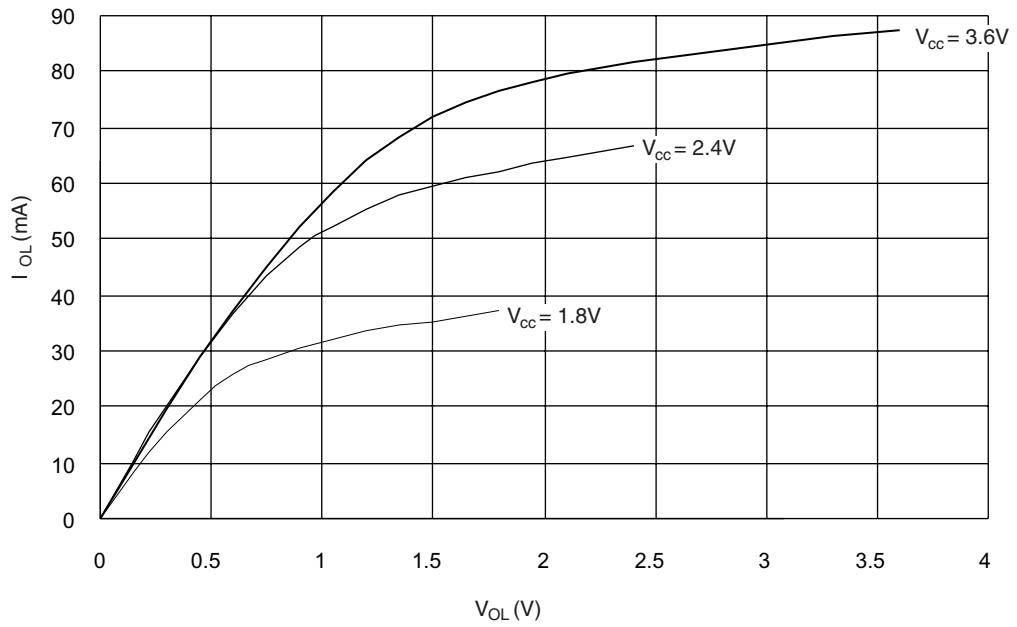
**Figure 47.** Pull-up Resistor Current vs. Input Voltage ( $V_{CC}=2.7V$ )



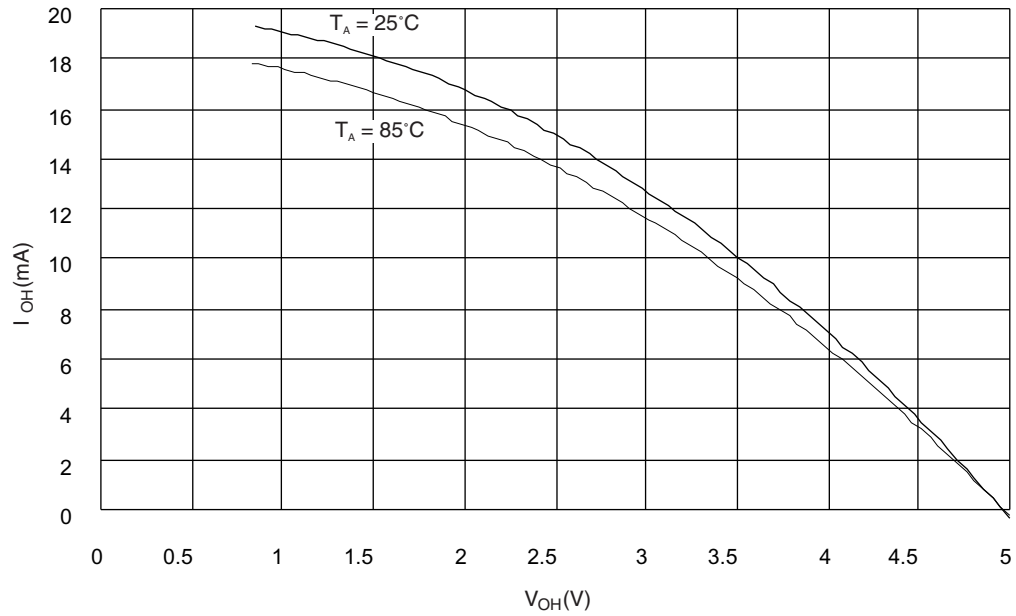
**Figure 48.** I/O Pin Sink Current vs. Output Voltage. All pins except PA2 ( $V_{CC}=5V$ )



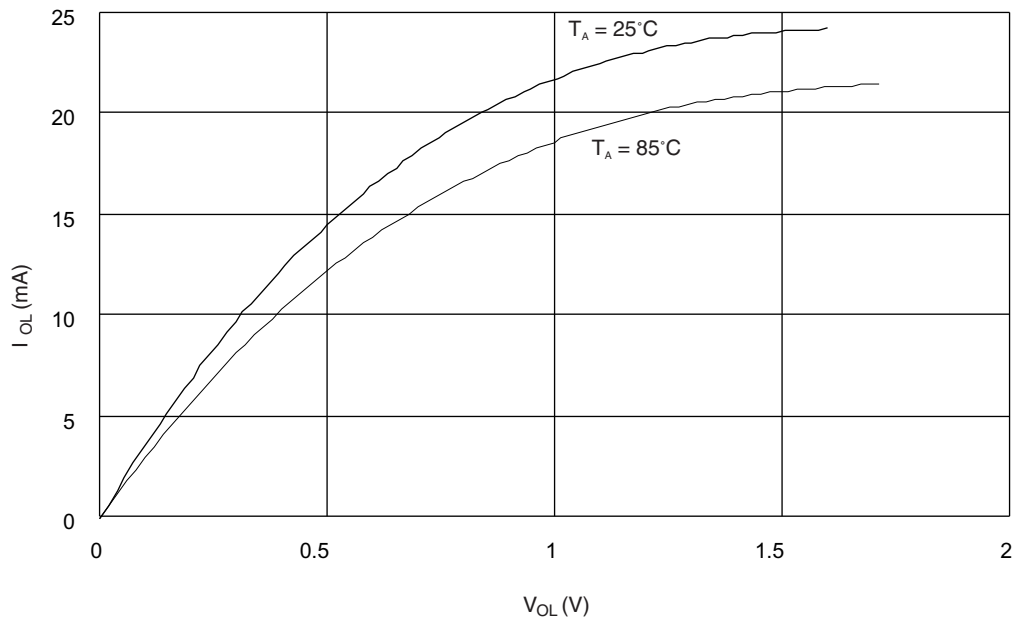
**Figure 49.** PA2 I/O Pin Sink Current vs. Output Voltage ( High current pin PA2;  $T_A=25^\circ C$ )



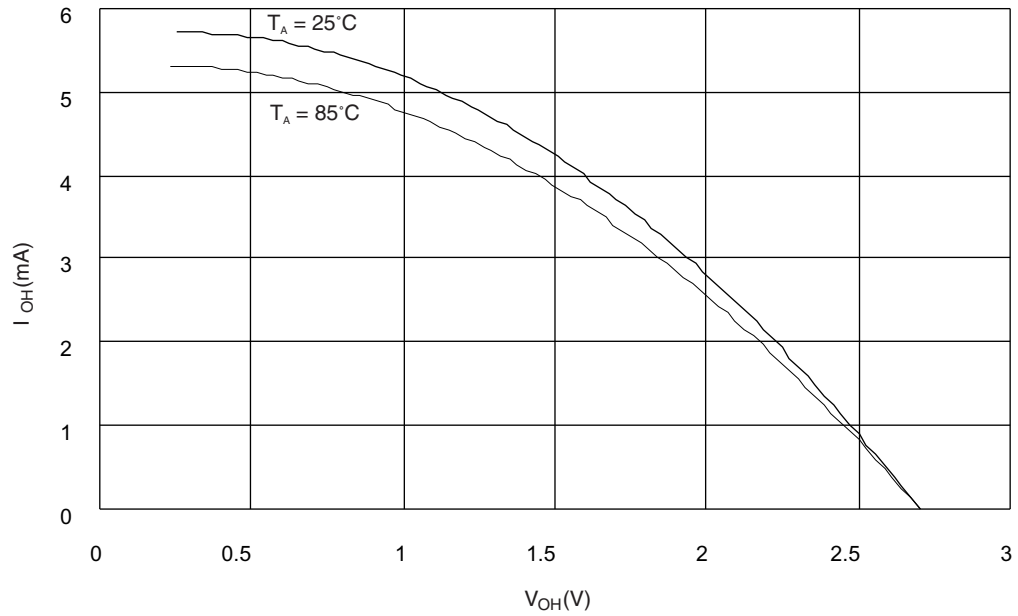
**Figure 50.** I/O Pin Source Current vs. Output voltage ( $V_{CC}= 5V$ )



**Figure 51.** I/O Pin Sink Current vs. Output Voltage. All pins except PA2 ( $V_{CC}= 2.7V$ ).



**Figure 52.** I/O Pin source Current vs. Output Voltage ( $V_{CC}= 2.7V$ )



**Figure 53.** I/O Pin Input Threshold Voltage vs.  $V_{CC}$  ( $T_A= 25^\circ C$ )

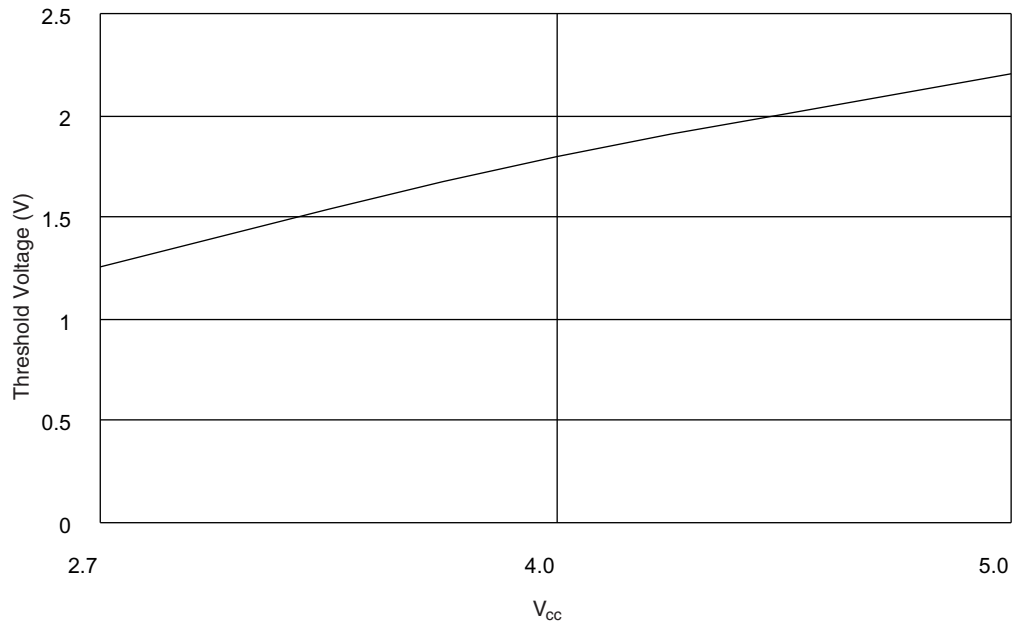
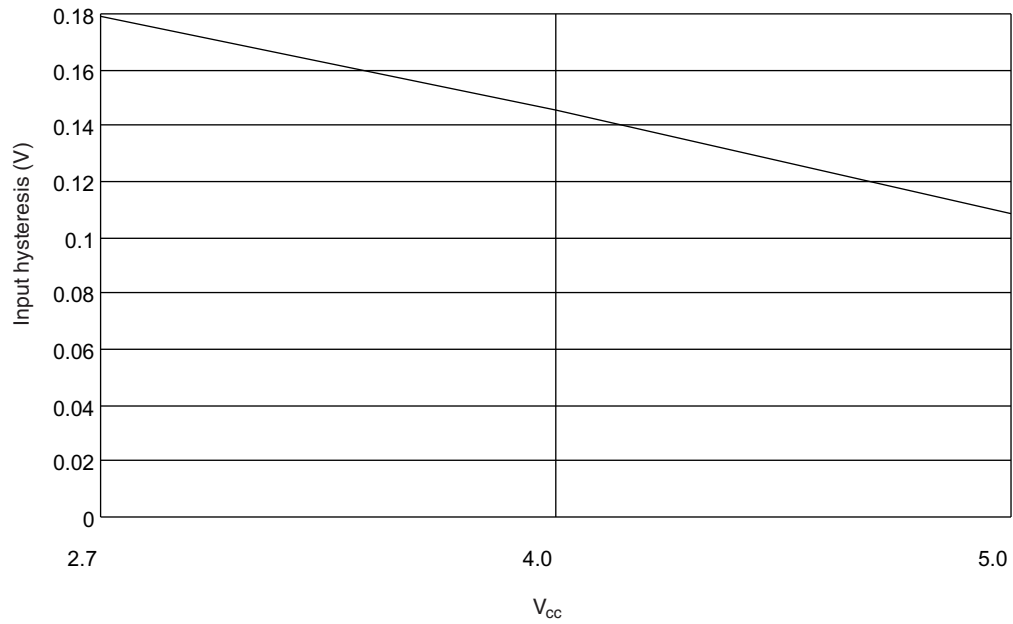




Figure 54. I/O Pin Input Hysteresis vs.  $V_{CC}$  ( $T_A = 25^\circ\text{C}$ )





## Register Summary

| Address | Name     | Bit 7                           | Bit 6  | Bit 5  | Bit 4  | Bit 3  | Bit 2  | Bit 1  | Bit 0  | Page    |
|---------|----------|---------------------------------|--------|--------|--------|--------|--------|--------|--------|---------|
| \$3F    | SREG     | I                               | T      | H      | S      | V      | N      | Z      | C      | page 12 |
| \$3E    | Reserved |                                 |        |        |        |        |        |        |        |         |
| ...     | Reserved |                                 |        |        |        |        |        |        |        |         |
| \$20    | Reserved |                                 |        |        |        |        |        |        |        |         |
| \$1F    | Reserved |                                 |        |        |        |        |        |        |        |         |
| \$1E    | Reserved |                                 |        |        |        |        |        |        |        |         |
| \$1D    | Reserved |                                 |        |        |        |        |        |        |        |         |
| \$1C    | Reserved |                                 |        |        |        |        |        |        |        |         |
| \$1B    | PORTA    | -                               | -      | -      | -      | PORTA3 | PORTA2 | PORTA1 | PORTA0 | page 32 |
| \$1A    | PACR     | -                               | -      | -      | -      | DDA3   | PA2HC  | DDA1   | DDA0   | page 32 |
| \$19    | PINA     | -                               | -      | -      | -      | PINA3  | -      | PINA1  | PINA0  | page 33 |
| \$18    | Reserved |                                 |        |        |        |        |        |        |        |         |
| \$17    | Reserved |                                 |        |        |        |        |        |        |        |         |
| \$16    | PINB     | PINB7                           | PINB6  | PINB5  | PINB4  | PINB3  | PINB2  | PINB1  | PINB0  | page 35 |
| \$15    | Reserved |                                 |        |        |        |        |        |        |        |         |
| \$14    | Reserved |                                 |        |        |        |        |        |        |        |         |
| \$13    | Reserved |                                 |        |        |        |        |        |        |        |         |
| \$12    | PORTD    | PORTD7                          | PORTD6 | PORTD5 | PORTD4 | PORTD3 | PORTD2 | PORTD1 | PORTD0 | page 38 |
| \$11    | DDRD     | DDD7                            | DDD6   | DDD5   | DDD4   | DDD3   | DDD2   | DDD1   | DDD0   | page 38 |
| \$10    | PIND     | PIND7                           | PIND6  | PIND5  | PIND4  | PIND3  | PIND2  | PIND1  | PIND0  | page 38 |
| \$0F    | Reserved |                                 |        |        |        |        |        |        |        |         |
| \$0E    | Reserved |                                 |        |        |        |        |        |        |        |         |
| \$0D    | Reserved |                                 |        |        |        |        |        |        |        |         |
| \$0C    | Reserved |                                 |        |        |        |        |        |        |        |         |
| \$0B    | Reserved |                                 |        |        |        |        |        |        |        |         |
| \$0A    | Reserved |                                 |        |        |        |        |        |        |        |         |
| \$09    | Reserved |                                 |        |        |        |        |        |        |        |         |
| \$08    | ACSR     | ACD                             | -      | ACO    | ACI    | ACIE   | -      | ACIS1  | ACIS0  | page 31 |
| \$07    | MCUCS    | PLUPB                           | -      | SE     | SM     | WDRF   | -      | EXTRF  | PORF   | page 18 |
| \$06    | ICR      | INT1                            | INT0   | LLIE   | TOIE0  | ISC11  | ISC10  | ISC01  | ISC00  | page 19 |
| \$05    | IFR      | INTF1                           | INTF0  | -      | TOV0   | -      | -      | -      | -      | page 20 |
| \$04    | TCCR0    | FOV0                            | -      | -      | OOM01  | OOM00  | CS02   | CS01   | CS00   | page 23 |
| \$03    | TCNT0    | Timer/Counter0 (8 BIT)          |        |        |        |        |        |        |        | page 24 |
| \$02    | MODCR    | ONTIM4                          | ONTIM3 | ONTIM2 | ONTIM1 | ONTIM0 | MCONF2 | MCONF1 | MCONF0 | page 27 |
| \$01    | WDTCSR   | -                               | -      | -      | WDTOE  | WDE    | WDP2   | WDP1   | WDP0   | page 25 |
| \$00    | OSCCAL   | Oscillator Calibration Register |        |        |        |        |        |        |        | page 26 |

- Notes:
1. For compatibility with future devices, reserved bits should be written to zero if accessed. Reserved I/O memory addresses should never be written.
  2. Some of the status flags are cleared by writing a logical one to them. Note that the CBI and SBI instructions will operate on all bits in the I/O register, writing a one back into any flag read as set, thus clearing the flag. The CBI and SBI instructions work with registers \$00 to \$1F only.

## Instruction Set Summary

| Mnemonics                                | Operands | Description                            | Operation   | Flags         | #Clocks |
|--|----------|--|---|---------------|---------|
| <b>ARITHMETIC AND LOGIC INSTRUCTIONS</b> |          |  |   |               |         |
| ADD                                      | Rd, Rr   | Add two Registers                      | $Rd \leftarrow Rd + Rr$                               | Z,C,N,V,H     | 1       |
| ADC                                      | Rd, Rr   | Add with Carry two Registers           | $Rd \leftarrow Rd + Rr + C$                           | Z,C,N,V,H     | 1       |
| SUB                                      | Rd, Rr   | Subtract two Registers                 | $Rd \leftarrow Rd - Rr$                               | Z,C,N,V,H     | 1       |
| SUBI                                     | Rd, K    | Subtract Constant from Register        | $Rd \leftarrow Rd - K$                                | Z,C,N,V,H     | 1       |
| SBC                                      | Rd, Rr   | Subtract with Carry two Registers      | $Rd \leftarrow Rd - Rr - C$                           | Z,C,N,V,H     | 1       |
| SBCI                                     | Rd, K    | Subtract with Carry Constant from Reg. | $Rd \leftarrow Rd - K - C$                            | Z,C,N,V,H     | 1       |
| AND                                      | Rd, Rr   | Logical AND Registers                  | $Rd \leftarrow Rd \bullet Rr$                         | Z,N,V         | 1       |
| ANDI                                     | Rd, K    | Logical AND Register and Constant      | $Rd \leftarrow Rd \bullet K$                          | Z,N,V         | 1       |
| OR                                       | Rd, Rr   | Logical OR Registers                   | $Rd \leftarrow Rd \vee Rr$                            | Z,N,V         | 1       |
| ORI                                      | Rd, K    | Logical OR Register and Constant       | $Rd \leftarrow Rd \vee K$                             | Z,N,V         | 1       |
| EOR                                      | Rd, Rr   | Exclusive OR Registers                 | $Rd \leftarrow Rd \oplus Rr$                          | Z,N,V         | 1       |
| COM                                      | Rd       | One's Complement                       | $Rd \leftarrow \$FF - Rd$                             | Z,C,N,V       | 1       |
| NEG                                      | Rd       | Two's Complement                       | $Rd \leftarrow \$00 - Rd$                             | Z,C,N,V,H     | 1       |
| SBR                                      | Rd, K    | Set Bit(s) in Register                 | $Rd \leftarrow Rd \vee K$                             | Z,N,V         | 1       |
| CBR                                      | Rd, K    | Clear Bit(s) in Register               | $Rd \leftarrow Rd \bullet (\text{FFh} - K)$           | Z,N,V         | 1       |
| INC                                      | Rd       | Increment                              | $Rd \leftarrow Rd + 1$                                | Z,N,V         | 1       |
| DEC                                      | Rd       | Decrement                              | $Rd \leftarrow Rd - 1$                                | Z,N,V         | 1       |
| TST                                      | Rd       | Test for Zero or Minus                 | $Rd \leftarrow Rd \bullet Rd$                         | Z,N,V         | 1       |
| CLR                                      | Rd       | Clear Register                         | $Rd \leftarrow Rd \oplus Rd$                          | Z,N,V         | 1       |
| SER                                      | Rd       | Set Register                           | $Rd \leftarrow \$FF$                                  | None          | 1       |
| <b>BRANCH INSTRUCTIONS</b>               |          |  |   |               |         |
| RJMP                                     | k        | Relative Jump                          | $PC \leftarrow PC + k + 1$                            | None          | 2       |
| RCALL                                    | k        | Relative Subroutine Call               | $PC \leftarrow PC + k + 1$                            | None          | 3       |
| RET                                      |          | Subroutine Return                      | $PC \leftarrow \text{STACK}$                          | None          | 4       |
| RETI                                     |          | Interrupt Return                       | $PC \leftarrow \text{STACK}$                          | I             | 4       |
| CPSE                                     | Rd,Rr    | Compare, Skip if Equal                 | if (Rd = Rr) $PC \leftarrow PC + 2$ or 3              | None          | 1 / 2   |
| CP                                       | Rd,Rr    | Compare                                | $Rd - Rr$   | Z, N, V, C, H | 1       |
| CPC                                      | Rd,Rr    | Compare with Carry                     | $Rd - Rr - C$   | Z, N, V, C, H | 1       |
| CPI                                      | Rd,K     | Compare Register with Immediate        | $Rd - K$  | Z, N, V, C, H | 1       |
| SBRC                                     | Rr, b    | Skip if Bit in Register Cleared        | if (Rr(b)=0) $PC \leftarrow PC + 2$ or 3              | None          | 1 / 2   |
| SBRS                                     | Rr, b    | Skip if Bit in Register is Set         | if (Rr(b)=1) $PC \leftarrow PC + 2$ or 3              | None          | 1 / 2   |
| SBIC                                     | P, b     | Skip if Bit in I/O Register Cleared    | if (P(b)=0) $PC \leftarrow PC + 2$ or 3               | None          | 1 / 2   |
| SBIS                                     | P, b     | Skip if Bit in I/O Register is Set     | if (P(b)=1) $PC \leftarrow PC + 2$ or 3               | None          | 1 / 2   |
| BRBS                                     | s, k     | Branch if Status Flag Set              | if (SREG(s) = 1) then $PC \leftarrow PC + k + 1$      | None          | 1 / 2   |
| BRBC                                     | s, k     | Branch if Status Flag Cleared          | if (SREG(s) = 0) then $PC \leftarrow PC + k + 1$      | None          | 1 / 2   |
| BREQ                                     | k        | Branch if Equal                        | if (Z = 1) then $PC \leftarrow PC + k + 1$            | None          | 1 / 2   |
| BRNE                                     | k        | Branch if Not Equal                    | if (Z = 0) then $PC \leftarrow PC + k + 1$            | None          | 1 / 2   |
| BRCS                                     | k        | Branch if Carry Set                    | if (C = 1) then $PC \leftarrow PC + k + 1$            | None          | 1 / 2   |
| BRCC                                     | k        | Branch if Carry Cleared                | if (C = 0) then $PC \leftarrow PC + k + 1$            | None          | 1 / 2   |
| BRSH                                     | k        | Branch if Same or Higher               | if (C = 0) then $PC \leftarrow PC + k + 1$            | None          | 1 / 2   |
| BRLO                                     | k        | Branch if Lower                        | if (C = 1) then $PC \leftarrow PC + k + 1$            | None          | 1 / 2   |
| BRMI                                     | k        | Branch if Minus                        | if (N = 1) then $PC \leftarrow PC + k + 1$            | None          | 1 / 2   |
| BRPL                                     | k        | Branch if Plus                         | if (N = 0) then $PC \leftarrow PC + k + 1$            | None          | 1 / 2   |
| BRGE                                     | k        | Branch if Greater or Equal, Signed     | if (N $\oplus$ V = 0) then $PC \leftarrow PC + k + 1$ | None          | 1 / 2   |
| BRLT                                     | k        | Branch if Less Than Zero, Signed       | if (N $\oplus$ V = 1) then $PC \leftarrow PC + k + 1$ | None          | 1 / 2   |
| BRHS                                     | k        | Branch if Half Carry Flag Set          | if (H = 1) then $PC \leftarrow PC + k + 1$            | None          | 1 / 2   |
| BRHC                                     | k        | Branch if Half Carry Flag Cleared      | if (H = 0) then $PC \leftarrow PC + k + 1$            | None          | 1 / 2   |
| BRTS                                     | k        | Branch if T Flag Set                   | if (T = 1) then $PC \leftarrow PC + k + 1$            | None          | 1 / 2   |
| BRTC                                     | k        | Branch if T Flag Cleared               | if (T = 0) then $PC \leftarrow PC + k + 1$            | None          | 1 / 2   |
| BRVS                                     | k        | Branch if Overflow Flag is Set         | if (V = 1) then $PC \leftarrow PC + k + 1$            | None          | 1 / 2   |
| BRVC                                     | k        | Branch if Overflow Flag is Cleared     | if (V = 0) then $PC \leftarrow PC + k + 1$            | None          | 1 / 2   |
| BRIE                                     | k        | Branch if Interrupt Enabled            | if (I = 1) then $PC \leftarrow PC + k + 1$            | None          | 1 / 2   |
| BRID                                     | k        | Branch if Interrupt Disabled           | if (I = 0) then $PC \leftarrow PC + k + 1$            | None          | 1 / 2   |



## Instruction Set Summary (Continued)

| Mnemonics                            | Operands | Description                    | Operation  | Flags   | #Clocks |
|--------------------------------------|----------|--------------------------------|--|---------|---------|
| <b>DATA TRANSFER INSTRUCTIONS</b>    |          |                                |  |         |         |
| LD                                   | Rd,Z     | Load Register Indirect         | $Rd \leftarrow (Z)$  | None    | 2       |
| ST                                   | Z,Rr     | Store Register Indirect        | $(Z) \leftarrow Rr$  | None    | 2       |
| MOV                                  | Rd, Rr   | Move Between Registers         | $Rd \leftarrow Rr$   | None    | 1       |
| LDI                                  | Rd, K    | Load Immediate                 | $Rd \leftarrow K$  | None    | 1       |
| IN                                   | Rd, P    | In Port                        | $Rd \leftarrow P$  | None    | 1       |
| OUT                                  | P, Rr    | Out Port                       | $P \leftarrow Rr$  | None    | 1       |
| LPM                                  |          | Load Program Memory            | $R0 \leftarrow (Z)$  | None    | 3       |
| <b>BIT AND BIT-TEST INSTRUCTIONS</b> |          |                                |  |         |         |
| SBI                                  | Pb       | Set Bit in I/O Register        | $I/O(P,b) \leftarrow 1$  | None    | 2       |
| CBI                                  | Pb       | Clear Bit in I/O Register      | $I/O(P,b) \leftarrow 0$  | None    | 2       |
| LSL                                  | Rd       | Logical Shift Left             | $Rd(n+1) \leftarrow Rd(n), Rd(0) \leftarrow 0$                     | Z,C,N,V | 1       |
| LSR                                  | Rd       | Logical Shift Right            | $Rd(n) \leftarrow Rd(n+1), Rd(7) \leftarrow 0$                     | Z,C,N,V | 1       |
| ROL                                  | Rd       | Rotate Left Through Carry      | $Rd(0) \leftarrow C, Rd(n+1) \leftarrow Rd(n), C \leftarrow Rd(7)$ | Z,C,N,V | 1       |
| ROR                                  | Rd       | Rotate Right Through Carry     | $Rd(7) \leftarrow C, Rd(n) \leftarrow Rd(n+1), C \leftarrow Rd(0)$ | Z,C,N,V | 1       |
| ASR                                  | Rd       | Arithmetic Shift Right         | $Rd(n) \leftarrow Rd(n+1), n=0..6$                                 | Z,C,N,V | 1       |
| SWAP                                 | Rd       | Swap Nibbles                   | $Rd(3..0) \leftarrow Rd(7..4), Rd(7..4) \leftarrow Rd(3..0)$       | None    | 1       |
| BSET                                 | s        | Flag Set                       | $SREG(s) \leftarrow 1$   | SREG(s) | 1       |
| BCLR                                 | s        | Flag Clear                     | $SREG(s) \leftarrow 0$   | SREG(s) | 1       |
| BST                                  | Rr, b    | Bit Store from Register to T   | $T \leftarrow Rr(b)$   | T       | 1       |
| BLD                                  | Rd, b    | Bit load from T to Register    | $Rd(b) \leftarrow T$   | None    | 1       |
| SEC                                  |          | Set Carry                      | $C \leftarrow 1$   | C       | 1       |
| CLC                                  |          | Clear Carry                    | $C \leftarrow 0$   | C       | 1       |
| SEN                                  |          | Set Negative Flag              | $N \leftarrow 1$   | N       | 1       |
| CLN                                  |          | Clear Negative Flag            | $N \leftarrow 0$   | N       | 1       |
| SEZ                                  |          | Set Zero Flag                  | $Z \leftarrow 1$   | Z       | 1       |
| CLZ                                  |          | Clear Zero Flag                | $Z \leftarrow 0$   | Z       | 1       |
| SEI                                  |          | Global Interrupt Enable        | $I \leftarrow 1$   | I       | 1       |
| CLI                                  |          | Global Interrupt Disable       | $I \leftarrow 0$   | I       | 1       |
| SES                                  |          | Set Signed Test Flag           | $S \leftarrow 1$   | S       | 1       |
| CLS                                  |          | Clear Signed Test Flag         | $S \leftarrow 0$   | S       | 1       |
| SEV                                  |          | Set Twos Complement Overflow   | $V \leftarrow 1$   | V       | 1       |
| CLV                                  |          | Clear Twos Complement Overflow | $V \leftarrow 0$   | V       | 1       |
| SET                                  |          | Set T in SREG                  | $T \leftarrow 1$   | T       | 1       |
| CLT                                  |          | Clear T in SREG                | $T \leftarrow 0$   | T       | 1       |
| SEH                                  |          | Set Half Carry Flag in SREG    | $H \leftarrow 1$   | H       | 1       |
| CLH                                  |          | Clear Half Carry Flag in SREG  | $H \leftarrow 0$   | H       | 1       |
| NOP                                  |          | No Operation                   |  | None    | 1       |
| SLEEP                                |          | Sleep                          | (see specific descr. for Sleep function)                           | None    | 3       |
| WDR                                  |          | Watch Dog Reset                | (see specific descr. for WDR/timer)                                | None    | 1       |

## Ordering Information

| Speed (MHz) | Power Supply (Volts) | Ordering Code | Package | Operation Range               |
|-------------|----------------------|---------------|---------|-------------------------------|
| 4           | 2.7 - 5.5            | ATtiny28L-4AC | 32A     | Commercial<br>(0°C to 70°C)   |
|             |                      | ATtiny28L-4PC | 28P3    | Commercial<br>(0°C to 70°C)   |
|             |                      | ATtiny28L-4AI | 32A     | Industrial<br>(-40°C to 85°C) |
|             |                      | ATtiny28L-4PI | 28P3    | Industrial<br>(-40°C to 85°C) |
| 1           | 1.8 - 5.5            | ATtiny28V-1AC | 32A     | Commercial<br>(0°C to 70°C)   |
|             |                      | ATtiny28V-1PC | 28P3    | Commercial<br>(0°C to 70°C)   |
|             |                      | ATtiny28V-1AI | 32A     | Industrial<br>(-40°C to 85°C) |
|             |                      | ATtiny28V-1PI | 28P3    | Industrial<br>(-40°C to 85°C) |

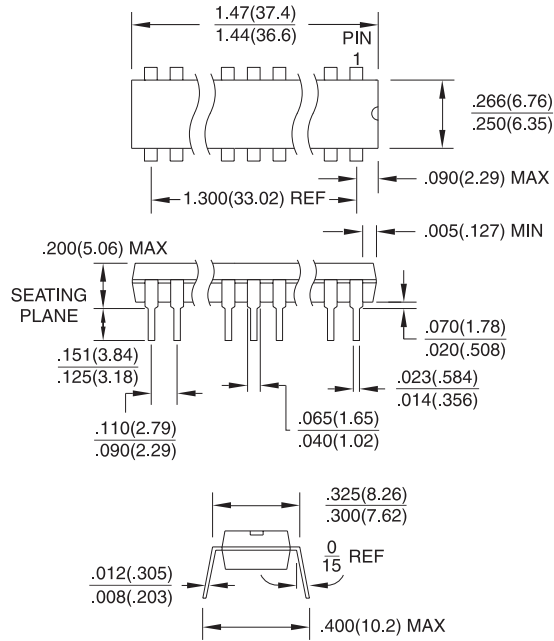
| Package Type |   |
|--------------|---|
| <b>28P3</b>  | 28-pin, 0.300" Wide, Plastic Dual Inline Package (PDIP)           |
| <b>32A</b>   | 32-lead, Thin (1.0 mm) Plastic Gull-Wing Quad Flat Package (TQFP) |



## Packaging Information

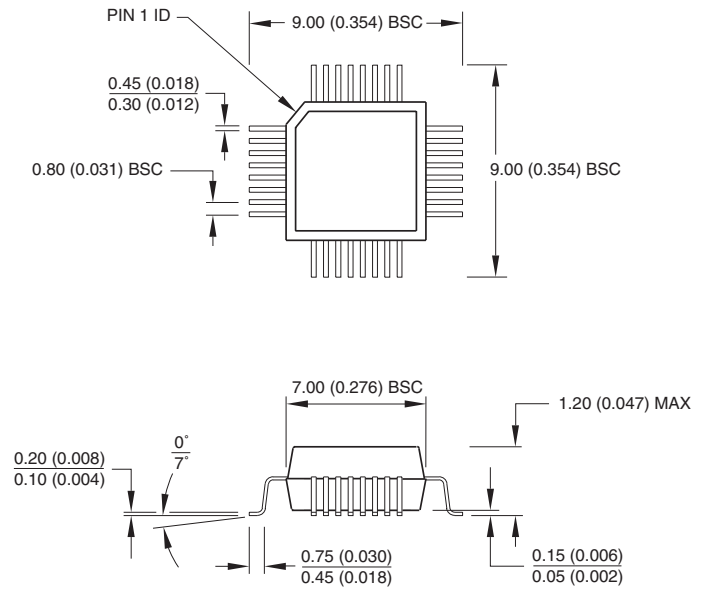
**28P3**, 28-pin, 0.300" Wide, Plastic Dual Inline Package (PDIP)

Dimensions in Inches and (Millimeters)



**32A**, 32-lead, Thin (1.0 mm) Plastic Gull-Wing Quad Flat Package (TQFP)

Dimensions in Millimeters and (Inches)\*  
 JEDEC STANDARD MS-026 ACB



\*Controlling dimension: millimeters



## Atmel Headquarters

### *Corporate Headquarters*

2325 Orchard Parkway  
San Jose, CA 95131  
TEL (408) 441-0311  
FAX (408) 487-2600

### *Europe*

Atmel U.K., Ltd.  
Coliseum Business Centre  
Riverside Way  
Camberley, Surrey GU15 3YL  
England  
TEL (44) 1276-686-677  
FAX (44) 1276-686-697

### *Asia*

Atmel Asia, Ltd.  
Room 1219  
Chinachem Golden Plaza  
77 Mody Road Tsimhatsui  
East Kowloon  
Hong Kong  
TEL (852) 2721-9778  
FAX (852) 2722-1369

### *Japan*

Atmel Japan K.K.  
9F, Tonetsu Shinkawa Bldg.  
1-24-8 Shinkawa  
Chuo-ku, Tokyo 104-0033  
Japan  
TEL (81) 3-3523-3551  
FAX (81) 3-3523-7581

## Atmel Operations

### *Atmel Colorado Springs*

1150 E. Cheyenne Mtn. Blvd.  
Colorado Springs, CO 80906  
TEL (719) 576-3300  
FAX (719) 540-1759

### *Atmel Rousset*

Zone Industrielle  
13106 Rousset Cedex  
France  
TEL (33) 4-4253-6000  
FAX (33) 4-4253-6001

---

### *Fax-on-Demand*

North America:  
1-(800) 292-8635  
International:  
1-(408) 441-0732

### *e-mail*

literature@atmel.com

### *Web Site*

<http://www.atmel.com>

### *BBS*

1-(408) 436-4309

### © Atmel Corporation 2000.

Atmel Corporation makes no warranty for the use of its products, other than those expressly contained in the Company's standard warranty which is detailed in Atmel's Terms and Conditions located on the Company's web site. The Company assumes no responsibility for any errors which may appear in this document, reserves the right to change devices or specifications detailed herein at any time without notice, and does not make any commitment to update the information contained herein. No licenses to patents or other intellectual property of Atmel are granted by the Company in connection with the sale of Atmel products, expressly or by implication. Atmel's products are not authorized for use as critical components in life support devices or systems.

Marks bearing ® and/or ™ are registered trademarks and trademarks of Atmel Corporation.

Terms and product names in this document may be trademarks of others.



Printed on recycled paper.

1062C-05/00/xM