## Features

- Utilizes the AVR® RISC Architecture
- AVR – High-performance and Low-power RISC Architecture
  - 118 Powerful Instructions – Most Single Clock Cycle Execution
  - 32 x 8 General-purpose Working Registers
  - Up to 8 MIPS Throughput at 8 MHz
- Data and Nonvolatile Program Memory
  - 8K Bytes of In-System Programmable Flash
    Endurance: 1,000 Write/Erase Cycles
  - 512 Bytes of SRAM
  - 512 Bytes of In-System Programmable EEPROM
    Endurance: 100,000 Write/Erase Cycles
  - Programming Lock for Flash Program and EEPROM Data Security
- Peripheral Features
  - One 8-bit Timer/Counter with Separate Prescaler
  - One 16-bit Timer/Counter with Separate Prescaler
    Compare, Capture Modes and Dual 8-, 9-, or 10-bit PWM
  - On-chip Analog Comparator
  - Programmable Watchdog Timer with On-chip Oscillator
  - Programmable Serial UART
  - Master/Slave SPI Serial Interface
- Special Microcontroller Features
  - Low-power Idle and Power-down Modes
  - External and Internal Interrupt Sources
- Specifications
  - Low-power, High-speed CMOS Process Technology
  - Fully Static Operation
- Power Consumption at 4 MHz, 3V, 25°C
  - Active: 3.0 mA
  - Idle Mode: 1.0 mA
  - Power-down Mode: <1 µA
- I/O and Packages
  - 32 Programmable I/O Lines
  - 40-lead PDIP, 44-lead PLCC and TQFP
- Operating Voltages
  - 2.7 - 6.0V for AT90S8515-4
  - 4.0 - 6.0V for AT90S8515-8
- Speed Grades
  - 0 - 4 MHz for AT90S8515-4
  - 0 - 8 MHz for AT90S8515-8

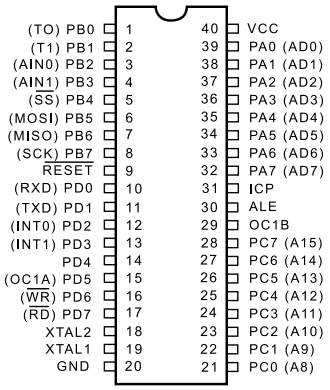
# 8-bit AVR® Microcontroller with 8K Bytes In-System Programmable Flash

# AT90S8515

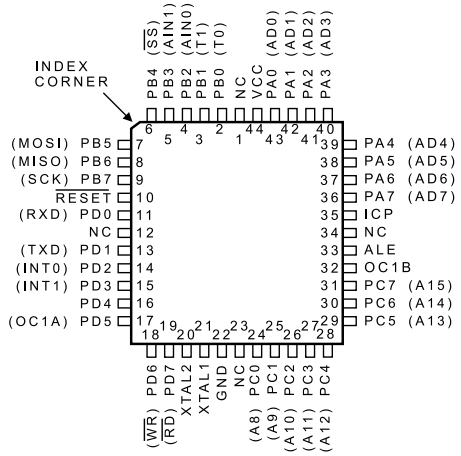
Rev. 0841G–09/01

## Pin Configurations

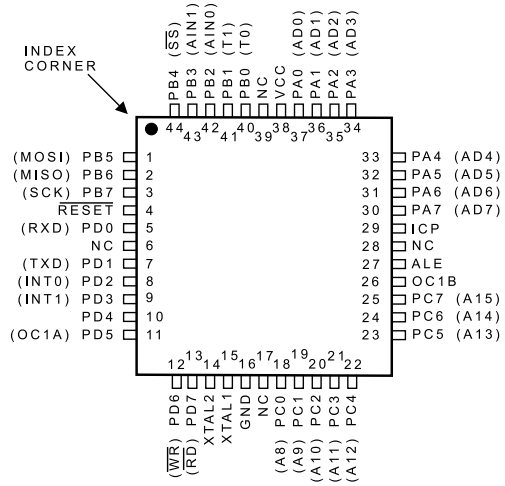
PDIP




PLCC




TQFP

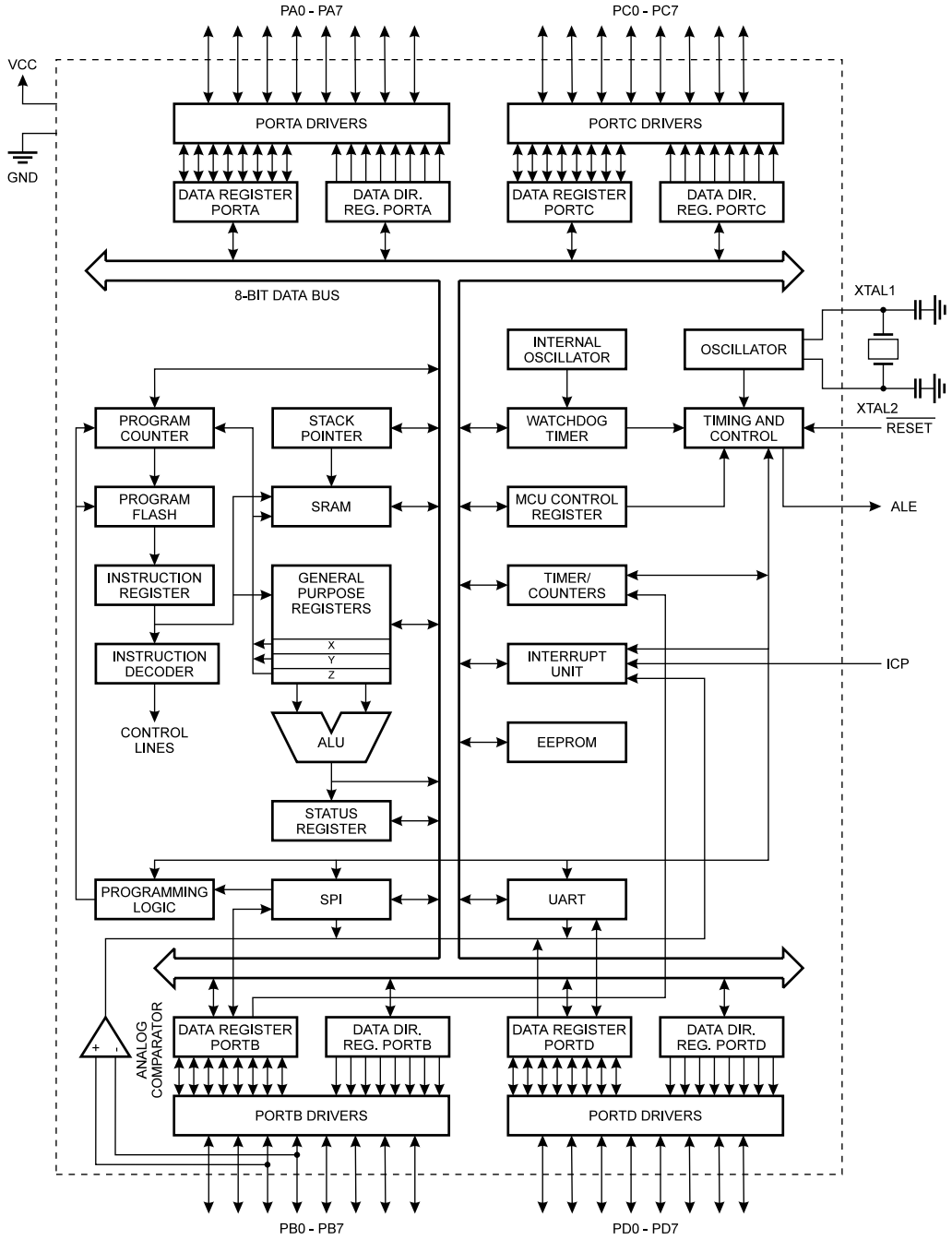


## Description

The AT90S8515 is a low-power CMOS 8-bit microcontroller based on the AVR RISC architecture. By executing powerful instructions in a single clock cycle, the AT90S8515 achieves throughputs approaching 1 MIPS per MHz, allowing the system designer to optimize power consumption versus processing speed.

## Block Diagram

**Figure 1.** The AT90S8515 Block Diagram




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.

The AT90S8515 provides the following features: 8K bytes of In-System Programmable Flash, 512 bytes EEPROM, 512 bytes SRAM, 32 general-purpose I/O lines, 32 general-purpose working registers, flexible timer/counters with compare modes, internal and external interrupts, a programmable serial UART, programmable Watchdog Timer with internal oscillator, an SPI serial port and two software-selectable power-saving modes. The Idle Mode stops the CPU while allowing the SRAM, timer/counters, SPI port 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 external interrupt or hardware reset.

The device is manufactured using Atmel's high-density nonvolatile memory technology. The On-chip In-System Programmable Flash allows the program memory to be reprogrammed In-System through an SPI serial interface or by a conventional nonvolatile memory programmer. By combining an enhanced RISC 8-bit CPU with In-System Programmable Flash on a monolithic chip, the Atmel AT90S8515 is a powerful microcontroller that provides a highly flexible and cost-effective solution to many embedded control applications.

The AT90S8515 AVR is supported with a full suite of program and system development tools including: C compilers, macro assemblers, program debugger/simulators, in-circuit emulators and evaluation kits.

## Pin Descriptions

**VCC** Supply voltage.

**GND** Ground.

**Port A (PA7..PA0)** Port A is an 8-bit bi-directional I/O port. Port pins can provide internal pull-up resistors (selected for each bit). The Port A output buffers can sink 20 mA and can drive LED displays directly. When pins PA0 to PA7 are used as inputs and are externally pulled low, they will source current if the internal pull-up resistors are activated. The Port A pins are tri-stated when a reset condition becomes active, even if the clock is not active.

Port A serves as multiplexed address/data input/output when using external SRAM.

**Port B (PB7..PB0)** Port B is an 8-bit bi-directional I/O port with internal pull-up resistors. The Port B output buffers can sink 20 mA. As inputs, Port B pins that are externally pulled low will source current if the pull-up resistors are activated. The Port B pins are tri-stated when a reset condition becomes active, even if the clock is not active.

Port B also serves the functions of various special features of the AT90S8515 as listed on page 66.

**Port C (PC7..PC0)** Port C is an 8-bit bi-directional I/O port with internal pull-up resistors. The Port C output buffers can sink 20 mA. As inputs, Port C pins that are externally pulled low will source current if the pull-up resistors are activated. The Port C pins are tri-stated when a reset condition becomes active, even if the clock is not active.

Port C also serves as address output when using external SRAM.

**Port D (PD7..PD0)** Port D is an 8-bit bi-directional I/O port with internal pull-up resistors. The Port D output buffers can sink 20 mA. As inputs, Port D pins that are externally pulled low will source

current if the pull-up resistors are activated. The Port D pins are tri-stated when a reset condition becomes active, even if the clock is not active.

Port D also serves the functions of various special features of the AT90S8515 as listed on page 73.

**$\overline{\text{RESET}}$** Reset input. A low level on this pin for more than 50 ns will generate a reset, even if the clock is not running. Shorter pulses are not guaranteed to generate a reset.

**XTAL1** Input to the inverting oscillator amplifier and input to the internal clock operating circuit.

**XTAL2** Output from the inverting oscillator amplifier.

**ICP** ICP is the input pin for the Timer/Counter1 Input Capture function.

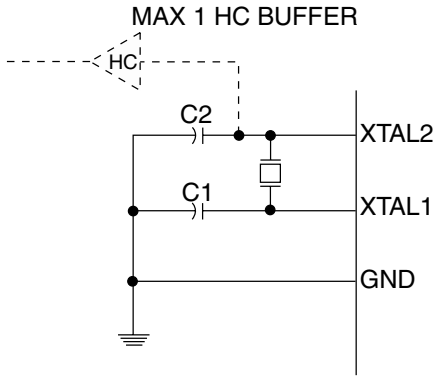
**OC1B** OC1B is the output pin for the Timer/Counter1 Output CompareB function.

**ALE** ALE is the Address Latch Enable used when the External Memory is enabled. The ALE strobe is used to latch the low-order address (8 bits) into an address latch during the first access cycle, and the AD0 - 7 pins are used for data during the second access cycle.

## Crystal Oscillator

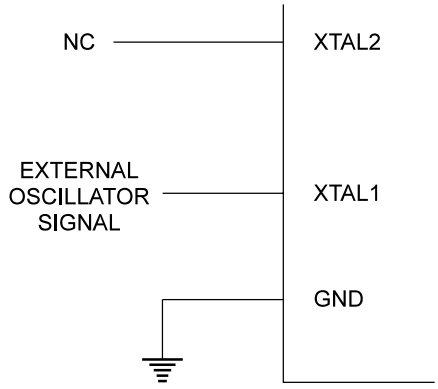
XTAL1 and XTAL2 are input and output, respectively, of an inverting amplifier that can be configured for use as an on-chip oscillator, as shown in Figure 2. Either a quartz crystal or a ceramic resonator may be used. To drive the device from an external clock source, XTAL2 should be left unconnected while XTAL1 is driven as shown in Figure 3.

**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.

**Figure 3.** External Clock Drive 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.

Six of the 32 registers can be used as three 16-bit indirect address register pointers for Data Space addressing, enabling efficient address calculations. One of the three address pointers is also used as the address pointer for the constant table look-up function. These added function registers are the 16-bit X-, Y-, and Z-register.

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 4 shows the AT90S8515 AVR RISC microcontroller architecture.

In addition to the register operation, the conventional memory addressing modes can be used on the register file as well. This is enabled by the fact that the register file is assigned the 32 lowermost Data Space addresses ($00 - $1F), allowing them to be accessed as though they were ordinary memory locations.

The I/O memory space contains 64 addresses for CPU peripheral functions such as Control Registers, Timer/Counters, A/D converters and other I/O functions. The I/O memory can be accessed directly or as the Data Space locations following those of the register file, $20 - $5F.

The AVR uses a Harvard architecture concept – with separate memories and buses for program and data. The program memory is executed 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 in every clock cycle. The program memory is In-System Programmable Flash memory.

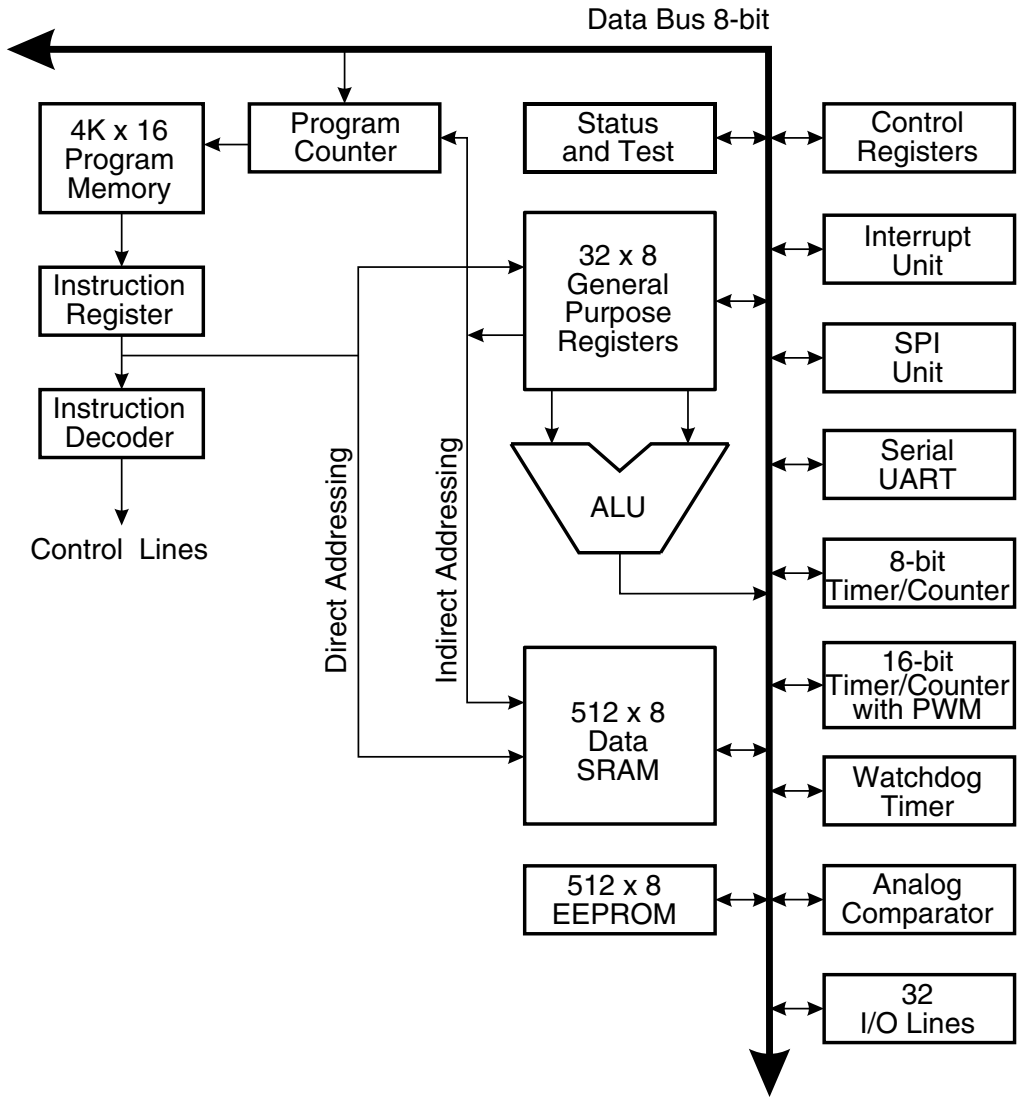
With the relative jump and call instructions, the whole 4K address space is directly accessed. Most AVR instructions have a single 16-bit word format. Every program memory address contains a 16- or 32-bit instruction.

During interrupts and subroutine calls, the return address Program Counter (PC) is stored on the stack. The stack is effectively allocated in the general data SRAM and consequently, the stack size is only limited by the total SRAM size and the usage of the SRAM. All user programs must initialize the SP in the reset routine (before subroutines or interrupts are executed). The 16-bit Stack Pointer (SP) is read/write-accessible in the I/O space.

The 512-byte data SRAM can be easily accessed through the five different addressing modes supported in the AVR architecture.

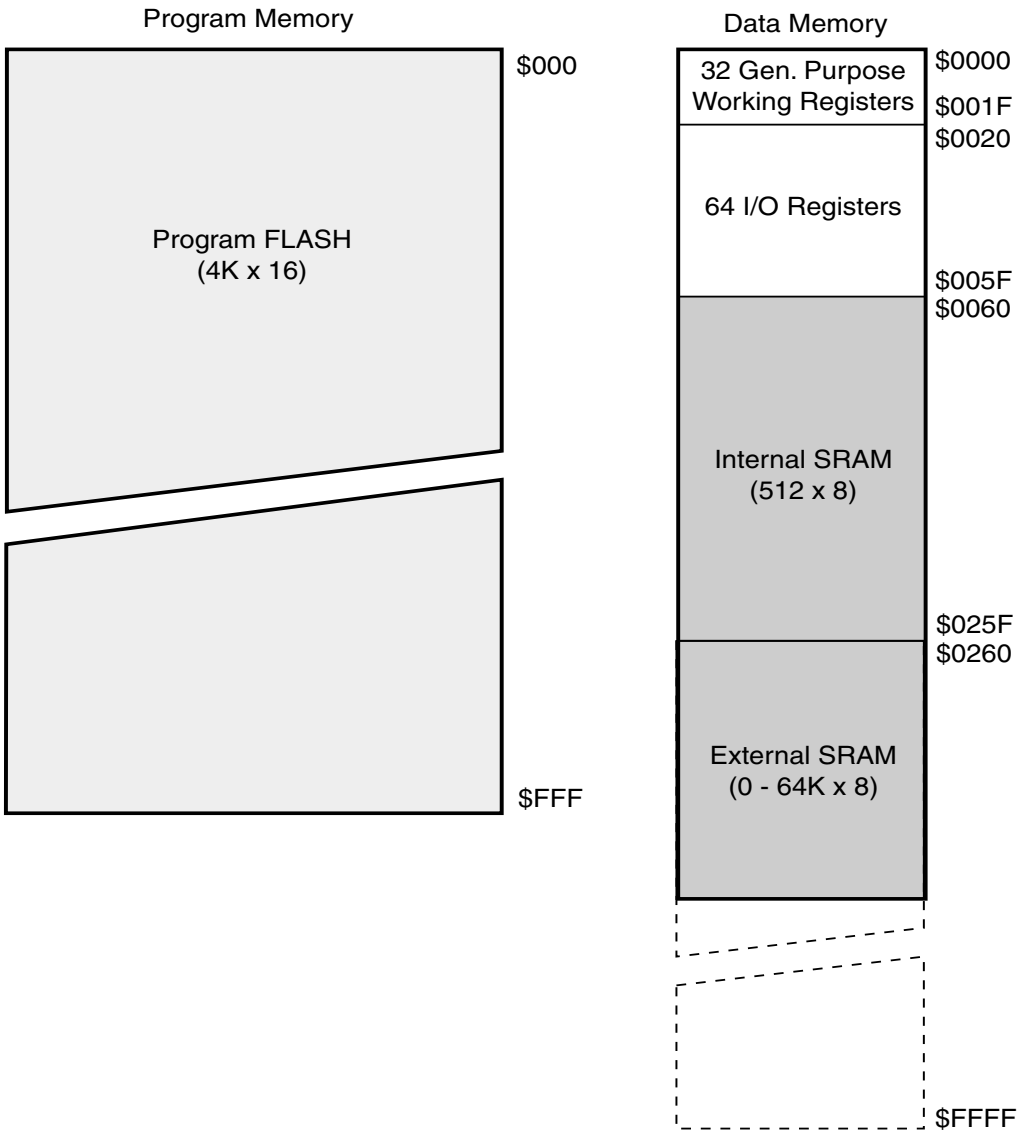
The memory spaces in the AVR architecture are all linear and regular memory maps.

**Figure 4.** The AT90S8515 AVR RISC Architecture




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.

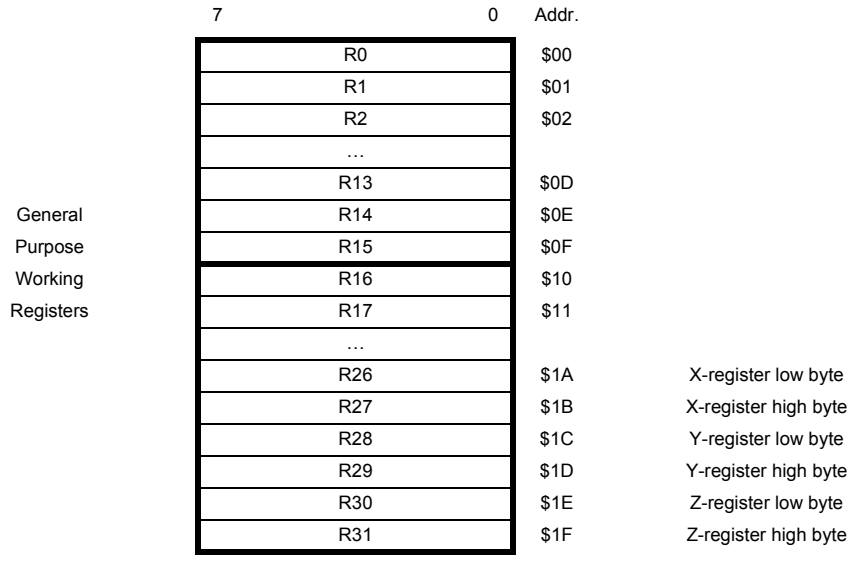
**Figure 5.** Memory Maps



## General-purpose Register File

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

**Figure 6.** AVR CPU General-purpose Working Registers



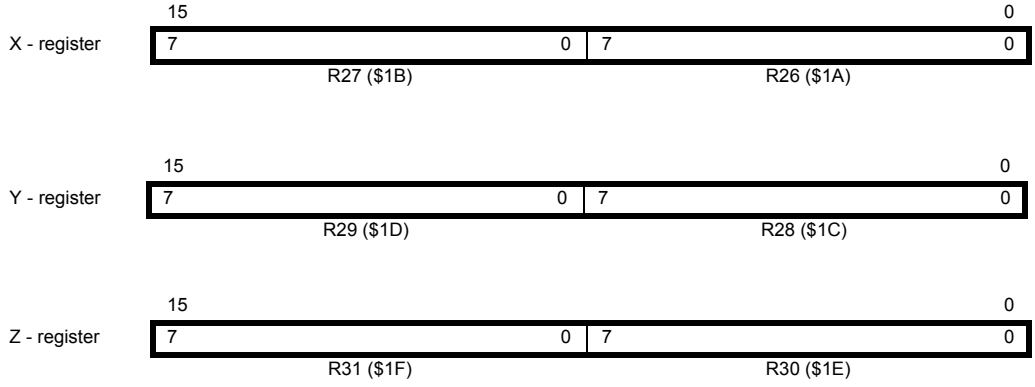

All the register operating instructions in the instruction set have direct and single-cycle access to all registers. The only exception are the five constant arithmetic and logic instructions SBCI, SUBI, CPI, ANDI and ORI between a constant and a register and the LDI instruction for load immediate constant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

As shown in Figure 6, each register is also assigned a data memory address, mapping them directly into the first 32 locations of the user Data Space. Although not being physically implemented as SRAM locations, this memory organization provides great flexibility in access of the registers, as the X-, Y- and Z-registers can be set to index any register in the file.

### X-register, Y-register and Z-register

The registers R26..R31 have some added functions to their general-purpose usage. These registers are address pointers for indirect addressing of the Data Space. The three indirect address registers X, Y, and Z are defined as:

**Figure 7.** X-, Y-, and Z-registers



In the different addressing modes these address registers have functions as fixed displacement, automatic increment and decrement (see the descriptions for the different instructions).

## 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, logical and bit functions.

## In-System Programmable Flash Program Memory

The AT90S8515 contains 8K bytes On-chip In-System Programmable Flash memory for program storage. Since all instructions are 16- or 32-bit words, the Flash is organized as 4K x 16. The Flash memory has an endurance of at least 1000 write/erase cycles. The AT90S8515 Program Counter (PC) is 12 bits wide, thus addressing the 4096 program memory addresses.

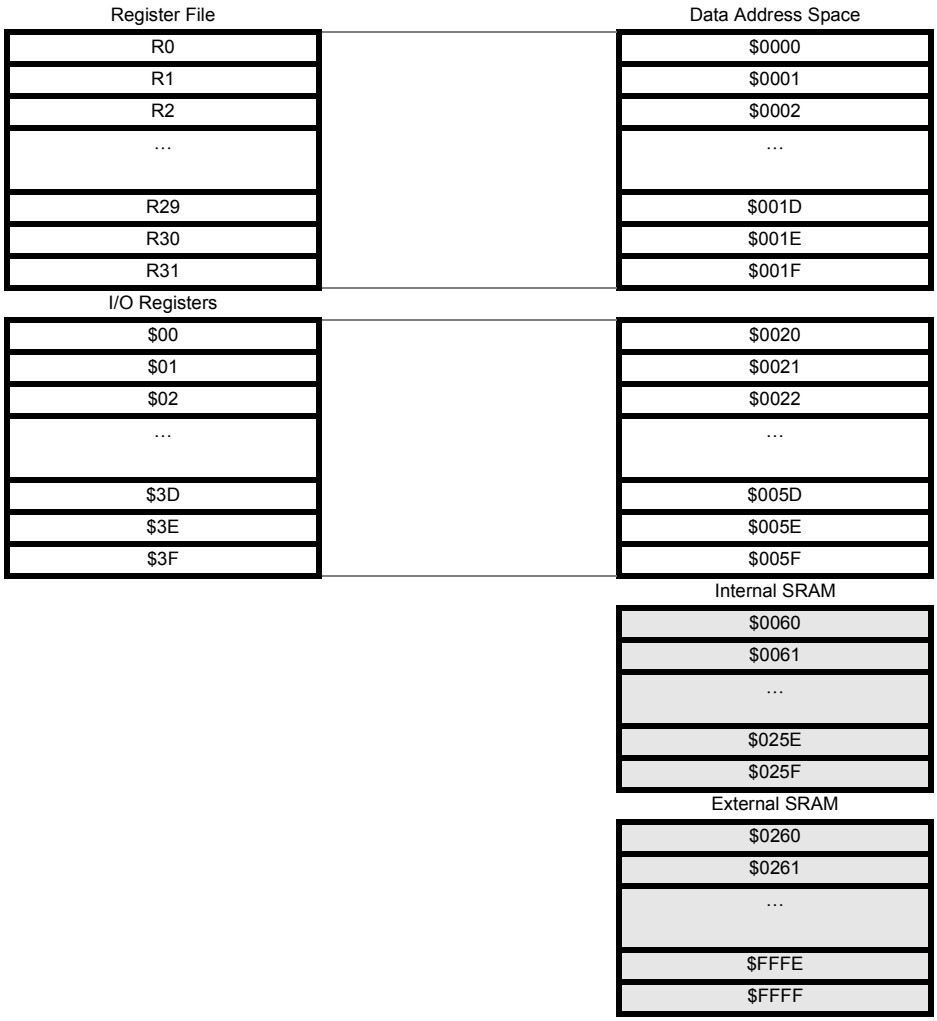
See page 86 for a detailed description of Flash data downloading.

See page 13 for the different program memory addressing modes.

## SRAM Data Memory – Internal and External

Figure 8 shows how the AT90S8515 SRAM memory is organized.

**Figure 8.** SRAM Organization




The lower 608 data memory locations address the Register file, the I/O memory and the internal data SRAM. The first 96 locations address the Register file + I/O memory, and the next 512 locations address the internal data SRAM. An optional external data SRAM can be placed in the same SRAM memory space. This SRAM will occupy the location following the internal SRAM and up to as much as 64K - 1, depending on SRAM size.

When the addresses accessing the data memory space exceed the internal data SRAM locations, the external data SRAM is accessed using the same instructions as for the internal data SRAM access. When the internal data space is accessed, the read and write strobe pins ($\overline{RD}$ and $\overline{WR}$) are inactive during the whole access cycle. External SRAM operation is enabled by setting the SRE bit in the MCUCR register. See page 29 for details.

Accessing external SRAM takes one additional clock cycle per byte compared to access of the internal SRAM. This means that the commands LD, ST, LDS, STS, PUSH and POP take one additional clock cycle. If the stack is placed in external SRAM, interrupts, subroutine calls and returns take two clock cycles extra because the 2-byte program counter is pushed and popped. When external SRAM interface is used with wait state,

two additional clock cycles is used per byte. This has the following effect: Data transfer instructions take two extra clock cycles, whereas interrupt, subroutine calls and returns will need four clock cycles more than specified in the instruction set manual.

The five different addressing modes for the data memory cover: Direct, Indirect with Displacement, Indirect, Indirect with Pre-decrement and Indirect with Post-increment. In the register file, registers R26 to R31 feature the indirect addressing pointer registers.

The direct addressing reaches the entire data space.

The Indirect with Displacement mode features 63 address locations reached from the base address given by the Y- or Z-register.

When using register indirect addressing modes with automatic pre-decrement and post-increment, the address registers X, Y and Z are decremented and incremented.

The 32 general-purpose working registers, 64 I/O registers, the 512 bytes of internal data SRAM, and the 64K bytes of optional external data SRAM in the AT90S8515 are all accessible through all these addressing modes.

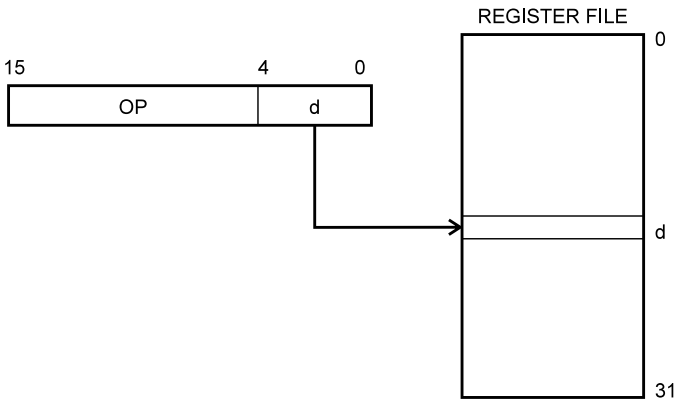
See the next section for a detailed description of the different addressing modes.

## Program and Data Addressing Modes

The AT90S8515 AVR RISC microcontroller supports powerful and efficient addressing modes for access to the program memory (Flash) and data memory (SRAM, Register file and I/O memory). This section describes the different addressing modes supported by the AVR architecture. 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

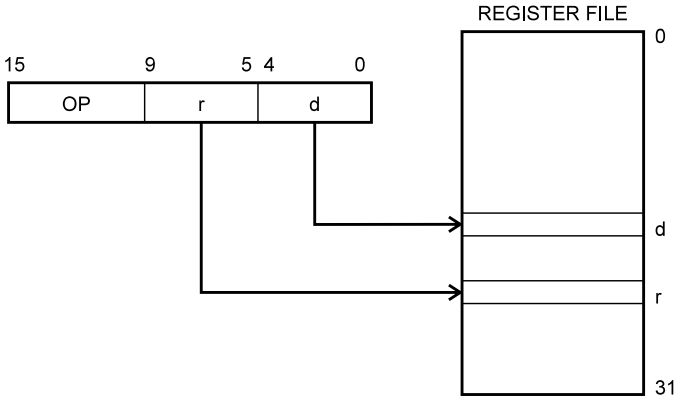
**Figure 9.** Direct Single Register Addressing




The operand is contained in register d (Rd).

### Register Direct, Two Registers Rd and Rr

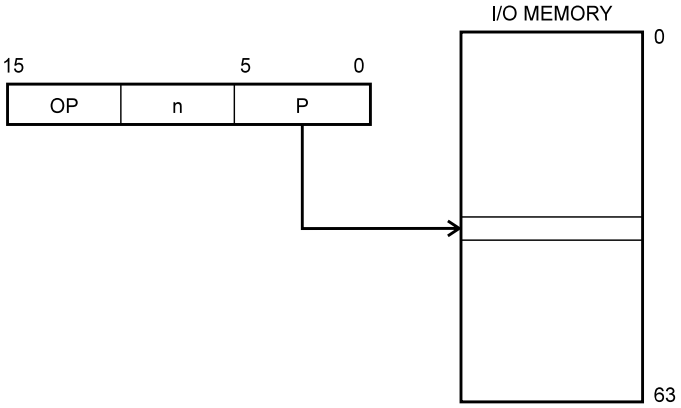
**Figure 10.** Direct Register Addressing, Two Registers




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

### I/O Direct

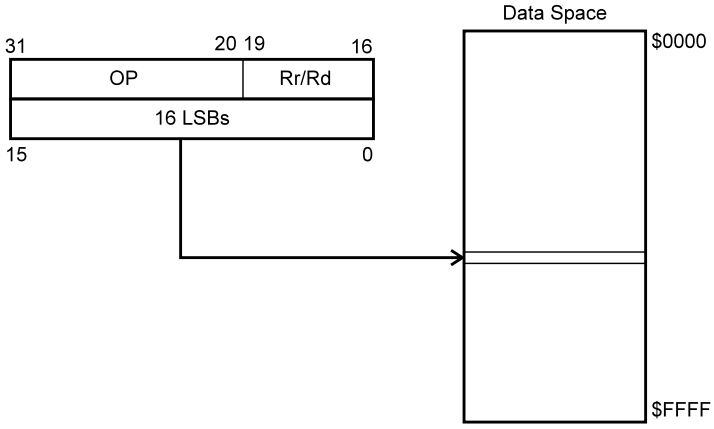
**Figure 11.** I/O Direct Addressing




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

### Data Direct

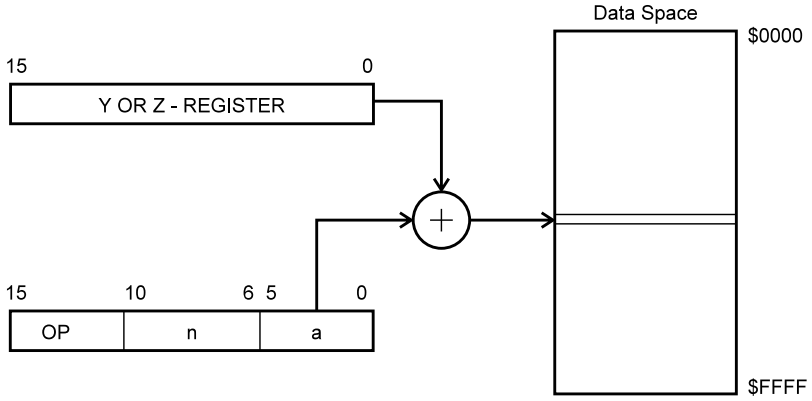
**Figure 12.** Direct Data Addressing



A 16-bit data address is contained in the 16 LSBs of a 2-word instruction. Rd/Rr specify the destination or source register.

### Data Indirect with Displacement

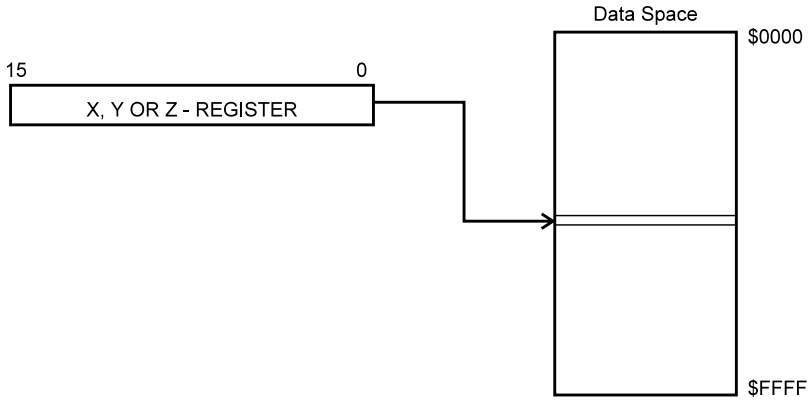
**Figure 13.** Data Indirect with Displacement




Operand address is the result of the Y- or Z-register contents added to the address contained in six bits of the instruction word.

### Data Indirect

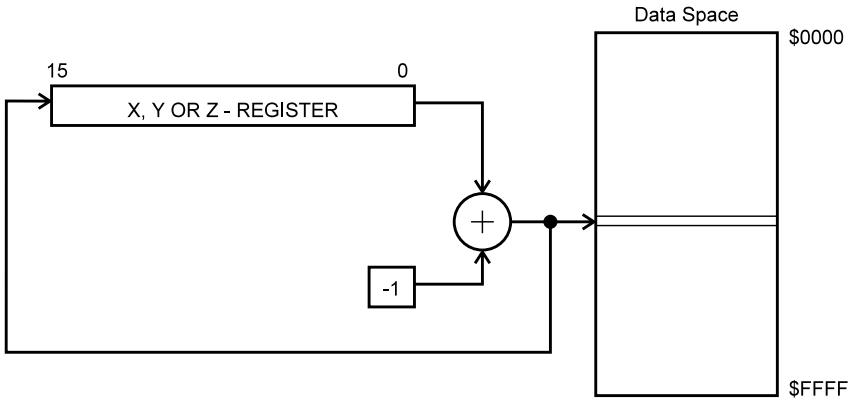
**Figure 14.** Data Indirect Addressing




Operand address is the contents of the X-, Y-, or the Z-register.

### Data Indirect with Pre-decrement

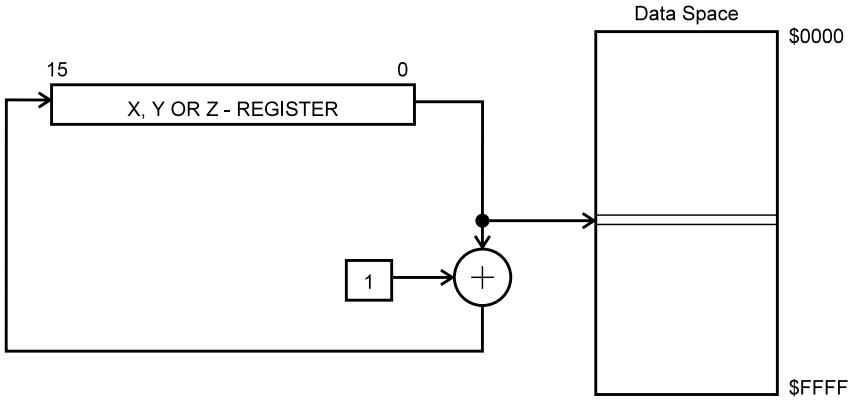
**Figure 15.** Data Indirect Addressing with Pre-decrement



The X-, Y-, or the Z-register is decremented before the operation. Operand address is the decremented contents of the X-, Y-, or the Z-register.

**Data Indirect with Post-increment**

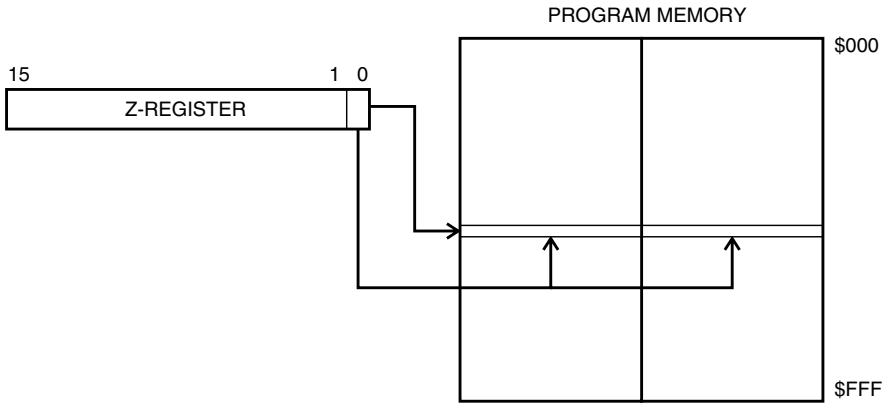
**Figure 16.** Data Indirect Addressing with Post-increment




The X-, Y-, or the Z-register is incremented after the operation. Operand address is the content of the X-, Y-, or the Z-register prior to incrementing.

**Constant Addressing Using the LPM Instruction**

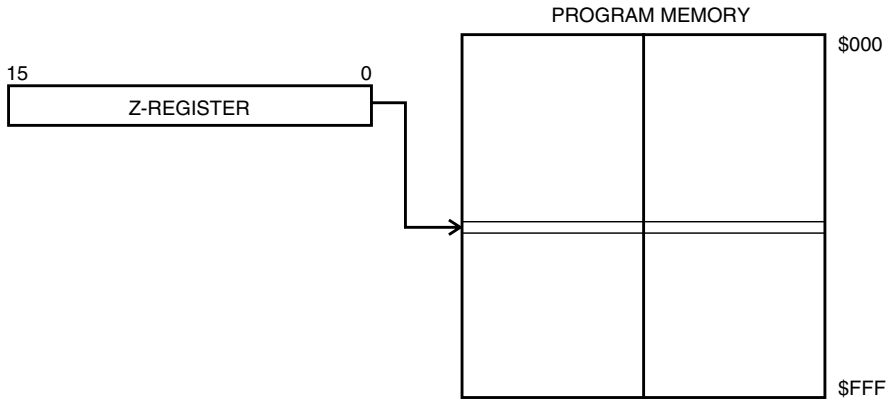
**Figure 17.** Code Memory Constant Addressing




Constant byte address is specified by the Z-register contents. The 15 MSBs select word address (0 - 4K), the LSB selects low byte if cleared (LSB = 0) or high byte if set (LSB = 1).

### Indirect Program Addressing, IJMP and ICALL

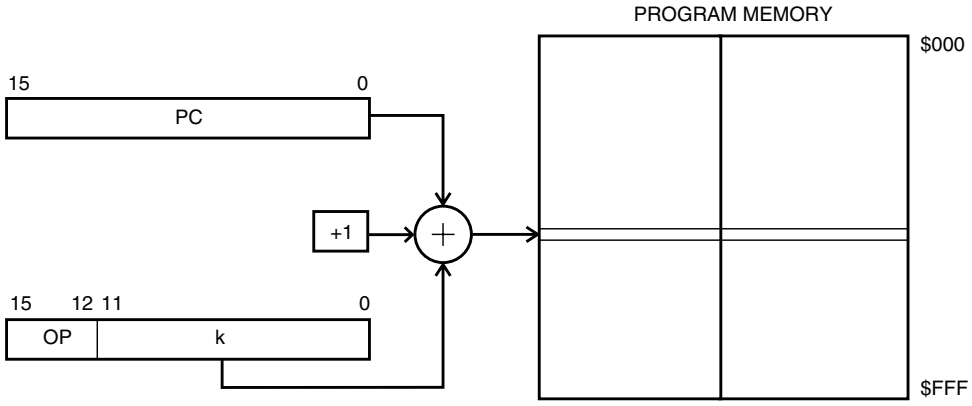
**Figure 18.** Indirect Program Memory Addressing




Program execution continues at address contained by the Z-register (i.e., the PC is loaded with the contents of the Z-register).

### Relative Program Addressing, RJMP and RCALL

**Figure 19.** Relative Program Memory Addressing




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

## EEPROM Data Memory

The AT90S8515 contains 512 bytes of data EEPROM memory. It is organized as a separate data space, in which single bytes can be read and written. The EEPROM has an endurance of at least 100,000 write/erase cycles. The access between the EEPROM and the CPU is described on page 44, specifying the EEPROM address registers, the EEPROM data register and the EEPROM control register.

For the SPI data downloading, see page 86 for a detailed description.

## Memory Access Times 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 Ø, directly generated from the external clock crystal for the chip. No internal clock division is used.

Figure 20 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 20.** The Parallel Instruction Fetches and Instruction Executions

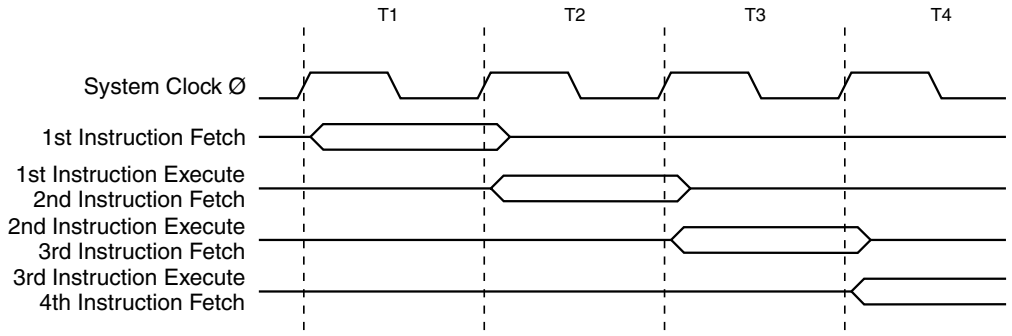
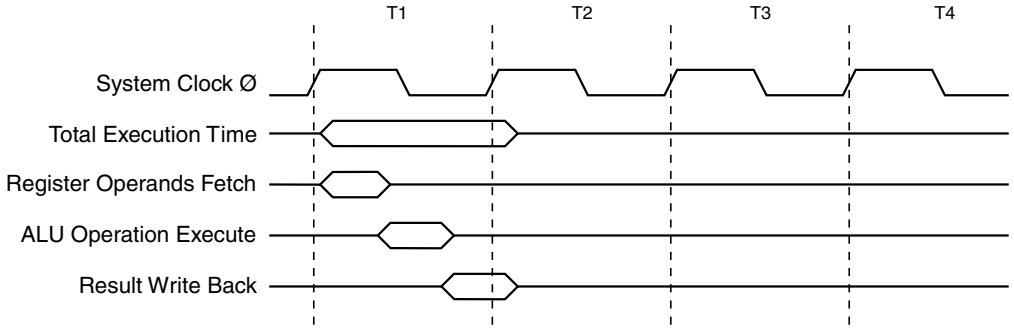


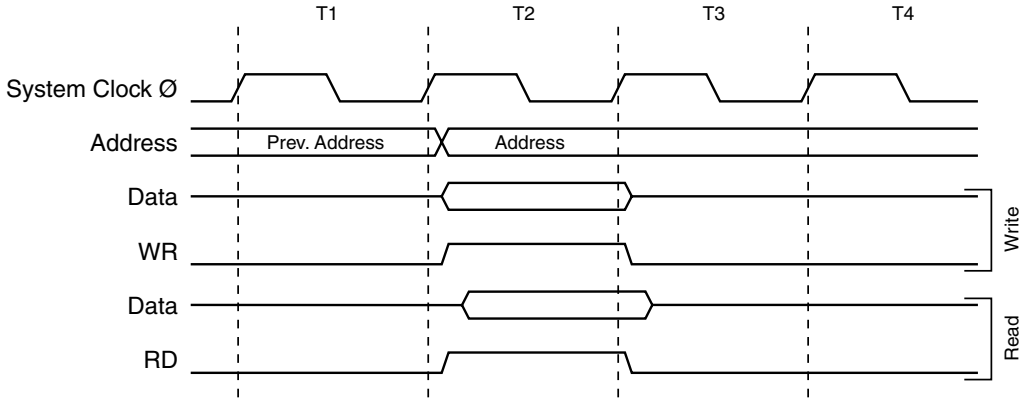

Figure 21 shows the internal timing concept for the register file. In a single clock cycle an ALU operation using two register operands is executed and the result is stored back to the destination register.

**Figure 21.** Single Cycle ALU Operation




The internal data SRAM access is performed in two System Clock cycles as described in Figure 22.

**Figure 22.** On-chip Data SRAM Access Cycles




See “Interface to External SRAM” on page 60 for a description of the access to the external SRAM.

## I/O Memory

The I/O space definition of the AT90S8515 is shown in Table 1.

**Table 1.** AT90S8515 I/O Space

| Address Hex | Name | Function |
|---|---|---|
| $3F ($5F) | SREG | Status Register |
| $3E ($5E) | SPH | Stack Pointer High |
| $3D ($5D) | SPL | Stack Pointer Low |
| $3B ($5B) | GIMSK | General Interrupt Mask register |
| $3A ($5A) | GIFR | General Interrupt Flag Register |
| $39 ($59) | TIMSK | Timer/Counter Interrupt Mask register |
| $38 ($58) | TIFR | Timer/Counter Interrupt Flag register |
| $35 ($55) | MCUCR | MCU general Control Register |
| $33 ($53) | TCCR0 | Timer/Counter0 Control Register |
| $32 ($52) | TCNT0 | Timer/Counter0 (8-bit) |
| $2F ($4F) | TCCR1A | Timer/Counter1 Control Register A |
| $2E ($4E) | TCCR1B | Timer/Counter1 Control Register B |
| $2D ($4D) | TCNT1H | Timer/Counter1 High Byte |
| $2C ($4C) | TCNT1L | Timer/Counter1 Low Byte |
| $2B ($4B) | OCR1AH | Timer/Counter1 Output Compare Register A High Byte |
| $2A ($4A) | OCR1AL | Timer/Counter1 Output Compare Register A Low Byte |
| $29 ($49) | OCR1BH | Timer/Counter1 Output Compare Register B High Byte |
| $28 ($48) | OCR1BL | Timer/Counter1 Output Compare Register B Low Byte |
| $25 ($45) | ICR1H | T/C 1 Input Capture Register High Byte |
| $24 ($44) | ICR1L | T/C 1 Input Capture Register Low Byte |
| $21 ($41) | WDTCR | Watchdog Timer Control Register |
| $1F ($3E) | EEARH | EEPROM Address Register High Byte (AT90S8515) |
| $1E ($3E) | EEARL | EEPROM Address Register Low Byte |
| $1D ($3D) | EEDR | EEPROM Data Register |
| $1C ($3C) | EECR | EEPROM Control Register |
| $1B ($3B) | PORTA | Data Register, Port A |
| $1A ($3A) | DDRA | Data Direction Register, Port A |
| $19 ($39) | PINA | Input Pins, Port A |
| $18 ($38) | PORTB | Data Register, Port B |
| $17 ($37) | DDRB | Data Direction Register, Port B |
| $16 ($36) | PINB | Input Pins, Port B |
| $15 ($35) | PORTC | Data Register, Port C |
| $14 ($34) | DDRC | Data Direction Register, Port C |
| $13 ($33) | PINC | Input Pins, Port C |
| $12 ($32) | PORTD | Data Register, Port D |

**Table 1.** AT90S8515 I/O Space (Continued)

| Address Hex | Name | Function |
|---|---|---|
| $11 ($31) | DDRD | Data Direction Register, Port D |
| $10 ($30) | PIND | Input Pins, Port D |
| $0F ($2F) | SPDR | SPI I/O Data Register |
| $0E ($2E) | SPSR | SPI Status Register |
| $0D ($2D) | SPCR | SPI Control Register |
| $0C ($2C) | UDR | UART I/O Data Register |
| $0B ($2B) | USR | UART Status Register |
| $0A ($2A) | UCR | UART Control Register |
| $09 ($29) | UBRR | UART Baud Rate Register |
| $08 ($28) | ACSR | Analog Comparator Control and Status Register |

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

All AT90S8515 I/Os 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 section for more details. When using the I/O-specific commands IN and OUT, the I/O addresses $00 - $3F must be used. When addressing I/O registers as SRAM, $20 must be added to this address. All I/O register addresses throughout this document are shown with the SRAM address in parentheses.

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

Some of the status flags are cleared by writing a logical “1” to them. Note that the CBI and SBI instructions will operate on all bits in the I/O register, writing a “1” back into any flag read as set, thus clearing the flag. The CBI and SBI instructions work with registers $00 to $1F only.

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 ($5F) is defined as:

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---|---|---|---|---|---|---|---|---|---|
| $3F ($5F) | 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 bit 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 ⊕ V**

The S-bit is always an exclusive or between the negative flag N and the two's complement overflow flag V. See the Instruction Set description for detailed information.

- **Bit 3 – V: Two's Complement Overflow Flag**

The two's complement overflow flag V supports two's complement arithmetics. See the Instruction Set description for detailed information.

- **Bit 2 – N: Negative Flag**

The negative flag N indicates a negative result after the different arithmetic and logic operations. See the Instruction Set description for detailed information.

- **Bit 1 – Z: Zero Flag**

The zero flag Z indicates a zero result after the different arithmetic and logic operations. See the Instruction Set description for detailed information.

- **Bit 0 – C: Carry Flag**

The carry flag C indicates a carry in an arithmetic or logic operation. See the Instruction Set description for detailed information.

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.

**Stack Pointer – SP**

The general AVR 16-bit Stack Pointer is effectively built up of two 8-bit registers in the I/O space locations $3E ($5E) and $3D ($5D). As the AT90S8515 supports up to 64 Kb external SRAM, all 16 bits are used.

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | |
|---|---|---|---|---|---|---|---|---|---|
| $3E ($5E) | **SP15** | **SP14** | **SP13** | **SP12** | **SP11** | **SP10** | **SP9** | **SP8** | **SPH** |
| $3D ($5D) | **SP7** | **SP6** | **SP5** | **SP4** | **SP3** | **SP2** | **SP1** | **SP0** | **SPL** |
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |

The Stack Pointer points to the data SRAM stack area where the Subroutine and Interrupt stacks are located. This stack space in the data SRAM must be defined by the program before any subroutine calls are executed or interrupts are enabled. The Stack Pointer must be set to point above $60. The Stack Pointer is decremented by 1 when data is pushed onto the stack with the PUSH instruction and it is decremented by 2 when an address is pushed onto the stack with subroutine calls and interrupts. The Stack Pointer is incremented by 1 when data is popped from the stack with the POP instruction and it is incremented by 2 when an address is popped from the stack with return from subroutine RET or return from interrupt RETI.

## Reset and Interrupt Handling

The AT90S8515 provides 12 different interrupt sources. These interrupts and the separate reset vector each have a separate program vector in the program memory space. All interrupts are assigned individual enable bits that must be set (one) together with the I-bit in the Status Register in order to enable the interrupt.

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 2. The list also determines the priority levels of the different interrupts. The lower the address, the higher the priority level. RESET has the highest priority, and next is INT0 (the External Interrupt Request 0), etc.

**Table 2.** Reset and Interrupt Vectors

| Vector No. | Program Address | Source | Interrupt Definition |
|---|---|---|---|
| 1 | $000 | RESET | External Reset, Power-on Reset and Watchdog Reset |
| 2 | $001 | INT0 | External Interrupt Request 0 |
| 3 | $002 | INT1 | External Interrupt Request 1 |
| 4 | $003 | TIMER1 CAPT | Timer/Counter1 Capture Event |
| 5 | $004 | TIMER1 COMPA | Timer/Counter1 Compare Match A |
| 6 | $005 | TIMER1 COMPB | Timer/Counter1 Compare Match B |
| 7 | $006 | TIMER1 OVF | Timer/Counter1 Overflow |
| 8 | $007 | TIMER0, OVF | Timer/Counter0 Overflow |
| 9 | $008 | SPI, STC | Serial Transfer Complete |
| 10 | $009 | UART, RX | UART, Rx Complete |
| 11 | $00A | UART, UDRE | UART Data Register Empty |
| 12 | $00B | UART, TX | UART, Tx Complete |
| 13 | $00C | 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   TIM1_CAPT    ; Timer1 Capture Handler
$004                   rjmp   TIM1_COMPA   ; Timer1 CompareA Handler
$005                   rjmp   TIM1_COMPB   ; Timer1 CompareB Handler
$006                   rjmp   TIM1_OVF       ; Timer1 Overflow Handler
$007                   rjmp   TIM0_OVF     ; Timer0 Overflow Handler
$008                   rjmp   SPI_STC      ; SPI Transfer Complete Handler
$009                   rjmp   UART_RXC     ; UART RX Complete Handler
$00a                   rjmp   UART_DRE     ; UDR Empty Handler
$00b                   rjmp   UART_TXC     ; UART TX Complete Handler
$00c                   rjmp   ANA_COMP     ; Analog Comparator Handler
;
$00d      MAIN:        ldi r16,high(RAMEND); Main program start
$00e                   out SPH,r16
```

```
$00f                ldi r16,low(RAMEND)
$010                out SPL,r16
$011                <instr>  xxx

   …         …           …       …
```

## Reset Sources

The AT90S8515 has 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 $\overline{\text{RESET}}$ 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 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 23 shows the reset logic. Table 3 defines the timing and electrical parameters of the reset circuitry.

**Figure 23.** Reset Logic

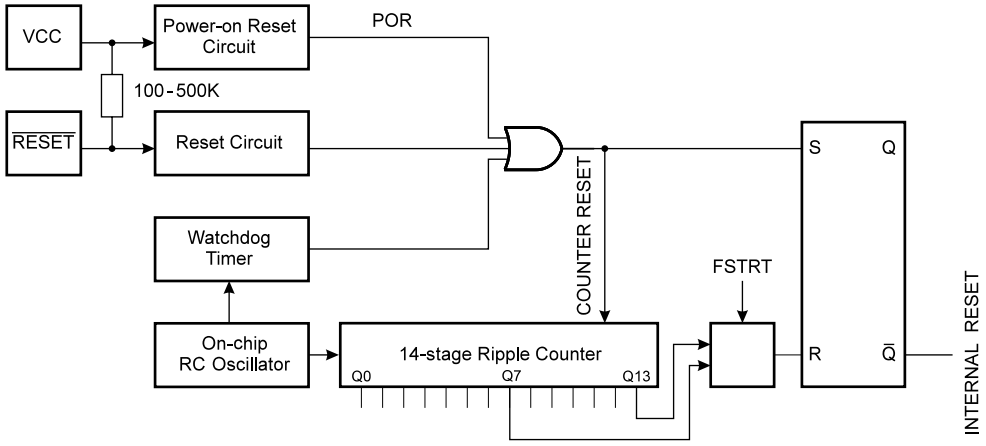



**Table 3.** Reset Characteristics

| Symbol | Parameter | Min | Typ | Max | Units |
|---|---|---|---|---|---|
| $V_{POT}$ (Note:) | Power-on Reset Threshold Voltage (rising) | 0.8 | 1.2 | 1.6 | V |
| | Power-on Reset Threshold Voltage (falling) | 0.2 | 0.4 | 0.6 | V |
| $V_{RST}$ | $\overline{\text{RESET}}$ Pin Threshold Voltage | – | – | 0.9 $V_{CC}$ | V |
| $t_{TOUT}$ | Reset Delay Time-out Period FSTRT Unprogrammed | 11.0 | 16.0 | 21.0 | ms |
| $t_{TOUT}$ | Reset Delay Time-out Period FSTRT Programmed | 0.25 | 0.28 | 0.31 | ms |

Note: The Power-on Reset will not work unless the supply voltage has been below $V_{POT}$ (falling).

The user can select the start-up time according to typical oscillator start-up. The number of WDT oscillator cycles used for each time-out is shown in Table 4. The frequency of the Watchdog Oscillator is voltage-dependent as shown in "Typical Characteristics" on page 95.

**Table 4.** Number of Watchdog Oscillator Cycles

| FSTRT | Time-out at $V_{CC}$ = 5V | Number of WDT Cycles |
|---|---|---|
| Programmed | 0.28 ms | 256 |
| Unprogrammed | 16.0 ms | 16K |

## Power-on Reset

A Power-on Reset (POR) circuit ensures that the device is reset from power-on. As shown in Figure 23, an internal timer clocked from the Watchdog Timer oscillator prevents the MCU from starting until after a certain period after $V_{CC}$ has reached the Power-on Threshold Voltage ($V_{POT}$), regardless of the $V_{CC}$ rise time (see Figure 24). The FSTRT Fuse bit in the Flash can be programmed to give a shorter start-up time if a ceramic resonator or any other fast-start oscillator is used to clock the MCU.

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 pin low for a period after $V_{CC}$ has been applied, the Power-on Reset period can be extended. Refer to Figure 25 for a timing example of this.

**Figure 24.** MCU Start-up, $\overline{\text{RESET}}$ Tied to $V_{CC}$.

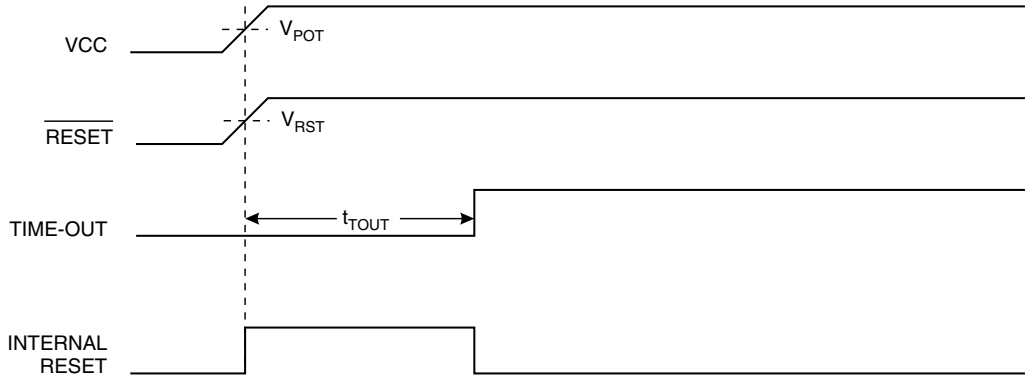
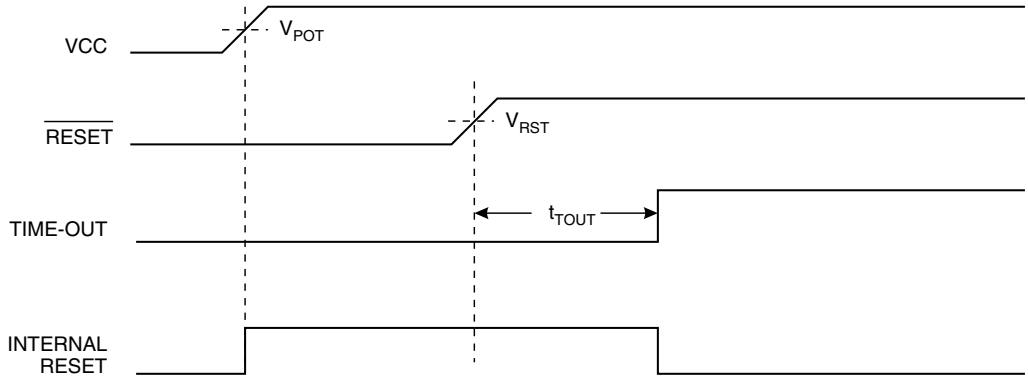


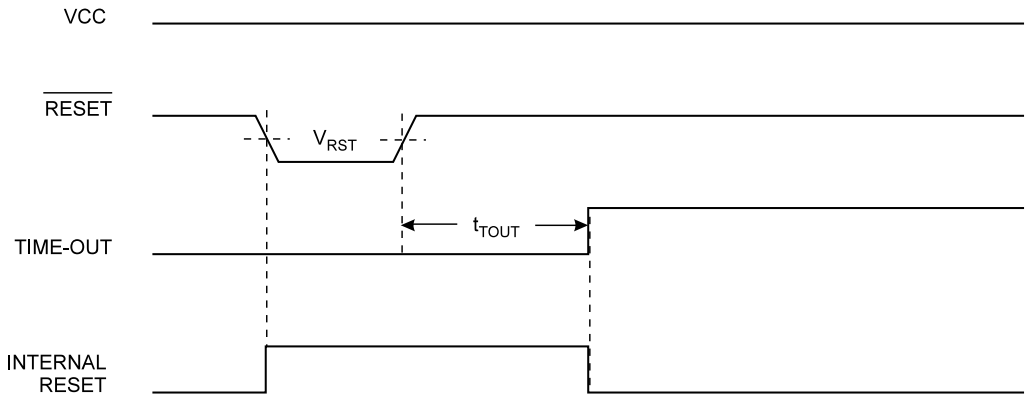

**Figure 25.** MCU Start-up, $\overline{\text{RESET}}$ Controlled Externally



### External Reset

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

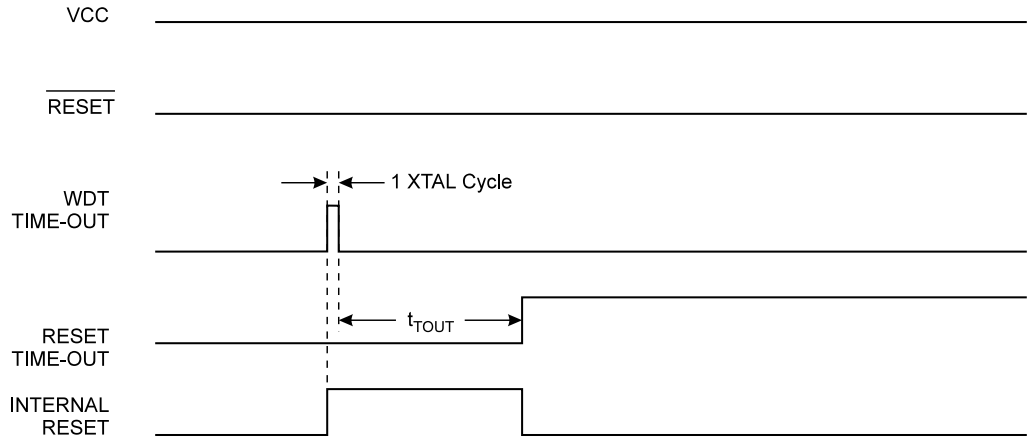
**Figure 26.** External Reset during Operation




### 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 $t_{TOUT}$. Refer to page 42 for details on operation of the Watchdog.

**Figure 27.** Watchdog Reset during Operation




### Interrupt Handling

The AT90S8515 has two 8-bit interrupt mask control registers; GIMSK (General Interrupt Mask register) and TIMSK (Timer/Counter Interrupt Mask register).

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.

For interrupts triggered by events that can remain static (e.g., the Output Compare Register1 matching the value of Timer/Counter1), the interrupt flag is set when the event occurs. If the interrupt flag is cleared and the interrupt condition persists, the flag will not be set until the event occurs the next time.

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 logical “1” 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.

### General Interrupt Mask Register – GIMSK

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---|---|---|---|---|---|---|---|---|---|
| $3B ($5B) | INT1 | INT0 | – | – | – | – | – | – | GIMSK |
| Read/Write | R/W | R/W | R | R | R | R | R | R | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |

- **Bit 7 – INT1: External Interrupt Request 1 Enable**

When the INT1 bit is set (one) and the I-bit in the Status Register (SREG) is set (one), the external pin interrupt is enabled. The Interrupt Sense Control1 bits 1/0 (ISC11 and ISC10) in the MCU general Control Register (MCUCR) define whether the external interrupt is activated on rising or falling edge of the INT1 pin or is level-sensed. Activity on the pin will cause an interrupt request even if INT1 is configured as an output. The corresponding interrupt of External Interrupt Request 1 is executed from 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 is enabled. The Interrupt Sense Control0 bits 1/0 (ISC01 and ISC00) in the MCU general Control Register (MCUCR) define whether the external interrupt is activated on rising or falling edge of the INT0 pin or is level-sensed. Activity on the pin will cause an interrupt request even if INT0 is configured as an output. The corresponding interrupt of External Interrupt Request 0 is executed from program memory address $001. See also “External Interrupts”.

- **Bits 5..0 – Res: Reserved Bits**

These bits are reserved bits in the AT90S8515 and always read as zero.

### General Interrupt Flag Register – GIFR

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---|---|---|---|---|---|---|---|---|---|
| $3A ($5A) | INTF1 | INTF0 | – | – | – | – | – | – | GIFR |
| Read/Write | R/W | R/W | R | R | R | R | R | R | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |

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

When an edge on the INT1 pin triggers an interrupt request, the corresponding interrupt flag, INTF1 becomes set (one). If the I-bit in SREG and the corresponding interrupt enable bit, INT1 in GIMSK is set (one), the MCU will jump to the interrupt vector. The flag is cleared when the interrupt routine is executed. Alternatively, the flag can be cleared by writing a logical “1” to it. This flag is always cleared when INT1 is configured as level interrupt.

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

When an edge on the INT0 pin triggers an interrupt request, the corresponding interrupt flag, INTF0, becomes set (one). If the I-bit in SREG and the corresponding interrupt enable bit, INT0 in GIMSK are set (one), the MCU will jump to the interrupt vector. The flag is cleared when the interrupt routine is executed. Alternatively, the flag is cleared by writing a logical "1" to it. This flag is always cleared when INT0 is configured as level interrupt.

- **Bits 5..0 – Res: Reserved Bits**

These bits are reserved bits in the AT90S8515 and always read as zero.

### Timer/Counter Interrupt Mask Register – TIMSK

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---|---|---|---|---|---|---|---|---|---|
| $39 ($59) | TOIE1 | OCIE1A | OCIE1B | – | TICIE1 | – | TOIE0 | – | TIMSK |
| Read/Write | R/W | R/W | R/W | R | R/W | R | R/W | R | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |

- **Bit 7 – TOIE1: Timer/Counter1 Overflow Interrupt Enable**

When the TOIE1 bit is set (one) and the I-bit in the Status Register is set (one), the Timer/Counter1 Overflow interrupt is enabled. The corresponding interrupt (at vector $006) is executed if an overflow in Timer/Counter1 occurs, i.e., when the TOV1 bit is set in the Timer/Counter Interrupt Flag Register (TIFR).

- **Bit 6 – OCE1A: Timer/Counter1 Output CompareA Match Interrupt Enable**

When the OCIE1A bit is set (one) and the I-bit in the Status Register is set (one), the Timer/Counter1 CompareA Match interrupt is enabled. The corresponding interrupt (at vector $004) is executed if a CompareA match in Timer/Counter1 occurs, i.e., when the OCF1A bit is set in the Timer/Counter Interrupt Flag Register (TIFR).

- **Bit 5 – OCIE1B: Timer/Counter1 Output CompareB Match Interrupt Enable**

When the OCIE1B bit is set (one) and the I-bit in the Status Register is set (one), the Timer/Counter1 CompareB Match interrupt is enabled. The corresponding interrupt (at vector $005) is executed if a CompareB match in Timer/Counter1 occurs, i.e., when the OCF1B bit is set in the Timer/Counter Interrupt Flag Register (TIFR).

- **Bit 4 – Res: Reserved Bit**

This bit is a reserved bit in the AT90S8515 and always reads zero.

- **Bit 3 – TICIE1: Timer/Counter1 Input Capture Interrupt Enable**

When the TICIE1 bit is set (one) and the I-bit in the Status Register is set (one), the Timer/Counter1 Input Capture Event interrupt is enabled. The corresponding interrupt (at vector $003) is executed if a capture-triggering event occurs on pin 31, ICP, i.e., when the ICF1 bit is set in the Timer/Counter Interrupt Flag Register (TIFR).

- **Bit 2 – Res: Reserved Bit**

This bit is a reserved bit in the AT90S8515 and always reads zero.

- **Bit 1 – 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 $007) is executed if an overflow in Timer/Counter0 occurs, i.e., when the TOV0 bit is set in the Timer/Counter Interrupt Flag Register (TIFR).

- **Bit 0 – Res: Reserved Bit**

This bit is a reserved bit in the AT90S8515 and always reads zero.

**Timer/Counter Interrupt Flag Register – TIFR**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---|---|---|---|---|---|---|---|---|---|
| $38 ($58) | TOV1 | OCF1A | OCIFB | – | ICF1 | – | TOV0 | – | TIFR |
| Read/Write | R/W | R/W | R/W | R | R/W | R | R/W | R | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |

- **Bit 7 – TOV1: Timer/Counter1 Overflow Flag**

The TOV1 is set (one) when an overflow occurs in Timer/Counter1. TOV1 is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, TOV1 is cleared by writing a logical “1” to the flag. When the I-bit in SREG, TOIE1 (Timer/Counter1 Overflow Interrupt Enable) and TOV1 are set (one), the Timer/Counter1 Overflow interrupt is executed. In PWM mode, this bit is set when Timer/Counter1 changes counting direction at $0000.

- **Bit 6 – OCF1A: Output Compare Flag 1A**

The OCF1A bit is set (one) when compare match occurs between the Timer/Counter1 and the data in OCR1A (Output Compare Register 1A). OCF1A is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, OCF1A is cleared by writing a logical “1” to the flag. When the I-bit in SREG, OCIE1A (Timer/Counter1 Compare Match InterruptA Enable) and the OCF1A are set (one), the Timer/Counter1 CompareA Match interrupt is executed.

- **Bit 5 – OCF1B: Output Compare Flag 1B**

The OCF1B bit is set (one) when compare match occurs between the Timer/Counter1 and the data in OCR1B (Output Compare Register 1B). OCF1B is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, OCF1B is cleared by writing a logical “1” to the flag. When the I-bit in SREG, OCIE1B (Timer/Counter1 Compare Match InterruptB Enable) and the OCF1B are set (one), the Timer/Counter1 CompareB Match interrupt is executed.

- **Bit 4 – Res: Reserved Bit**

This bit is a reserved bit in the AT90S8515 and always reads zero.

- **Bit 3 – ICF1: Input Capture Flag 1**

The ICF1 bit is set (one) to flag an input capture event, indicating that the Timer/Counter1 value has been transferred to the input capture register (ICR1). ICF1 is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, ICF1 is cleared by writing a logical “1” to the flag. When the SREG I-bit, TICIE1 (Timer/Counter1 Input Capture Interrupt Enable) and ICF1 are set (one), the Timer/Counter1 Capture interrupt is executed.

- **Bit 2 – Res: Reserved Bit**

This bit is a reserved bit in the AT90S8515 and always reads zero.

- **Bit 1 – TOV: 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. Alternatively, TOV0 is cleared by writing a logical “1” to the flag. When the SREG I-bit, TOIE0 (Timer/Counter0 Overflow Interrupt Enable) and TOV0 are set (one), the Timer/Counter0 Overflow interrupt is executed.

- **Bit 0 – Res: Reserved Bit**

This bit is a reserved bit in the AT90S8515 and always reads zero.

### External Interrupts

The external interrupts are triggered by the INT1 and INT0 pins. Observe that, if enabled, the interrupts will trigger even if the INT0/INT1 pins are configured as outputs. This feature provides a way of generating a software interrupt. The external interrupts can be triggered by a falling or rising edge or a low level. This is set up as indicated in the specification for the MCU Control Register (MCUCR). 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 interrupts are set up as described in the specification for the MCU Control Register (MCUCR).

### Interrupt Response Time

The interrupt execution response for all the enabled AVR interrupts is four clock cycles minimum. Four clock cycles after the interrupt flag has been set, the program vector address for the actual interrupt handling routine is executed. During this 4-clock-cycle period, the Program Counter (2 bytes) is pushed onto the stack and the Stack Pointer is decremented by 2. The vector is normally a relative jump to the interrupt routine, and this jump takes two clock cycles. If an interrupt occurs during execution of a multi-cycle instruction, this instruction is completed before the interrupt is served.

A return from an interrupt handling routine (same as for a subroutine call routine) takes four clock cycles. During these four clock cycles, the Program Counter (2 bytes) is popped back from the stack, the Stack Pointer is incremented by 2 and the I-flag in SREG is set. When the AVR exits from an interrupt, it will always return to the main program and execute one more instruction before any pending interrupt is served.

Note that the Status Register (SREG) is not handled by the AVR hardware, for neither interrupts nor subroutines. For the interrupt handling routines requiring a storage of the SREG, this must be performed by user software.

For interrupts triggered by events that can remain static (e.g., the Output Compare Register1 A matching the value of Timer/Counter1), the interrupt flag is set when the event occurs. If the interrupt flag is cleared and the interrupt condition persists, the flag will not be set until the event occurs the next time. Note that an external level interrupt will only be remembered for as long as the interrupt condition is active.

### MCU Control Register – MCUCR

The MCU Control Register contains control bits for general MCU functions.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---|---|---|---|---|---|---|---|---|---|
| $35 ($55) | SRE | SRW | SE | SM | ISC11 | ISC10 | ISC01 | ISC00 | MCUCR |
| 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 – SRE: External SRAM Enable**

When the SRE bit is set (one), the external data SRAM is enabled and the pin functions AD0 - 7 (Port A), A8 - 15 (Port C), $\overline{WR}$ and $\overline{RD}$ (Port D) are activated as the alternate pin functions. Then the SRE bit overrides any pin direction settings in the respective data direction registers. See “SRAM Data Memory – Internal and External” on page 12 for a description of the external SRAM pin functions. When the SRE bit is cleared (zero), the external data SRAM is disabled and the normal pin and data direction settings are used.

- **Bit 6 – SRW: External SRAM Wait State**

When the SRW bit is set (one), a one-cycle wait state is inserted in the external data SRAM access cycle. When the SRW bit is cleared (zero), the external data SRAM access is executed with the normal three-cycle scheme. See Figure 43 and Figure 44.

- **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 programmer's 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 section "Sleep Modes".

- **Bits 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 mask in the GIMSK are set. The level and edges on the external INT1 pin that activate the interrupt are defined in Table 5.

**Table 5.** Interrupt 1 Sense Control

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

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

The External Interrupt 0 is activated by the external pin INT0 if the SREG I-flag and the corresponding interrupt mask are set. The level and edges on the external INT0 pin that activate the interrupt are defined in Table 6.

**Table 6.** Interrupt 0 Sense Control

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

The value on the INTn pin is sampled before detecting edges. If edge interrupt is selected, pulses with a duration 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.

## Sleep Modes

To enter the sleep modes, the SE bit in MCUCR must be set (one) and a SLEEP instruction must be executed. If an enabled interrupt occurs while the MCU is in a sleep mode, the MCU awakes, executes the interrupt routine and resumes execution from the instruction following SLEEP. The contents of the register file, SRAM 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. When the MCU wakes up from Idle Mode, the CPU starts program execution immediately.

### 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 on INT0 or INT1 can wake up the MCU.

Note that when a level-triggered interrupt is used for wake-up from power-down, the low level must be held for a time longer than the reset delay Time-out period $t_{TOUT}$. Otherwise, the MCU will fail to wake up.

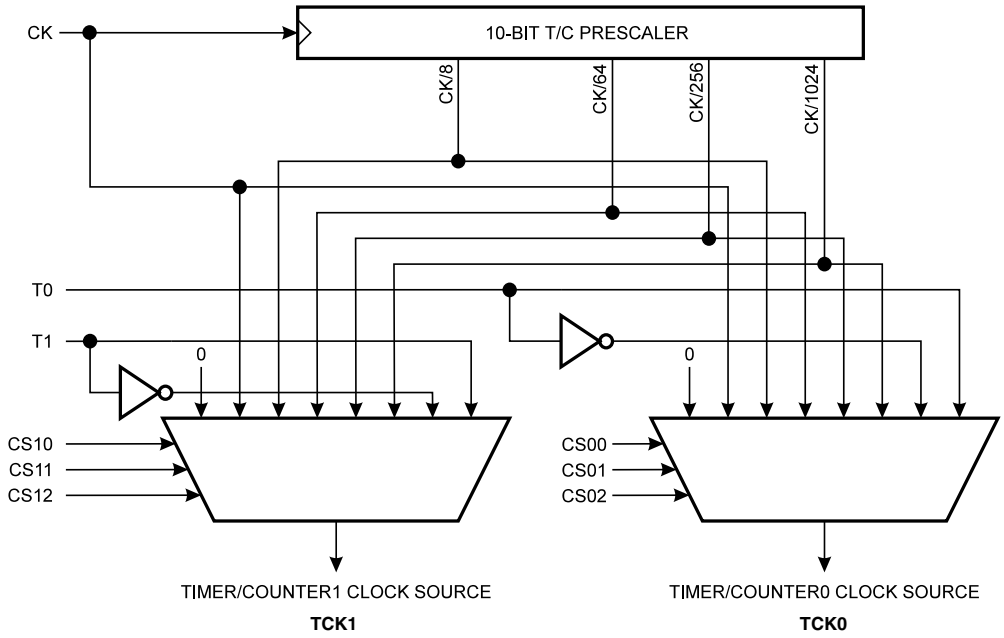
## Timer/Counters

The AT90S8515 provides two general-purpose Timer/Counters – one 8-bit T/C and one 16-bit T/C. The Timer/Counters have individual prescaling selection from the same 10-bit prescaling timer. Both Timer/Counters 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 28 shows the general Timer/Counter prescaler.

**Figure 28.** Timer/Counter Prescaler




The four different prescaled selections are: CK/8, CK/64, CK/256 and CK/1024, where CK is the oscillator clock. For the two Timer/Counters, added selections such as CK, external source and stop can be selected as clock sources.

## 8-bit Timer/Counter0

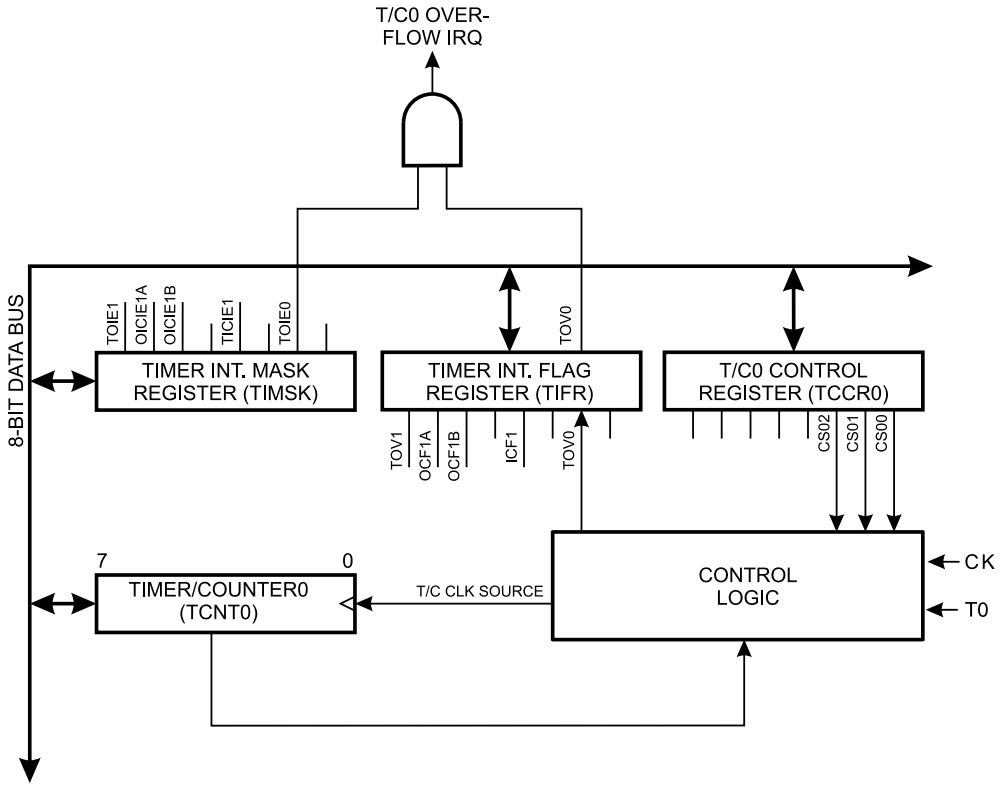
Figure 29 shows the block diagram for Timer/Counter0.

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 Timer/Counter Insterrupt Flag Register (TIFR). Control signals are found in the Timer/Counter0 Control Register (TCCR0). The interrupt enable/disable settings for Timer/Counter0 are found in the Timer/Counter Interrupt Mask Register (TIMSK).

When Timer/Counter0 is externally clocked, the external signal is synchronized with the oscillator frequency of the CPU. To assure 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.

**Figure 29.** Timer/Counter0 Block Diagram




## Timer/Counter0 Control Register – TCCR0

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---|---|---|---|---|---|---|---|---|---|
| $33 ($53) | – | – | – | – | – | CS02 | CS01 | CS00 | TCCR0 |
| Read/Write | R | R | R | R | R | R/W | R/W | R/W | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |

- **Bits 7..3 – Res: Reserved Bits**

These bits are reserved bits in the AT90S8515 and always read as zero.

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

The Clock Select0 bits 2, 1 and 0 define the prescaling source of Timer/Counter0.

**Table 7.** Clock 0 Prescale Select

| CS02 | CS01 | CS00 | Description |
|---|---|---|---|
| 0 | 0 | 0 | Stop, the Timer/Counter0 is stopped. |
| 0 | 0 | 1 | CK |
| 0 | 1 | 0 | CK/8 |
| 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 PB0/(T0) will clock the counter even if the pin is configured as an output. This feature can give the user software control of the counting.

### Timer Counter0 – TCNT0

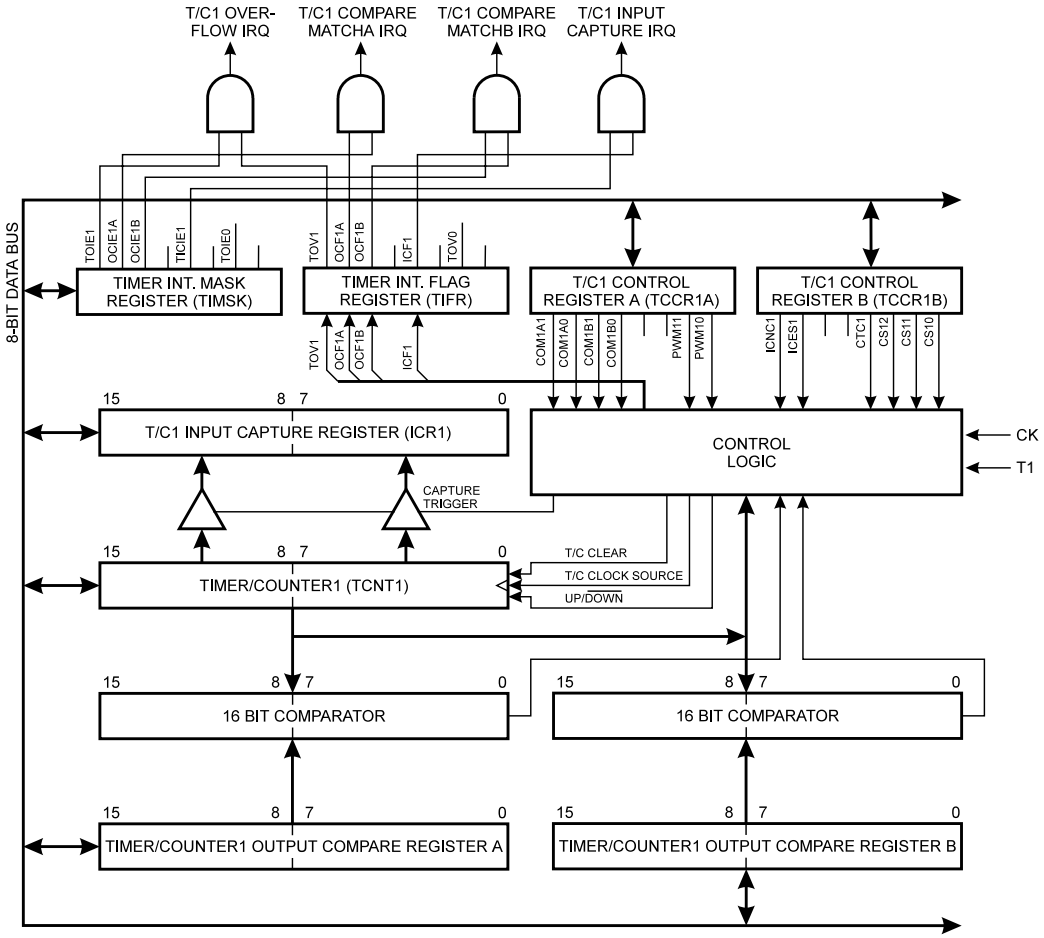
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---|---|---|---|---|---|---|---|---|---|
| $32 ($52) | MSB | | | | | | | LSB | TCNT0 |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |

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 clock cycle following the write operation.

## 16-bit Timer/Counter1

Figure 30 shows the block diagram for Timer/Counter1.

**Figure 30.** Timer/Counter1 Block Diagram




The 16-bit Timer/Counter1 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/Counter1 Control Registers (TCCR1A and TCCR1B). The different status flags (overflow, compare match and capture event) are found in the Timer/Counter Interrupt Flag Register (TIFR). Control signals are found in the Timer/Counter1 Control Registers

(TCCR1A and TCCR1B). The interrupt enable/disable settings for Timer/Counter1 are found in the Timer/Counter Interrupt Mask Register (TIMSK).

When Timer/Counter1 is externally clocked, the external signal is synchronized with the oscillator frequency of the CPU. To assure 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.

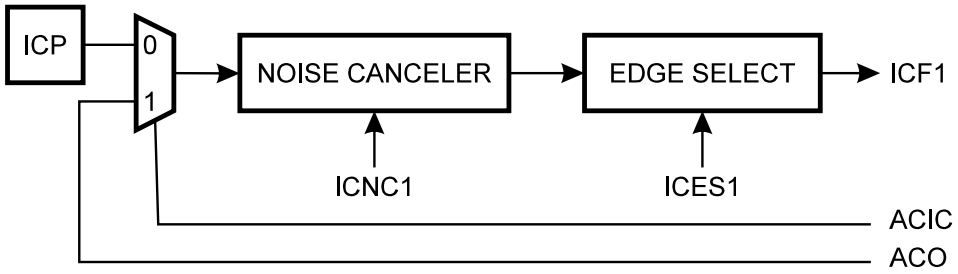
The 16-bit Timer/Counter1 features both a high-resolution and a high-accuracy usage with the lower prescaling opportunities. Similarly, the high prescaling opportunities make the Timer/Counter1 useful for lower speed functions or exact timing functions with infrequent actions.

The Timer/Counter1 supports two Output Compare functions using the Output Compare Register 1 A and B (OCR1A and OCR1B) as the data sources to be compared to the Timer/Counter1 contents. The Output Compare functions include optional clearing of the counter on compareA match and actions on the Output Compare pins on both compare matches.

Timer/Counter1 can also be used as an 8-, 9- or 10-bit Pulse Width Modulator. In this mode, the counter and the OCR1A/OCR1B registers serve as a dual, glitch-free, standalone PWM with centered pulses. Refer to page 47 for a detailed description of this function.

The Input Capture function of Timer/Counter1 provides a capture of the Timer/Counter1 contents to the Input Capture Register (ICR1), triggered by an external event on the input capture pin (ICP). The actual capture event settings are defined by the Timer/Counter1 Control Register (TCCR1B). In addition, the Analog Comparator can be set to trigger the Input Capture. Refer to “Analog Comparator” on page 59 for details on this. The ICP pin logic is shown in Figure 31.

**Figure 31.** ICP Pin Schematic Diagram




ACIC: COMPARATOR IC ENABLE
ACO: COMPARATOR OUTPUT

If the Noise Canceler function is enabled, the actual trigger condition for the capture event is monitored over four samples and all four must be equal to activate the capture flag.

### Timer/Counter1 Control Register A – TCCR1A

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---|---|---|---|---|---|---|---|---|---|
| $2F ($4F) | COM1A1 | COM1A0 | COM1B1 | COM1B0 | – | – | PWM11 | PWM10 | TCCR1A |
| Read/Write | R/W | R/W | R/W | R/W | R | R | R/W | R/W | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |

- **Bits 7, 6 – COM1A1, COM1A0: Compare Output Mode1A, Bits 1 and 0**

The COM1A1 and COM1A0 control bits determine any output pin action following a compare match in Timer/Counter1. Any output pin actions affect pin OC1A (Output CompareA pin 1). This is an alternative function to an I/O port and the corresponding direction control bit must be set (one) to control the output pin. The control configuration is shown in Table 8.

- **Bits 5, 4 – COM1B1, COM1B0: Compare Output Mode1B, Bits 1 and 0**

The COM1B1 and COM1B0 control bits determine any output pin action following a compare match in Timer/Counter1. Any output pin actions affect pin OC1B (Output CompareB). The control configuration is given in Table 8.

**Table 8.** Compare 1 Mode Select

| COM1X1 | COM1X0 | Description |
|---|---|---|
| 0 | 0 | Timer/Counter1 disconnected from output pin OC1X |
| 0 | 1 | Toggle the OC1X output line. |
| 1 | 0 | Clear the OC1X output line (to zero). |
| 1 | 1 | Set the OC1X output line (to one). |

Note: X = A or B
In PWM mode, these bits have a different function. Refer to Table 12 on page 40 for a detailed description.

- **Bits 3..2 – Res: Reserved Bits**

These bits are reserved bits in the AT90S8515 and always read zero.

- **Bits 1..0 – PWM11, PWM10: Pulse Width Modulator Select Bits 1 and 0**

These bits select PWM operation of Timer/Counter1 as specified in Table 9. This mode is described on page 40.

**Table 9.** PWM Mode Select

| PWM11 | PWM10 | Description |
|---|---|---|
| 0 | 0 | PWM operation of Timer/Counter1 is disabled |
| 0 | 1 | Timer/Counter1 is an 8-bit PWM |
| 1 | 0 | Timer/Counter1 is a 9-bit PWM |
| 1 | 1 | Timer/Counter1 is a 10-bit PWM |

### Timer/Counter1 Control Register B – TCCR1B

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---|---|---|---|---|---|---|---|---|---|
| $2E ($4E) | ICNC1 | ICES1 | – | – | CTC1 | CS12 | CS11 | CS10 | TCCR1B |
| Read/Write | R/W | R/W | R | R | R/W | R/W | R/W | R/W | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |

- **Bit 7 – ICNC1: Input Capture1 Noise Canceler (4 CKs)**

When the ICNC1 bit is cleared (zero), the input capture trigger noise canceler function is disabled. The input capture is triggered at the first rising/falling edge sampled on the ICP (input capture pin) as specified. When the ICNC1 bit is set (one), four successive samples are measured on the ICP, and all samples must be high/low according to the input capture trigger specification in the ICES1 bit. The actual sampling frequency is XTAL clock frequency.

- **Bit 6 – ICES1: Input Capture1 Edge Select**

While the ICES1 bit is cleared (zero), the Timer/Counter1 contents are transferred to the Input Capture Register (ICR1) on the falling edge of the input capture pin (ICP). While the ICES1 bit is set (one), the Timer/Counter1 contents are transferred to the ICR1 on the rising edge of the ICP.

- **Bits 5, 4 – Res: Reserved Bits**

These bits are reserved bits in the AT90S8515 and always read zero.

- **Bit 3 – CTC1: Clear Timer/Counter1 on Compare Match**

When the CTC1 control bit is set (one), the Timer/Counter1 is reset to $0000 in the clock cycle after a compareA match. If the CTC1 control bit is cleared, Timer/Counter1 continues counting and is unaffected by a compare match. Since the compare match is detected in the CPU clock cycle following the match, this function will behave differently when a prescaling higher than 1 is used for the timer. When a prescaling of 1 is used, and the compareA register is set to C, the timer will count as follows if CTC1 is set:

... | C-2 | C-1 | C | 0 | 1 | ...

When the prescaler is set to divide by 8, the timer will count like this:

... | C-2, C-2, C-2, C-2, C-2, C-2, C-2, C-2 | C-1, C-1, C-1, C-1, C-1, C-1, C-1, C-1 | C, 0, 0, 0, 0, 0, 0, 0 | ...

In PWM mode, this bit has no effect.

- **Bits 2, 1, 0 – CS12, CS11, CS10: Clock Select1, Bits 2, 1 and 0**

The Clock Select1 bits 2, 1 and 0 define the prescaling source of Timer/Counter1.

**Table 10.** Clock 1 Prescale Select

| CS12 | CS11 | CS10 | Description |
|---|---|---|---|
| 0 | 0 | 0 | Stop, the Timer/Counter1 is stopped. |
| 0 | 0 | 1 | CK |
| 0 | 1 | 0 | CK/8 |
| 0 | 1 | 1 | CK/64 |
| 1 | 0 | 0 | CK/256 |
| 1 | 0 | 1 | CK/1024 |
| 1 | 1 | 0 | External Pin T1, falling edge |
| 1 | 1 | 1 | External Pin T1, 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/Counter1, transitions on PB1/(T1) will clock the counter even if the pin is configured as an output. This feature can give the user software control of the counting.

### Timer/Counter1 – TCNT1H AND TCNT1L

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | |
|---|---|---|---|---|---|---|---|---|---|
| $2D ($4D) | MSB | | | | | | | | TCNT1H |
| $2C ($4C) | | | | | | | | LSB | TCNT1L |
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |

This 16-bit register contains the prescaled value of the 16-bit Timer/Counter1. To ensure that both the high and low bytes are read and written simultaneously when the CPU accesses these registers, the access is performed using an 8-bit temporary register (TEMP). This temporary register is also used when accessing OCR1A, OCR1B and ICR1. If the main program and interrupt routines perform access to registers using TEMP, interrupts must be disabled during access from the main program (and from interrupt routines if interrupts are allowed from within interrupt routines).

- TCNT1 Timer/Counter1 Write:
  When the CPU writes to the high byte TCNT1H, the written data is placed in the TEMP register. Next, when the CPU writes the low byte TCNT1L, this byte of data is combined with the byte data in the TEMP register, and all 16 bits are written to the TCNT1 Timer/Counter1 register simultaneously. Consequently, the high byte TCNT1H must be accessed first for a full 16-bit register write operation.
- TCNT1 Timer/Counter1 Read:
  When the CPU reads the low byte TCNT1L, the data of the low byte TCNT1L is sent to the CPU and the data of the high byte TCNT1H is placed in the TEMP register. When the CPU reads the data in the high byte TCNT1H, the CPU receives the data in the TEMP register. Consequently, the low byte TCNT1L must be accessed first for a full 16-bit register read operation.

The Timer/Counter1 is realized as an up or up/down (in PWM mode) counter with read and write access. If Timer/Counter1 is written to and a clock source is selected, the Timer/Counter1 continues counting in the timer clock cycle after it is preset with the written value.

### Timer/Counter1 Output Compare Register – OCR1AH AND OCR1AL

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | |
|---|---|---|---|---|---|---|---|---|---|
| $2B ($4B) | MSB | | | | | | | | OCR1AH |
| $2A ($4A) | | | | | | | | LSB | OCR1AL |
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |

### Timer/Counter1 Output Compare Register – OCR1BH AND OCR1BL

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | |
|---|---|---|---|---|---|---|---|---|---|
| $29 ($49) | MSB | | | | | | | | OCR1BH |
| $28 ($48) | | | | | | | | LSB | OCR1BL |
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | |
| | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |

The output compare registers are 16-bit read/write registers.

The Timer/Counter1 Output Compare registers contain the data to be continuously compared with Timer/Counter1. Actions on compare matches are specified in the Timer/Counter1 Control and Status registers. A compare match only occurs if Timer/Counter1 counts to the OCR value. A software write that sets TCNT1 and OCR1A or OCR1B to the same value does not generate a compare match.

A compare match will set the compare interrupt flag in the CPU clock cycle following the compare event.

Since the Output Compare Registers (OCR1A and OCR1B) are 16-bit registers, a temporary register (TEMP) is used when OCR1A/B are written to ensure that both bytes are updated simultaneously. When the CPU writes the high byte, OCR1AH or OCR1BH, the data is temporarily stored in the TEMP register. When the CPU writes the low byte, OCR1AL or OCR1BL, the TEMP register is simultaneously written to OCR1AH or OCR1BH. Consequently, the high byte OCR1AH or OCR1BH must be written first for a full 16-bit register write operation.

The TEMP register is also used when accessing TCNT1 and ICR1. If the main program and interrupt routines perform access to registers using TEMP, interrupts must be disabled during access from the main program (and from interrupt routines if interrupts are allowed from within interrupt routines).

### Timer/Counter1 Input Capture Register – ICR1H AND ICR1L

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | |
|---|---|---|---|---|---|---|---|---|---|
| $25 ($45) | MSB | | | | | | | | ICR1H |
| $24 ($44) | | | | | | | | LSB | ICR1L |
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
| Read/Write | R | R | R | R | R | R | R | R | |
| | R | R | R | R | R | R | R | R | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |

The input capture register is a 16-bit read-only register.

When the rising or falling edge (according to the input capture edge setting [ICES1]) of the signal at the input capture pin (ICP) is detected, the current value of the Timer/Counter1 is transferred to the Input Capture Register (ICR1). At the same time, the input capture flag (ICF1) is set (one).

Since the Input Capture Register (ICR1) is a 16-bit register, a temporary register (TEMP) is used when ICR1 is read to ensure that both bytes are read simultaneously. When the CPU reads the low byte ICR1L, the data is sent to the CPU and the data of the high byte ICR1H is placed in the TEMP register. When the CPU reads the data in the high byte ICR1H, the CPU receives the data in the TEMP register. Consequently, the low byte ICR1L must be accessed first for a full 16-bit register read operation.

The TEMP register is also used when accessing TCNT1, OCR1A and OCR1B. If the main program and interrupt routines perform access to registers using TEMP, interrupts must be disabled during access from the main program (and from interrupt routines if interrupts are allowed from within interrupt routines).

### Timer/Counter1 in PWM Mode

When the PWM mode is selected, Timer/Counter1, the Output Compare Register1A (OCR1A) and the Output Compare Register1B (OCR1B) form a dual 8-, 9- or 10-bit, free-running, glitch-free and phase-correct PWM with outputs on the PD5(OC1A) and OC1B pins. Timer/Counter1 acts as an up/down counter, counting up from $0000 to TOP (see Table 11), where it turns and counts down again to zero before the cycle is repeated. When the counter value matches the contents of the 10 least significant bits of OCR1A or OCR1B, the PD5(OC1A)/OC1B pins are set or cleared according to the settings of the COM1A1/COM1A0 or COM1B1/COM1B0 bits in the Timer/Counter1 Control Register (TCCR1A). Refer to Table 12 for details.

**Table 11.** Timer TOP Values and PWM Frequency

| PWM Resolution | Timer TOP Value | Frequency |
|---|---|---|
| 8-bit | $00FF (255) | $f_{TCK1}/510$ |
| 9-bit | $01FF (511) | $f_{TCK1}/1022$ |
| 10-bit | $03FF(1023) | $f_{TCK1}/2046$ |

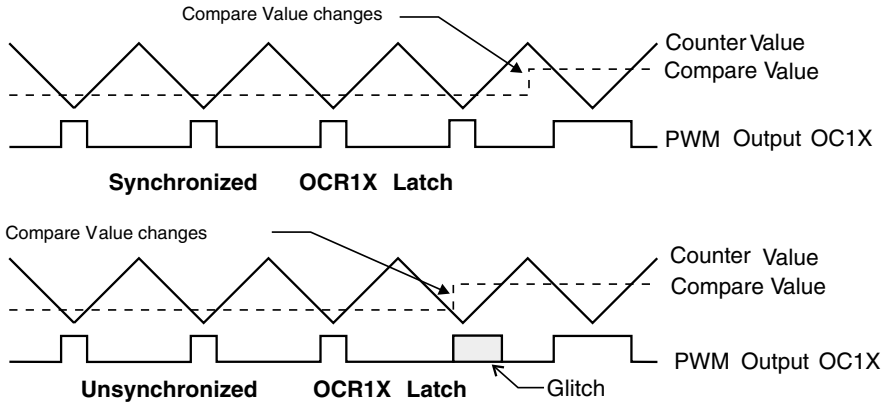
**Table 12.** Compare1 Mode Select in PWM Mode

| COM1X1 | COM1X0 | Effect on OCX1 |
|---|---|---|
| 0 | 0 | Not connected |
| 0 | 1 | Not connected |
| 1 | 0 | Cleared on compare match, up-counting. Set on compare match, down-counting (non-inverted PWM). |
| 1 | 1 | Cleared on compare match, down-counting. Set on compare match, up-counting (inverted PWM). |

Note: X = A or B

Note that in the PWM mode, the 10 least significant OCR1A/OCR1B bits, when written, are transferred to a temporary location. They are latched when Timer/Counter1 reaches the value TOP. This prevents the occurrence of odd-length PWM pulses (glitches) in the event of an unsynchronized OCR1A/OCR1B write. See Figure 32 for an example.

**Figure 32.** Effects on Unsynchronized OCR1 Latching




Note: X = A or B

During the time between the write and the latch operation, a read from OCR1A or OCR1B will read the contents of the temporary location. This means that the most recently written value always will read out of OCR1A/B.

When the OCR1 contains $0000 or TOP, the output OC1A/OC1B is updated to low or high on the next compare match according to the settings of COM1A1/COM1A0 or COM1B1/COM1B0. This is shown in Table 13.

Note: If the compare register contains TOP value and the prescaler is not in use (CS12..CS10 = 001), the PWM output will not produce any pulse at all, because up-counting and down-counting values are reached simultaneously. When the prescaler is in use (CS12..CS10 ≠ 001 or 000), the PWM output goes active when the counter reaches the TOP value; but the down-counting compare match is not interpreted to be reached before the next time the counter reaches the TOP value, making a one-period PWM pulse.

**Table 13.** PWM Outputs OCR1X = $0000 or TOP

| COM1X1 | COM1X0 | OCR1X | Output OC1X |
|---|---|---|---|
| 1 | 0 | $0000 | L |
| 1 | 0 | TOP | H |
| 1 | 1 | $0000 | H |
| 1 | 1 | TOP | L |

Note: X = A or B

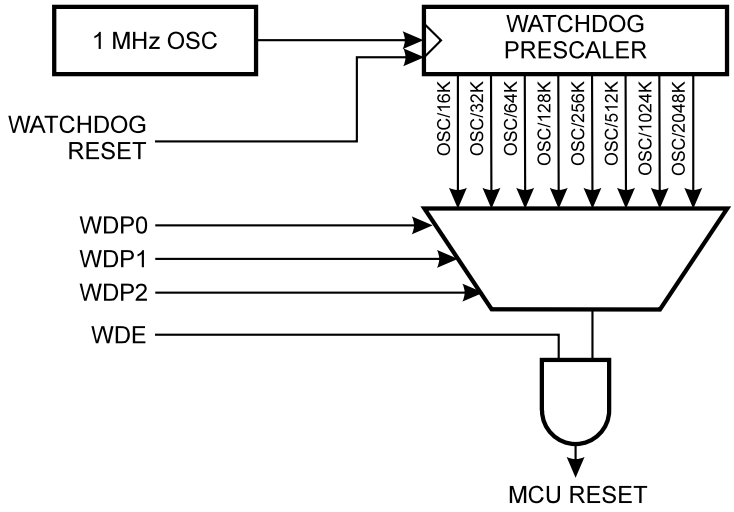
In PWM mode, the Timer Overflow Flag1 (TOV1) is set when the counter advances from $0000. Timer Overflow Interrupt1 operates exactly as in normal Timer/Counter mode, i.e., it is executed when TOV1 is set, provided that Timer Overflow Interrupt1 and global interrupts are enabled. This also applies to the Timer Output Compare1 flags and interrupts.

## Watchdog Timer

The Watchdog Timer is clocked from a separate On-chip oscillator that runs at 1 MHz. This is the typical value at $V_{CC}$ = 5V. See characterization data for typical values at other $V_{CC}$ levels. By controlling the Watchdog Timer prescaler, the Watchdog reset interval can be adjusted (see Table 14 for a detailed description). 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 AT90S8515 resets and executes from the reset vector. For timing details on the Watchdog reset, refer to page 25.

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 33.** Watchdog Timer




### Watchdog Timer Control Register – WDTCR

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---|---|---|---|---|---|---|---|---|---|
| $21 ($41) | – | – | – | 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 AT90S8515 and will always read as zero.

- **Bit 4 – WDTOE: Watchdog 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: Watchdog 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 “1” to WDTOE and WDE. A logical “1” 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 14.

**Table 14.** Watchdog Timer Prescale Select

| WDP2 | WDP1 | WDP0 | Number of WDT Oscillator Cycles | Typical Time-out at $V_{CC}$ = 3.0V | Typical Time-out at $V_{CC}$ = 5.0V |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 16K cycles | 47.0 ms | 15.0 ms |
| 0 | 0 | 1 | 32K cycles | 94.0 ms | 30.0 ms |
| 0 | 1 | 0 | 64K cycles | 0.19 s | 60.0 ms |
| 0 | 1 | 1 | 128K cycles | 0.38 s | 0.12 s |
| 1 | 0 | 0 | 256K cycles | 0.75 s | 0.24 s |
| 1 | 0 | 1 | 512K cycles | 1.5 s | 0.49 s |
| 1 | 1 | 0 | 1,024K cycles | 3.0 s | 0.97 s |
| 1 | 1 | 1 | 2,048K cycles | 6.0 s | 1.9 s |

Note: The frequency of the Watchdog oscillator is voltage-dependent as shown in the Electrical Characteristics section.
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 to count from zero.
To avoid unintentional MCU reset, the Watchdog Timer should be disabled or reset before changing the Watchdog Timer Prescale Select.

## EEPROM Read/Write Access

The EEPROM access registers are accessible in the I/O space.

The write access time is in the range of 2.5 - 4 ms, depending on the $V_{CC}$ voltages. A self-timing function, however, lets the user software detect when the next byte can be written. If the user code contains code that writes the EEPROM, some precaution must be taken. In heavily filtered power supplies, $V_{CC}$ is likely to rise or fall slowly on power-up/down. This causes the device for some period of time to run at a voltage lower than specified as minimum for the clock frequency used. CPU operation under these conditions is likely cause the program counter to perform unintentional jumps and eventually execute the EEPROM write code. To secure EEPROM integrity, the user is advised to use an external under-voltage reset circuit in this case.

In order to prevent unintentional EEPROM writes, a specific write procedure must be followed. Refer to the description of the EEPROM Control Register for details on this.

When the EEPROM is written, the CPU is halted for two clock cycles before the next instruction is executed. When the EEPROM is read, the CPU is halted for four clock cycles before the next instruction is executed.

### EEPROM Address Register – EEARH and EEARL

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | |
|---|---|---|---|---|---|---|---|---|---|
| $1F ($3F) | – | – | – | – | – | – | – | EEAR8 | EEARH |
| $1E ($3E) | EEAR7 | EEAR6 | EEAR5 | EEAR4 | EEAR3 | EEAR2 | EEAR1 | EEAR0 | EEARL |
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
| Read/Write | R | R | R | R | R | R | R | R/W | |
| | 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 | |
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |

The EEPROM address registers (EEARH and EEARL) specify the EEPROM address in the 512-byte EEPROM space for AT90S8515. The EEPROM data bytes are addressed linearly between 0 and 512.

### EEPROM Data Register – EEDR

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---|---|---|---|---|---|---|---|---|---|
| $1D ($3D) | MSB | | | | | | | LSB | EEDR |
| 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..0 – EEDR7..0: EEPROM Data**

For the EEPROM write operation, the EEDR register contains the data to be written to the EEPROM in the address given by the EEAR register. For the EEPROM read operation, the EEDR contains the data read out from the EEPROM at the address given by EEAR.

### EEPROM Control Register – EECR

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---|---|---|---|---|---|---|---|---|---|
| $1C ($3C) | – | – | – | – | – | EEMWE | EEWE | EERE | EECR |
| Read/Write | R | R | R | R | R | R/W | R/W | R/W | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |

- **Bits 7..3 – Res: Reserved Bits**

These bits are reserved bits in the AT90S8515 and will always read as zero.

- **Bit 2 – EEMWE: EEPROM Master Write Enable**

The EEMWE bit determines whether setting EEWE to one causes the EEPROM to be written. When EEMWE is set (one), setting EEWE will write data to the EEPROM at the selected address. If EEMWE is zero, setting EEWE will have no effect. When EEMWE has been set (one) by software, hardware clears the bit to zero after four clock cycles. See the description of the EEWE bit for a EEPROM write procedure.

- **Bit 1 – EEWE: EEPROM Write Enable**

The EEPROM Write Enable signal (EEWE) is the write strobe to the EEPROM. When address and data are correctly set up, the EEWE bit must be set to write the value into the EEPROM. The EEMWE bit must be set when the logical “1” is written to EEWE, otherwise no EEPROM write takes place. The following procedure should be followed when writing the EEPROM (the order of steps 2 and 3 is unessential):

1. Wait until EEWE becomes zero.
2. Write new EEPROM address to EEARL and EEARH (optional).
3. Write new EEPROM data to EEDR (optional).
4. Write a logical “1” to the EEMWE bit in EECR (to be able to write a logical “1” to the EEMWE bit, the EEWE bit must be written to zero in the same cycle).
5. Within four clock cycles after setting EEMWE, write a logical “1” to EEWE.

When the write access time (typically 2.5 ms at $V_{CC}$ = 5V or 4 ms at $V_{CC}$ = 2.7V) has elapsed, the EEWE bit is cleared (zero) by hardware. The user software can poll this bit and wait for a zero before writing the next byte. When EEWE has been set, the CPU is halted for two cycles before the next instruction is executed.

**Caution**: An interrupt between step 4 and step 5 will make the write cycle fail, since the EEPROM Master Write Enable will time-out. If an interrupt routine accessing the EEPROM is interrupting another EEPROM access, the EEAR or EEDR registers will be modified, causing the interrupted EEPROM access to fail. It is recommended to have the global interrupt flag cleared during the four last steps to avoid these problems.

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

The EEPROM Read Enable signal EERE is the read strobe to the EEPROM. When the correct address is set up in the EEAR register, the EERE bit must be set. When the EERE bit is cleared (zero) by hardware, requested data is found in the EEDR register. The EEPROM read access takes one instruction and there is no need to poll the EERE bit. When EERE has been set, the CPU is halted for four cycles before the next instruction is executed.

The user should poll the EEWE bit before starting the read operation. If a write operation is in progress when new data or address is written to the EEPROM I/O registers, the write operation will be interrupted and the result is undefined.

## Prevent EEPROM Corruption

During periods of low $V_{CC}$, the EEPROM data can be corrupted because the supply voltage is too low for the CPU and the EEPROM to operate properly. These issues are the same as for board level systems using the EEPROM and the same design solutions should be applied.

An EEPROM data corruption can be caused by two situations when the voltage is too low. First, a regular write sequence to the EEPROM requires a minimum voltage to operate correctly. Second, the CPU itself can execute instructions incorrectly if the supply voltage for executing instructions is too low.

EEPROM data corruption can easily be avoided by following these design recommendations (one is sufficient):

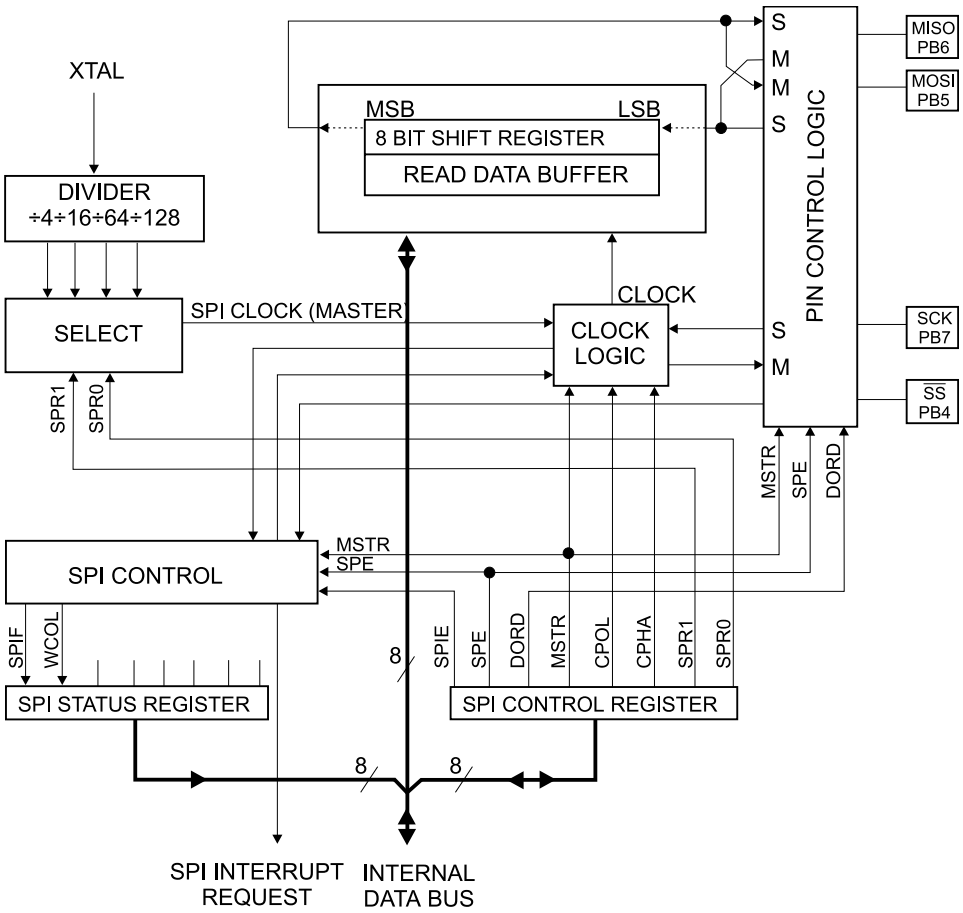
1. Keep the AVR RESET active (low) during periods of insufficient power supply voltage. This is best done by an external low $V_{CC}$ Reset Protection circuit, often referred to as a Brown-out Detector (BOD). Please refer to application note AVR 180 for design considerations regarding power-on reset and low-voltage detection.
2. Keep the AVR core in Power-down Sleep mode during periods of low $V_{CC}$. This will prevent the CPU from attempting to decode and execute instructions, effectively protecting the EEPROM registers from unintentional writes.
3. Store constants in Flash memory if the ability to change memory contents from software is not required. Flash memory cannot be updated by the CPU and will not be subject to corruption.

## Serial Peripheral Interface – SPI

The Serial Peripheral Interface (SPI) allows high-speed synchronous data transfer between the AT90S8515 and peripheral devices or between several AVR devices. The AT90S8515 SPI features include the following:

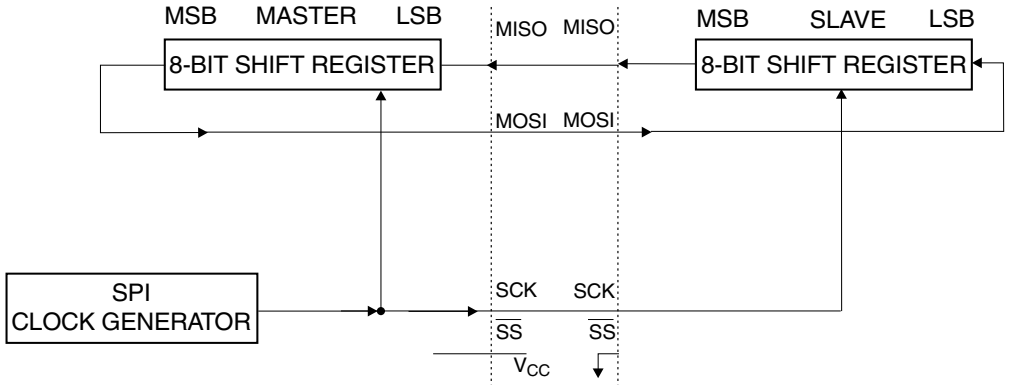
- **Full-duplex, 3-wire Synchronous Data Transfer**
- **Master or Slave Operation**
- **LSB First or MSB First Data Transfer**
- **Four Programmable Bit Rates**
- **End-of-Transmission Interrupt Flag**
- **Write Collision Flag Protection**
- **Wake-up from Idle Mode (Slave Mode Only)**

**Figure 34.** SPI Block Diagram




The interconnection between master and slave CPUs with SPI is shown in Figure 35. The PB7(SCK) pin is the clock output in the Master Mode and is the clock input in the Slave Mode. Writing to the SPI Data Register of the master CPU starts the SPI clock generator and the data written shifts out of the PB5(MOSI) pin and into the PB5(MOSI) pin of the slave CPU. After shifting one byte, the SPI clock generator stops, setting the end-of-transmission flag (SPIF). If the SPI interrupt enable bit (SPIE) in the SPCR register is set, an interrupt is requested. The Slave Select input, PB4($\overline{SS}$), is set low to select an individual slave SPI device. The two shift registers in the master and the slave can be considered as one distributed 16-bit circular shift register. This is shown in Figure 35. When data is shifted from the master to the slave, data is also shifted in the opposite direction, simultaneously. This means that during one shift cycle, data in the master and the slave are interchanged.

**Figure 35.** SPI Master-slave Interconnection




The system is single-buffered in the transmit direction and double-buffered in the receive direction. This means that bytes to be transmitted cannot be written to the SPI Data Register before the entire shift cycle is completed. When receiving data, however, a received byte must be read from the SPI Data Register before the next byte has been completely shifted in. Otherwise, the first byte is lost.

When the SPI is enabled, the data direction of the MOSI, MISO, SCK and $\overline{SS}$ pins is overridden according to Table 15.

**Table 15.** SPI Pin Overrides

| Pin | Direction, Master SPI | Direction, Slave SPI |
|---|---|---|
| MOSI | User Defined | Input |
| MISO | Input | User Defined |
| SCK | User Defined | Input |
| $\overline{SS}$ | User Defined | Input |

Note: See “Alternate Functions of Port B” on page 66 for a detailed description of how to define the direction of the user-defined SPI pins.

## $\overline{SS}$ Pin Functionality

When the SPI is configured as a master (MSTR in SPCR is set), the user can determine the direction of the $\overline{SS}$ pin. If $\overline{SS}$ is configured as an output, the pin is a general output pin, which does not affect the SPI system. If $\overline{SS}$ is configured as an input, it must be held high to ensure master SPI operation. If the $\overline{SS}$ pin is driven low by peripheral circuitry when the SPI is configured as master with the $\overline{SS}$ pin defined as an input, the SPI system interprets this as another master selecting the SPI as a slave and starts to send data to it. To avoid bus contention, the SPI system takes the following actions:

1. The MSTR bit in SPCR is cleared and the SPI system becomes a slave. As a result of the SPI becoming a slave, the MOSI and SCK pins become inputs.
2. The SPIF flag in SPSR is set, and if the SPI interrupt is enabled and the I-bit in SREG is set, the interrupt routine will be executed.

Thus, when interrupt-driven SPI transmittal is used in Master Mode and there exists a possibility that $\overline{SS}$ is driven low, the interrupt should always check that the MSTR bit is still set. Once the MSTR bit has been cleared by a slave select, it must be set by the user to re-enable SPI Master Mode.

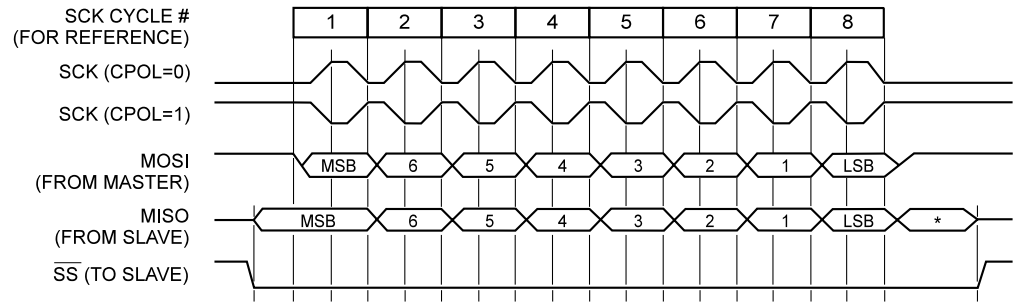
When the SPI is configured as a slave, the $\overline{SS}$ pin is always input. When $\overline{SS}$ is held low, the SPI is activated and MISO becomes an output if configured so by the user. All other

pins are inputs. When $\overline{SS}$ is driven high, all pins are inputs and the SPI is passive, which means that it will not receive incoming data. Note that the SPI logic will be reset once the $\overline{SS}$ pin is brought high. If the $\overline{SS}$ pin is brought high during a transmission, the SPI will stop sending and receiving immediately and both data received and data sent must be considered as lost.

## Data Modes

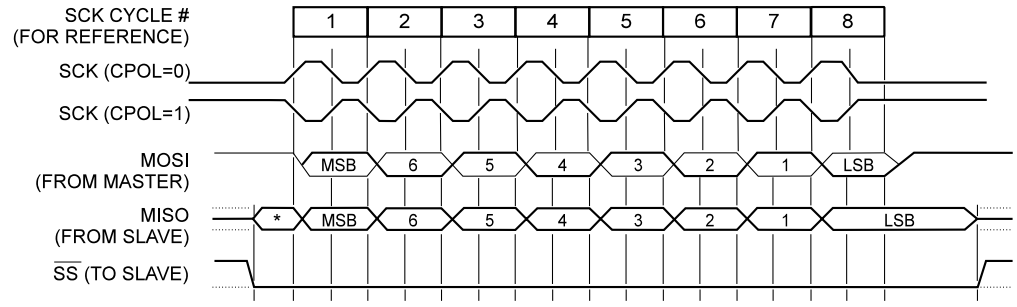
There are four combinations of SCK phase and polarity with respect to serial data, which are determined by control bits CPHA and CPOL. The SPI data transfer formats are shown in Figure 36 and Figure 37.

**Figure 36.** SPI Transfer Format with CPHA = 0 and DORD = 0




* Not defined but normally MSB of character just received

**Figure 37.** SPI Transfer Format with CPHA = 1 and DORD = 0




* Not defined but normally LSB of previously transmitted character

### SPI Control Register – SPCR

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---|---|---|---|---|---|---|---|---|---|
| $0D ($2D) | SPIE | SPE | DORD | MSTR | CPOL | CPHA | SPR1 | SPR0 | SPCR |
| 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 – SPIE: SPI Interrupt Enable**

This bit causes the SPI interrupt to be executed if SPIF bit in the SPSR register is set and the global interrupts are enabled.

- **Bit 6 – SPE: SPI Enable**

When the SPE bit is set (one), the SPI is enabled. This bit must be set to enable any SPI operations.

- **Bit 5 – DORD: Data Order**

When the DORD bit is set (one), the LSB of the data word is transmitted first.

When the DORD bit is cleared (zero), the MSB of the data word is transmitted first.

- **Bit 4 – MSTR: Master/Slave Select**

This bit selects Master SPI Mode when set (one), and Slave SPI Mode when cleared (zero). If $\overline{SS}$ is configured as an input and is driven low while MSTR is set, MSTR will be cleared and SPIF in SPSR will become set. The user will then have to set MSTR to re-enable SPI Master Mode.

- **Bit 3 – CPOL: Clock Polarity**

When this bit is set (one), SCK is high when idle. When CPOL is cleared (zero), SCK is low when idle. Refer to Figure 36 and Figure 37 for additional information.

- **Bit 2 – CPHA: Clock Phase**

Refer to Figure 36 or Figure 37 for the functionality of this bit.

- **Bits 1, 0 – SPR1, SPR0: SPI Clock Rate Select 1 and 0**

These two bits control the SCK rate of the device configured as a master. SPR1 and SPR0 have no effect on the slave. The relationship between SCK and the oscillator clock frequency $f_{cl}$ is shown in Table 16.

**Table 16.** Relationship between SCK and the Oscillator Frequency

| SPR1 | SPR0 | SCK Frequency |
|---|---|---|
| 0 | 0 | $f_{cl}/4$ |
| 0 | 1 | $f_{cl}/16$ |
| 1 | 0 | $f_{cl}/64$ |
| 1 | 1 | $f_{cl}/128$ |

### SPI Status Register – SPSR

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---|---|---|---|---|---|---|---|---|---|
| $0E ($2E) | SPIF | WCOL | – | – | – | – | – | – | SPSR |
| Read/Write | R | R | R | R | R | R | R | R | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |

- **Bit 7 – SPIF: SPI Interrupt Flag**

When a serial transfer is complete, the SPIF bit is set (one) and an interrupt is generated if SPIE in SPCR is set (one) and global interrupts are enabled. If $\overline{SS}$ is an input and is driven low when the SPI is in Master Mode, this will also set the SPIF flag. SPIF is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, the SPIF bit is cleared by first reading the SPI Status Register when SPIF is set (one), then by accessing the SPI Data Register (SPDR).

- **Bit 6 – WCOL: Write Collision Flag**

The WCOL bit is set if the SPI Data Register (SPDR) is written during a data transfer. The WCOL bit (and the SPIF bit) are cleared (zero) by first reading the SPI Status Register when WCOL is set (one), and then by accessing the SPI Data Register.

- **Bits 5..0 – Res: Reserved Bits**

These bits are reserved bits in the AT90S8515 and will always read as zero.

The SPI interface on the AT90S8515 is also used for program memory and EEPROM downloading or uploading. See page 86 for serial programming and verification.

### SPI Data Register – SPDR

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---|---|---|---|---|---|---|---|---|---|
| $0F ($2F) | MSB | | | | | | | LSB | SPDR |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | |
| Initial Value | x | x | x | x | x | x | x | x | Undefined |

The SPI Data Register is a read/write register used for data transfer between the register file and the SPI Shift Register. Writing to the register initiates data transmission. Reading the register causes the Shift Register Receive buffer to be read.

## UART

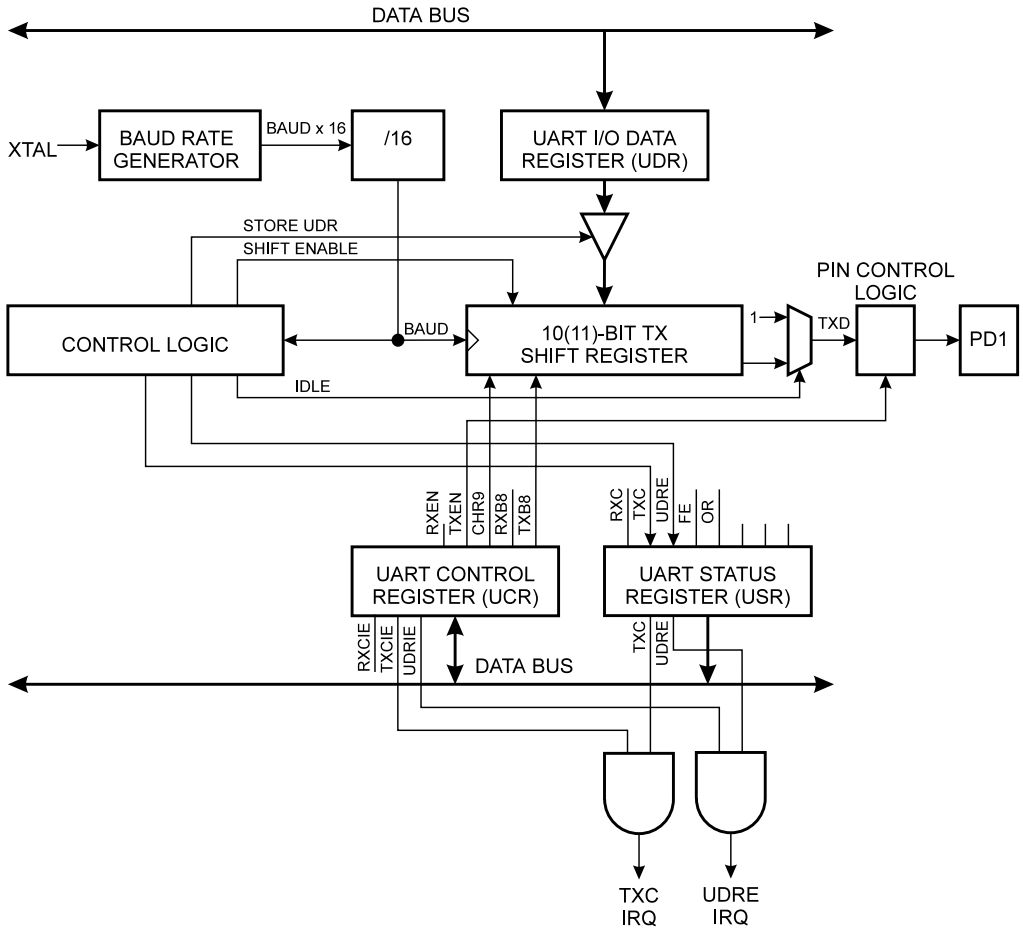
The AT90S8515 features a full duplex (separate receive and transmit registers) Universal Asynchronous Receiver and Transmitter (UART). The main features are:

- **Baud Rate Generator that can Generate a large Number of Baud Rates (bps)**
- **High Baud Rates at Low XTAL Frequencies**
- **8 or 9 Bits Data**
- **Noise Filtering**
- **Overrun Detection**
- **Framing Error Detection**
- **False Start Bit Detection**
- **Three separate Interrupts on TX Complete, TX Data Register Empty and RX Complete**

## Data Transmission

A block schematic of the UART transmitter is shown in Figure 38.

**Figure 38.** UART Transmitter




Data transmission is initiated by writing the data to be transmitted to the UART I/O Data Register, UDR. Data is transferred from UDR to the Transmit shift register when:

- A new character has been written to UDR after the stop bit from the previous character has been shifted out. The shift register is loaded immediately.
- A new character has been written to UDR before the stop bit from the previous character has been shifted out. The shift register is loaded when the stop bit of the character currently being transmitted has been shifted out.

If the 10(11)-bit Transmitter shift register is empty, data is transferred from UDR to the shift register. At this time the UDRE (UART Data Register Empty) bit in the UART Status Register, USR, is set. When this bit is set (one), the UART is ready to receive the next character. At the same time as the data is transferred from UDR to the 10(11)-bit shift register, bit 0 of the shift register is cleared (start bit) and bit 9 or 10 is set (stop bit). If 9-bit data word is selected (the CHR9 bit in the UART Control Register, UCR is set), the TXB8 bit in UCR is transferred to bit 9 in the Transmit shift register.

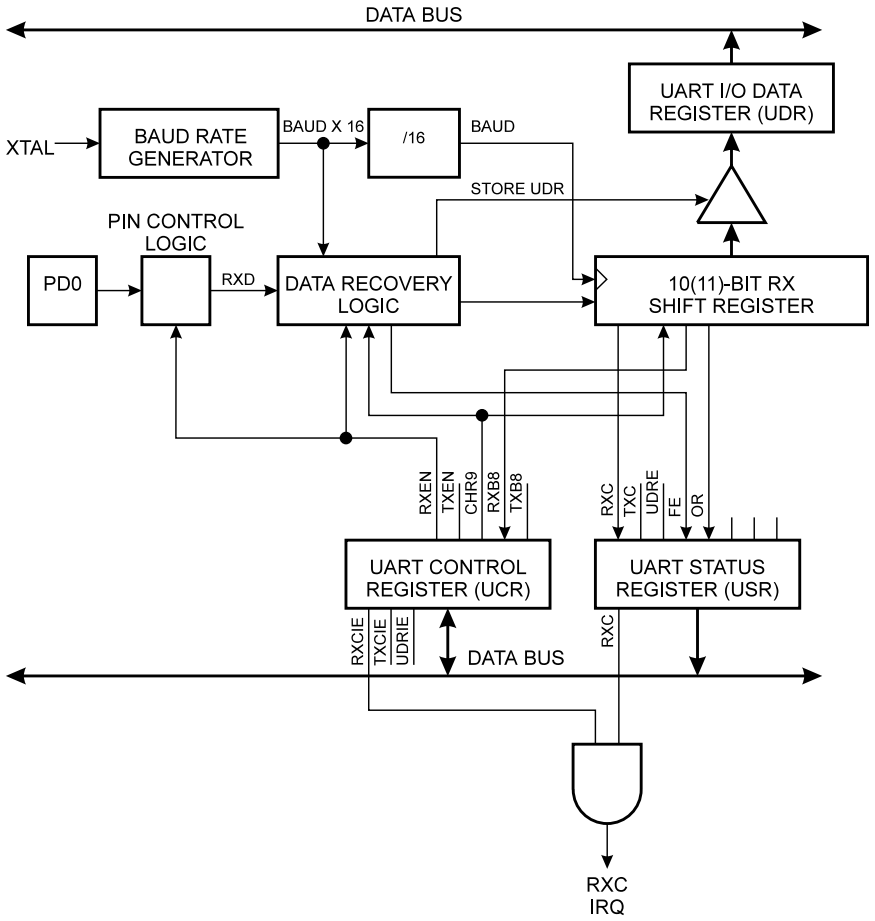
On the baud rate clock following the transfer operation to the shift register, the start bit is shifted out on the TXD pin. Then follows the data, LSB first. When the stop bit has been shifted out, the shift register is loaded if any new data has been written to the UDR during the transmission. During loading, UDRE is set. If there is no new data in the UDR register to send when the stop bit is shifted out, the UDRE flag will remain set until UDR is written again. When no new data has been written and the stop bit has been present on TXD for one bit length, the TX Complete flag (TXC) in USR is set.

The TXEN bit in UCR enables the UART Transmitter when set (one). When this bit is cleared (zero), the PD1 pin can be used for general I/O. When TXEN is set, the UART Transmitter will be connected to PD1, which is forced to be an output pin regardless of the setting of the DDD1 bit in DDRD.

## Data Reception

Figure 39 shows a block diagram of the UART Receiver.

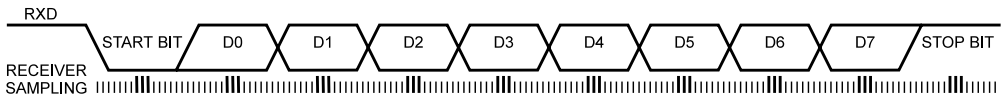
**Figure 39.** UART Receiver



The receiver front-end logic samples the signal on the RXD pin at a frequency 16 times the baud rate. While the line is idle, one single sample of logical “0” will be interpreted as the falling edge of a start bit and the start bit detection sequence is initiated. Let sample 1 denote the first zero-sample. Following the 1-to-0 transition, the receiver samples the RXD pin at samples 8, 9 and 10. If two or more of these three samples are found to be logical “1”s, the start bit is rejected as a noise spike and the receiver starts looking for the next 1-to-0 transition.

If, however, a valid start bit is detected, sampling of the data bits following the start bit is performed. These bits are also sampled at samples 8, 9 and 10. The logical value found in at least two of the three samples is taken as the bit value. All bits are shifted into the Transmitter Shift register as they are sampled. Sampling of an incoming character is shown in Figure 40.

**Figure 40.** Sampling Received Data




When the stop bit enters the receiver, the majority of the three samples must be “1” to accept the stop bit. If two or more samples are logical “0”s, the Framing Error (FE) flag in the UART Status Register (USR) is set. Before reading the UDR register, the user should always check the FE bit to detect framing errors.

Whether or not a valid stop bit is detected at the end of a character reception cycle, the data is transferred to UDR and the RXC flag in USR is set. UDR is in fact two physically separate registers, one for transmitted data and one for received data. When UDR is read, the Receive Data register is accessed, and when UDR is written, the Transmit Data register is accessed. If 9-bit data word is selected (the CHR9 bit in the UART Control Register, UCR is set), the RXB8 bit in UCR is loaded with bit 9 in the Transmit Shift register when data is transferred to UDR.

If, after having received a character, the UDR register has not been read since the last receive, the OverRun (OR) flag in USR is set. This means that the last data byte shifted into the shift register could not be transferred to UDR and has been lost. The OR bit is buffered and is updated when the valid data byte in UDR is read. Thus, the user should always check the OR bit after reading the UDR register in order to detect any overruns if the baud rate is high or CPU load is high.

When the RXEN bit in the UCR register is cleared (zero), the receiver is disabled. This means that the PD0 pin can be used as a general I/O pin. When RXEN is set, the UART Receiver will be connected to PD0, which is forced to be an input pin regardless of the setting of the DDD0 bit in DDRD. When PD0 is forced to input by the UART, the PORTD0 bit can still be used to control the pull-up resistor on the pin.

When the CHR9 bit in the UCR register is set, transmitted and received characters are 9 bits long, plus start and stop bits. The ninth data bit to be transmitted is the TXB8 bit in UCR register. This bit must be set to the wanted value before a transmission is initiated by writing to the UDR register. The ninth data bit received is the RXB8 bit in the UCR register.

## UART Control

### UART I/O Data Register – UDR

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---|---|---|---|---|---|---|---|---|---|
| $0C ($2C) | MSB | | | | | | | LSB | UDR |
| 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 UDR register is actually two physically separate registers sharing the same I/O address. When writing to the register, the UART Transmit Data register is written. When reading from UDR, the UART Receive Data register is read.

### UART Status Register – USR

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---|---|---|---|---|---|---|---|---|---|
| $0B ($2B) | RXC | TXC | UDRE | FE | OR | – | – | – | USR |
| Read/Write | R | R/W | R | R | R | R | R | R | |
| Initial Value | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | |

The USR register is a read-only register providing information on the UART status.

- **Bit 7 – RXC: UART Receive Complete**

This bit is set (one) when a received character is transferred from the Receiver Shift register to UDR. The bit is set regardless of any detected framing errors. When the RXCIE bit in UCR is set, the UART Receive Complete interrupt will be executed when RXC is set (one). RXC is cleared by reading UDR. When interrupt-driven data reception is used, the UART Receive Complete Interrupt routine must read UDR in order to clear RXC, otherwise a new interrupt will occur once the interrupt routine terminates.

- **Bit 6 – TXC: UART Transmit Complete**

This bit is set (one) when the entire character (including the stop bit) in the Transmit Shift register has been shifted out and no new data has been written to UDR. This flag is especially useful in half-duplex communications interfaces, where a transmitting application must enter receive mode and free the communications bus immediately after completing the transmission.

When the TXCIE bit in UCR is set, setting of TXC causes the UART Transmit Complete interrupt to be executed. TXC is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, the TXC bit is cleared (zero) by writing a logical “1” to the bit.

- **Bit 5 – UDRE: UART Data Register Empty**

This bit is set (one) when a character written to UDR is transferred to the Transmit Shift register. Setting of this bit indicates that the transmitter is ready to receive a new character for transmission.

When the UDRIE bit in UCR is set, the UART Transmit Complete interrupt to be executed as long as UDRE is set. UDRE is cleared by writing UDR. When interrupt-driven data transmittal is used, the UART Data Register Empty Interrupt routine must write UDR in order to clear UDRE, otherwise a new interrupt will occur once the interrupt routine terminates.

UDRE is set (one) during reset to indicate that the transmitter is ready.

- **Bit 4 – FE: Framing Error**

This bit is set if a Framing Error condition is detected, i.e., when the stop bit of an incoming character is zero.

The FE bit is cleared when the stop bit of received data is one.

- **Bit 3 – OR: Overrun**

This bit is set if an Overrun condition is detected, i.e., when a character already present in the UDR register is not read before the next character has been shifted into the Receiver Shift register. The OR bit is buffered, which means that it will be set once the valid data still in UDRE is read.

The OR bit is cleared (zero) when data is received and transferred to UDR.

- **Bits 2..0 – Res: Reserved Bits**

These bits are reserved bits in the AT90S8515 and will always read as zero.

### UART Control Register – UCR

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---|---|---|---|---|---|---|---|---|---|
| $0A ($2A) | RXCIE | TXCIE | UDRIE | RXEN | TXEN | CHR9 | RXB8 | TXB8 | UCR |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R | W | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | |

- **Bit 7 – RXCIE: RX Complete Interrupt Enable**

When this bit is set (one), a setting of the RXC bit in USR will cause the Receive Complete Interrupt routine to be executed provided that global interrupts are enabled.

- **Bit 6 – TXCIE: TX Complete Interrupt Enable**

When this bit is set (one), a setting of the TXC bit in USR will cause the Transmit Complete Interrupt routine to be executed provided that global interrupts are enabled.

- **Bit 5 – UDRIE: UART Data Register Empty Interrupt Enable**

When this bit is set (one), a setting of the UDRE bit in USR will cause the UART Data Register Empty Interrupt routine to be executed provided that global interrupts are enabled.

- **Bit 4 – RXEN: Receiver Enable**

This bit enables the UART receiver when set (one). When the receiver is disabled, the RXC, OR and FE status flags cannot become set. If these flags are set, turning off RXEN does not cause them to be cleared.

- **Bit 3 – TXEN: Transmitter Enable**

This bit enables the UART transmitter when set (one). When disabling the transmitter while transmitting a character, the transmitter is not disabled before the character in the shift register plus any following character in UDR has been completely transmitted.

- **Bit 2 – CHR9: 9-bit Characters**

When this bit is set (one) transmitted and received characters are 9 bits long plus start and stop bits. The ninth bit is read and written by using the RXB8 and TXB8 bits in UCR, respectively. The ninth data bit can be used as an extra stop bit or a parity bit.

- **Bit 1 – RXB8: Receive Data Bit 8**

When CHR9 is set (one), RXB8 is the ninth data bit of the received character.

- **Bit 0 – TXB8: Transmit Data Bit 8**

When CHR9 is set (one), TXB8 is the ninth data bit in the character to be transmitted.

**BAUD Rate Generator**

The baud rate generator is a frequency divider that generates baud rates according to the following equation:

$$\text{BAUD} = \frac{f_{CK}}{16(\text{UBRR} + 1)}$$

- BAUD = Baud rate
- $f_{CK}$ = Crystal Clock frequency
- UBRR = Contents of the UART Baud Rate register, UBRR (0 - 255)

For standard crystal frequencies, the most commonly used baud rates can be generated by using the UBRR settings in Table 17. UBRR values that yield an actual baud rate differing less than 2% from the target baud rate are boldface in the table. However, using baud rates that have more than 1% error is not recommended. High error ratings give less noise immunity.

**Table 17.** UBRR Settings at Various Crystal Frequencies

| Baud Rate | 1 MHz | | %Error | 1.8432 MHz | | %Error | 2 MHz | | %Error | 2.4576 MHz | | %Error |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **2400** | UBRR= | **25** | **0.2** | UBRR= | **47** | **0.0** | UBRR= | **51** | **0.2** | UBRR= | **63** | **0.0** |
| **4800** | UBRR= | **12** | **0.2** | UBRR= | **23** | **0.0** | UBRR= | **25** | **0.2** | UBRR= | **31** | **0.0** |
| **9600** | UBRR= | 6 | 7.5 | UBRR= | **11** | **0.0** | UBRR= | **12** | **0.2** | UBRR= | **15** | **0.0** |
| **14400** | UBRR= | 3 | 7.8 | UBRR= | **7** | **0.0** | UBRR= | 8 | 3.7 | UBRR= | 10 | 3.1 |
| **19200** | UBRR= | 2 | 7.8 | UBRR= | **5** | **0.0** | UBRR= | 6 | 7.5 | UBRR= | **7** | **0.0** |
| **28800** | UBRR= | 1 | 7.8 | UBRR= | **3** | **0.0** | UBRR= | 3 | 7.8 | UBRR= | 4 | 6.3 |
| **38400** | UBRR= | 1 | 22.9 | UBRR= | **2** | **0.0** | UBRR= | 2 | 7.8 | UBRR= | **3** | **0.0** |
| **57600** | UBRR= | 0 | 7.8 | UBRR= | **1** | **0.0** | UBRR= | 1 | 7.8 | UBRR= | 2 | 12.5 |
| **76800** | UBRR= | 0 | 22.9 | UBRR= | 1 | 33.3 | UBRR= | 1 | 22.9 | UBRR= | **1** | **0.0** |
| **115200** | UBRR= | 0 | 84.3 | UBRR= | **0** | **0.0** | UBRR= | 0 | 7.8 | UBRR= | 0 | 25.0 |

| Baud Rate | 3.2768 MHz | | %Error | 3.6864 MHz | | %Error | 4 MHz | | %Error | 4.608 MHz | | %Error |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **2400** | UBRR= | **84** | **0.4** | UBRR= | **95** | **0.0** | UBRR= | **103** | **0.2** | UBRR= | **119** | **0.0** |
| **4800** | UBRR= | **42** | **0.8** | UBRR= | **47** | **0.0** | UBRR= | **51** | **0.2** | UBRR= | **59** | **0.0** |
| **9600** | UBRR= | **20** | **1.6** | UBRR= | **23** | **0.0** | UBRR= | **25** | **0.2** | UBRR= | **29** | **0.0** |
| **14400** | UBRR= | **13** | **1.6** | UBRR= | **15** | **0.0** | UBRR= | 16 | 2.1 | UBRR= | **19** | **0.0** |
| **19200** | UBRR= | 10 | 3.1 | UBRR= | **11** | **0.0** | UBRR= | **12** | **0.2** | UBRR= | **14** | **0.0** |
| **28800** | UBRR= | **6** | **1.6** | UBRR= | **7** | **0.0** | UBRR= | 8 | 3.7 | UBRR= | **9** | **0.0** |
| **38400** | UBRR= | 4 | 6.3 | UBRR= | **5** | **0.0** | UBRR= | 6 | 7.5 | UBRR= | 7 | 6.7 |
| **57600** | UBRR= | 3 | 12.5 | UBRR= | **3** | **0.0** | UBRR= | 3 | 7.8 | UBRR= | **4** | **0.0** |
| **76800** | UBRR= | 2 | 12.5 | UBRR= | **2** | **0.0** | UBRR= | 2 | 7.8 | UBRR= | 3 | 6.7 |
| **115200** | UBRR= | 1 | 12.5 | UBRR= | **1** | **0.0** | UBRR= | 1 | 7.8 | UBRR= | 2 | 20.0 |

| Baud Rate | 7.3728 MHz | | %Error | 8 MHz | | %Error | 9.216 MHz | | %Error | 11.059 MHz | | %Error |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **2400** | UBRR= | **191** | **0.0** | UBRR= | **207** | **0.2** | UBRR= | **239** | **0.0** | UBRR= | 287 | - |
| **4800** | UBRR= | **95** | **0.0** | UBRR= | **103** | **0.2** | UBRR= | **119** | **0.0** | UBRR= | **143** | **0.0** |
| **9600** | UBRR= | **47** | **0.0** | UBRR= | **51** | **0.2** | UBRR= | **59** | **0.0** | UBRR= | **71** | **0.0** |
| **14400** | UBRR= | **31** | **0.0** | UBRR= | **34** | **0.8** | UBRR= | **39** | **0.0** | UBRR= | **47** | **0.0** |
| **19200** | UBRR= | **23** | **0.0** | UBRR= | **25** | **0.2** | UBRR= | **29** | **0.0** | UBRR= | **35** | **0.0** |
| **28800** | UBRR= | **15** | **0.0** | UBRR= | 16 | 2.1 | UBRR= | **19** | **0.0** | UBRR= | **23** | **0.0** |
| **38400** | UBRR= | **11** | **0.0** | UBRR= | **12** | **0.2** | UBRR= | **14** | **0.0** | UBRR= | **17** | **0.0** |
| **57600** | UBRR= | **7** | **0.0** | UBRR= | 8 | 3.7 | UBRR= | **9** | **0.0** | UBRR= | **11** | **0.0** |
| **76800** | UBRR= | **5** | **0.0** | UBRR= | 6 | 7.5 | UBRR= | 7 | 6.7 | UBRR= | **8** | **0.0** |
| **115200** | UBRR= | **3** | **0.0** | UBRR= | 3 | 7.8 | UBRR= | **4** | **0.0** | UBRR= | **5** | **0.0** |

**UART BAUD Rate Register – UBRR**

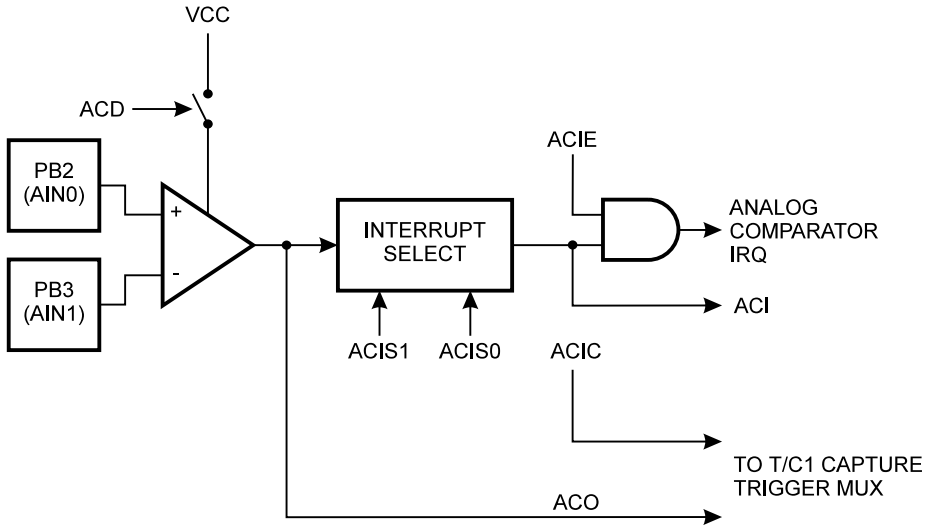
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---|---|---|---|---|---|---|---|---|---|
| $09 ($29) | **MSB** | | | | | | | **LSB** | **UBRR** |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |

The UBRR register is an 8-bit read/write register that specifies the UART Baud Rate according to the equation on the previous page.

## Analog Comparator

The Analog Comparator compares the input values on the positive input PB2 (AIN0) and negative input PB3 (AIN1). When the voltage on the positive input PB2 (AIN0) is higher than the voltage on the negative input PB3 (AIN1), the Analog Comparator Output (ACO) is set (one). The comparator's output can be set to trigger the Timer/Counter1 Input Capture function. In addition, 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 41.

**Figure 41.** Analog Comparator Block Diagram




### Analog Comparator Control and Status Register – ACSR

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---|---|---|---|---|---|---|---|---|---|
| $08 ($28) | ACD | – | ACO | ACI | ACIE | ACIC | ACIS1 | ACIS0 | ACSR |
| Read/Write | R/W | R | R | R/W | R/W | R/W | R/W | R/W | |
| Initial Value | 0 | 0 | N/A | 0 | 0 | 0 | 0 | 0 | |

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

When this bit is 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. This will reduce power consumption in active and idle mode. When changing the ACD bit, the Analog Comparator interrupt must be disabled by clearing the ACIE bit in ACSR. Otherwise an interrupt can occur when the bit is changed.

- **Bit 6 – Res: Reserved Bit**

This bit is a reserved bit in the AT90S8515 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 logical "1" to the flag. Observe however, that if another bit in this register is modified

using the SBI or CBI instruction, ACI will be cleared if it has become set before the operation.

- **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 – ACIC: Analog Comparator Input Capture Enable**

When set (one), this bit enables the Input Capture function in Timer/Counter1 to be triggered by the Analog Comparator. The comparator output is, in this case, directly connected to the Input Capture front-end logic, making the comparator utilize the noise canceler and edge select features of the Timer/Counter1 Input Capture interrupt. When cleared (zero), no connection between the Analog Comparator and the Input Capture function is given. To make the comparator trigger the Timer/Counter1 Input Capture interrupt, the TICIE1 bit in the Timer Interrupt Mask Register (TIMSK) must be set (one).

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

These bits determine which comparator events trigger the Analog Comparator interrupt. The different settings are shown in Table 18.

**Table 18.** 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.

## Interface to External SRAM

The interface to the SRAM consists of:

Port A: Multiplexed low-order address bus and data bus

Port C: High-order address bus

The ALE pin: Address latch enable

The $\overline{RD}$ and $\overline{WR}$ pins: Read and write strobes

The external data SRAM is enabled by setting the SRE (external SRAM enable) bit of the MCUCR (MCU Control Register) and will override the setting of the Data Direction Register (DDRA). When the SRE bit is cleared (zero), the external data SRAM is disabled and the normal pin and data direction settings are used. When SRE is cleared (zero), the address space above the internal SRAM boundary is not mapped into the internal SRAM, as AVR parts do not have an interface to the external SRAM.

When ALE goes from high to low, there is a valid address on Port A. ALE is low during a data transfer. $\overline{RD}$ and $\overline{WR}$ are active when accessing the external SRAM only.

When the external SRAM is enabled, the ALE signal may have short pulses when accessing the internal RAM, but the ALE signal is stable when accessing the external SRAM.

Figure 42 sketches how to connect an external SRAM to the AVR using eight latches that are transparent when G is high.

Default, the external SRAM access, is a 3-cycle scheme as depicted in Figure 43. When one extra wait state is needed in the access cycle, set the SRW bit (one) in the MCUCR register. The resulting access scheme is shown in Figure 44. In both cases, note that PORTA is data bus in one cycle only. As soon as the data access finishes, PORTA becomes a low-order address bus again.

For details of the timing for the SRAM interface, please refer to Figure 68, Table 37, Table 38, Table 39 and Table 40, beginning on page 92. Refer to “Architectural Overview” on page 7 for a description of the memory map, including address space for SRAM.

**Figure 42.** External SRAM Connected to the AVR

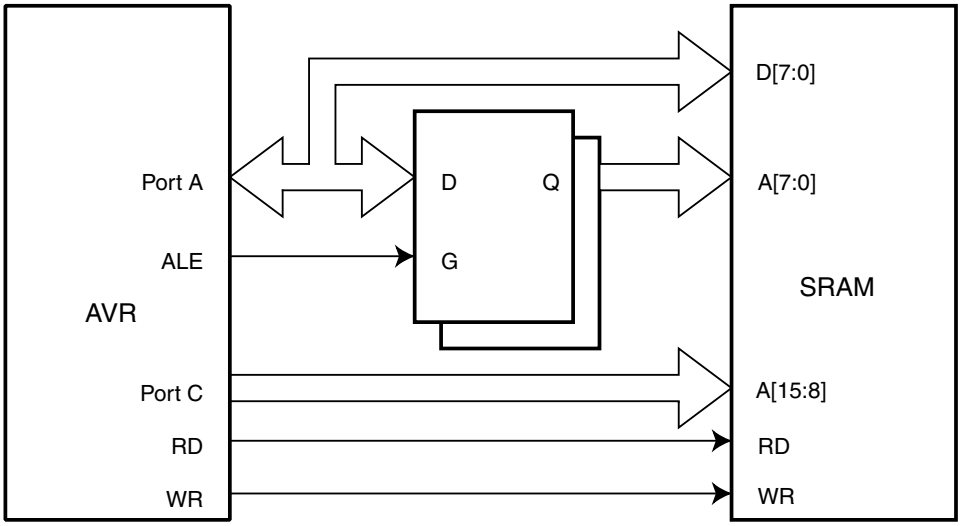



**Figure 43.** External Data SRAM Memory Cycles without Wait State

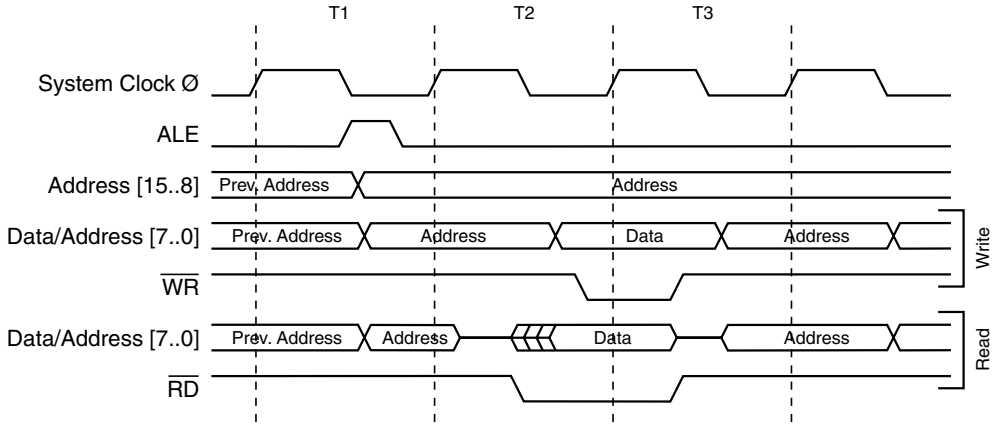
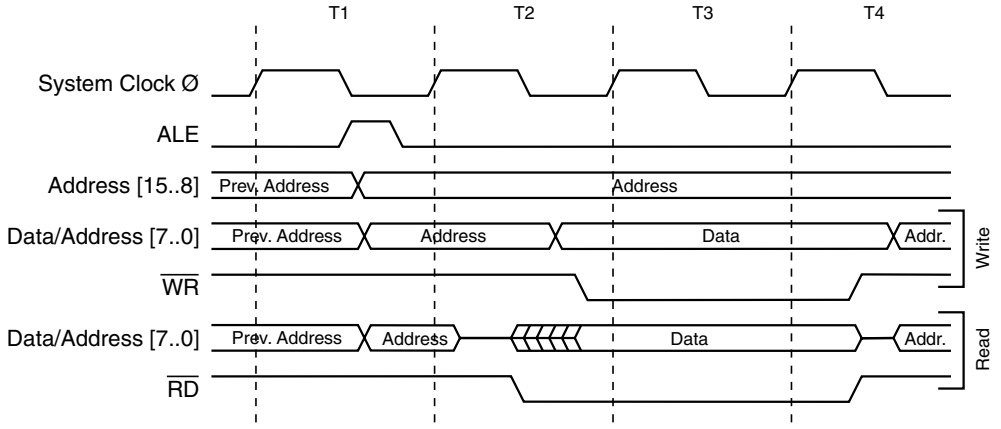


**Figure 44.** External Data SRAM Memory Cycles with Wait State



## 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 the enabling/disabling of pull-up resistors (if configured as input).

## Port A

Port A is an 8-bit bi-directional I/O port.

Three I/O memory address locations are allocated for the Port A, one each for the Data Register – PORTA, $1B($3B), Data Direction Register – DDRA, $1A($3A) and the Port A Input Pins – PINA, $19($39). The Port A Input Pins address is read-only, while the Data Register and the Data Direction Register are read/write.

All port pins have individually selectable pull-up resistors. The Port A output buffers can sink 20 mA and thus drive LED displays directly. When pins PA0 to PA7 are used as inputs and are externally pulled low, they will source current if the internal pull-up resistors are activated.

The Port A pins have alternate functions related to the optional external data SRAM. Port A can be configured to be the multiplexed low-order address/data bus during accesses to the external data memory. In this mode, Port A has internal pull-up resistors.

When Port A is set to the alternate function by the SRE (external SRAM enable) bit in the MCUCR (MCU Control Register), the alternate settings override the Data Direction Register.

### Port A Data Register – PORTA

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---|---|---|---|---|---|---|---|---|---|
| $1B ($3B) | PORTA7 | PORTA6 | PORTA5 | PORTA4 | PORTA3 | PORTA2 | PORTA1 | PORTA0 | PORTA |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |

### Port A Data Direction Register – DDRA

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---|---|---|---|---|---|---|---|---|---|
| $1A ($3A) | DDA7 | DDA6 | DDA5 | DDA4 | DDA3 | DDA2 | DDA1 | DDA0 | DDRA |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |

### Port A Input Pins Address – PINA

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---|---|---|---|---|---|---|---|---|---|
| $19 ($39) | PINA7 | PINA6 | PINA5 | PINA4 | PINA3 | PINA2 | PINA1 | PINA0 | PINA |
| 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 A Input Pins address (PINA) is not a register; 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.

### Port A as General Digital I/O

All eight pins in Port A have equal functionality when used as digital I/O pins.

PAn, general I/O pin: The DDAn bit in the DDRA register selects the direction of this pin. 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 has to be cleared (zero) or the pin has to be configured as an output pin. The Port A pins are tri-stated when a reset condition becomes active, even if the clock is not active..

**Table 19.** DDAn Effects on Port A Pins

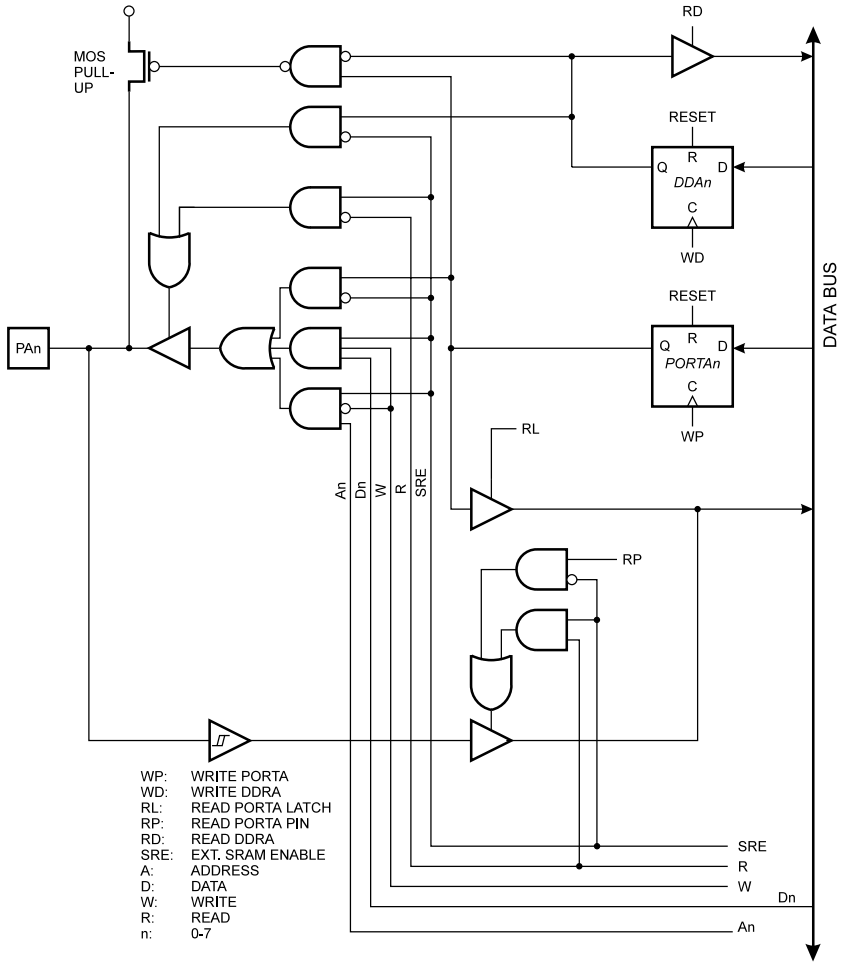
| DDAn | PORTAn | I/O | Pull-up | Comment |
|---|---|---|---|---|
| 0 | 0 | Input | No | Tri-state (high-Z) |
| 0 | 1 | Input | Yes | PAn will source current if ext. pulled low. |
| 1 | 0 | Output | No | Push-pull Zero Output |
| 1 | 1 | Output | No | Push-pull One Output |

Note: n: 7,6…0, pin number.

## Port A Schematics

Note that all port pins are synchronized. The synchronization latch is, however, not shown in the figure.

**Figure 45.** Port A Schematic Diagrams (Pins PA0 - PA7)



## Port B

Port B is an 8-bit bi-directional I/O port.

Three I/O memory address locations are allocated for the Port B, one each for the Data Register – PORTB, $18($38), Data Direction Register – DDRB, $17($37) and the Port B Input Pins – PINB, $16($36). The Port B Input Pins address is read-only, while the Data Register and the Data Direction Register are read/write.

All port pins have individually selectable pull-up resistors. The Port B output buffers can sink 20 mA and thus drive LED displays directly. When pins PB0 to PB7 are used as inputs and are externally pulled low, they will source current if the internal pull-up resistors are activated.

The Port B pins with alternate functions are shown in Table 20.

**Table 20.** Port B Pin Alternate Functions

| Port Pin | Alternate Functions |
|---|---|
| PB0 | T0 (Timer/Counter 0 External Counter Input) |
| PB1 | T1 (Timer/Counter 1 External Counter Input) |
| PB2 | AIN0 (Analog Comparator positive input) |
| PB3 | AIN1 (Analog Comparator negative input) |
| PB4 | $\overline{SS}$ (SPI Slave Select Input) |
| PB5 | MOSI (SPI Bus Master Output/Slave Input) |
| PB6 | MISO (SPI Bus Master Input/Slave Output) |
| PB7 | SCK (SPI Bus Serial Clock) |

When the pins are used for the alternate function, the DDRB and PORTB registers have to be set according to the alternate function description.

### Port B Data Register – PORTB

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---|---|---|---|---|---|---|---|---|---|
| $18 ($38) | PORTB7 | PORTB6 | PORTB5 | PORTB4 | PORTB3 | PORTB2 | PORTB1 | PORTB0 | PORTB |
| 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 B Data Direction Register – DDRB

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---|---|---|---|---|---|---|---|---|---|
| $17 ($37) | DDB7 | DDB6 | DDB5 | DDB4 | DDB3 | DDB2 | DDB1 | DDB0 | DDRB |
| 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 B Input Pins Address – PINB

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---|---|---|---|---|---|---|---|---|---|
| $16 ($36) | PINB7 | PINB6 | PINB5 | PINB4 | PINB3 | PINB2 | PINB1 | PINB0 | 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; this address enables access to the physical value on each Port B pin. When reading PORTB, the Port B Data Latch is read and when reading PINB, the logical values present on the pins are read.

### Port B as General Digital I/O

All eight pins in Port B have equal functionality when used as digital I/O pins.

PBn, general I/O pin: The DDBn bit in the DDRB register selects the direction of this pin. If DDBn is set (one), PBn is configured as an output pin. If DDBn is cleared (zero), PBn is configured as an input pin. If PORTBn 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 PORTBn has to be cleared (zero) or the pin has to be configured as an output pin. The Port B pins are tri-stated when a reset condition becomes active, even if the clock is not active.

**Table 21.** DDBn Effects on Port B Pins

| DDBn | PORTBn | I/O | Pull up | Comment |
|---|---|---|---|---|
| 0 | 0 | Input | No | Tri-state (high-Z) |
| 0 | 1 | Input | Yes | PBn will source current if ext. pulled low. |
| 1 | 0 | Output | No | Push-pull Zero Output |
| 1 | 1 | Output | No | Push-pull One Output |

Note: n: 7,6…0, pin number.

### Alternate Functions of Port B

The alternate pin configuration is as follows:

- **SCK – Port B, Bit 7**

SCK: Master clock output, slave clock input pin for SPI channel. When the SPI is enabled as a slave, this pin is configured as an input regardless of the setting of DDB7. When the SPI is enabled as a master, the data direction of this pin is controlled by DDB7. When the pin is forced to be an input, the pull-up can still be controlled by the PORTB7 bit. See the description of the SPI port for further details.

- **MISO – Port B, Bit 6**

MISO: Master data input, slave data output pin for SPI channel. When the SPI is enabled as a master, this pin is configured as an input regardless of the setting of DDB6. When the SPI is enabled as a slave, the data direction of this pin is controlled by DDB6. When the pin is forced to be an input, the pull-up can still be controlled by the PORTB6 bit. See the description of the SPI port for further details.

- **MOSI – Port B, Bit 5**

MOSI: SPI Master data output, slave data input for SPI channel. When the SPI is enabled as a slave, this pin is configured as an input regardless of the setting of DDB5. When the SPI is enabled as a master, the data direction of this pin is controlled by DDB5. When the pin is forced to be an input, the pull-up can still be controlled by the PORTB5 bit. See the description of the SPI port for further details.

- **SS – Port B, Bit 4**

$\overline{SS}$: Slave port select input. When the SPI is enabled as a slave, this pin is configured as an input regardless of the setting of DDB4. As a slave, the SPI is activated when this pin is driven low. When the SPI is enabled as a master, the data direction of this pin is controlled by DDB4. When the pin is forced to be an input, the pull-up can still be controlled by the PORTB4 bit. See the description of the SPI port for further details.

- **AIN1 – Port B, Bit 3**

AIN1: Analog Comparator Negative Input. When configured as an input (DDB3 is cleared [zero]) and with the internal MOS pull-up resistor switched off (PB3 is cleared [zero]), this pin also serves as the negative input of the On-chip Analog Comparator.

- **AIN0 – Port B, Bit 2**

AIN0: Analog Comparator Positive Input. When configured as an input (DDB2 is cleared [zero]) and with the internal MOS pull-up resistor switched off (PB2 is cleared [zero]), this pin also serves as the positive input of the On-chip Analog Comparator.

- **T1 – Port B, Bit 1**

T1: Timer/Counter1 counter source. See the timer description for further details

- **T0 – Port B, Bit 0**

T0: Timer/Counter0 counter source. See the timer description for further details.

**Port B Schematics**

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

**Figure 46.** Port B Schematic Diagram (Pins PB0 and PB1)

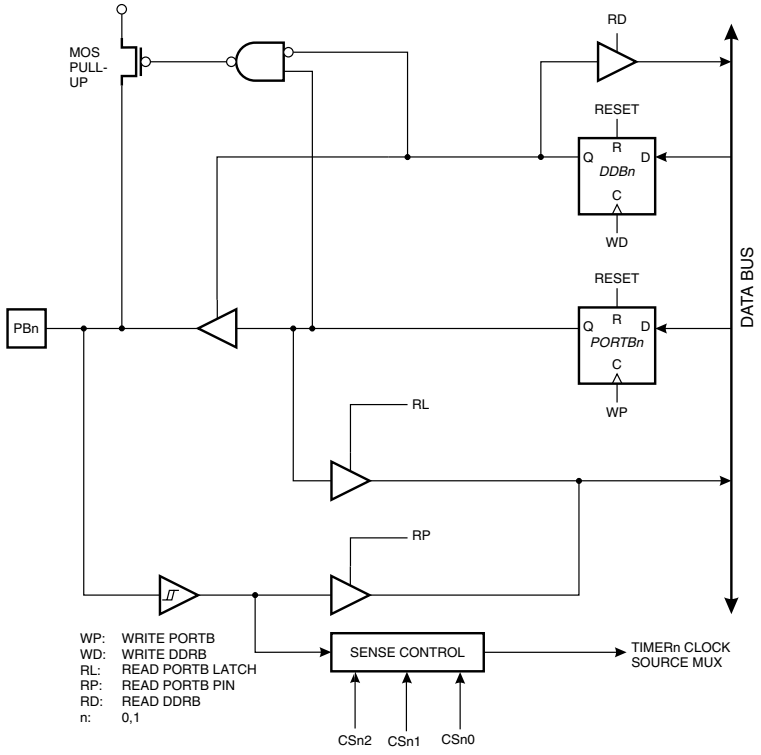


**Figure 47.** Port B Schematic Diagram (Pins PB2 and PB3)

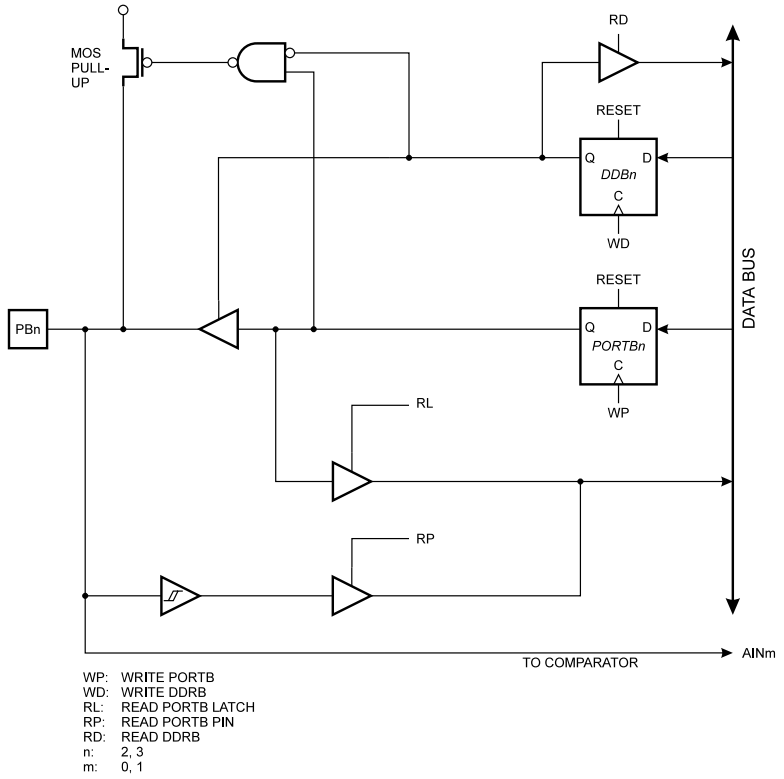



**Figure 48.** Port B Schematic Diagram (Pin PB4)

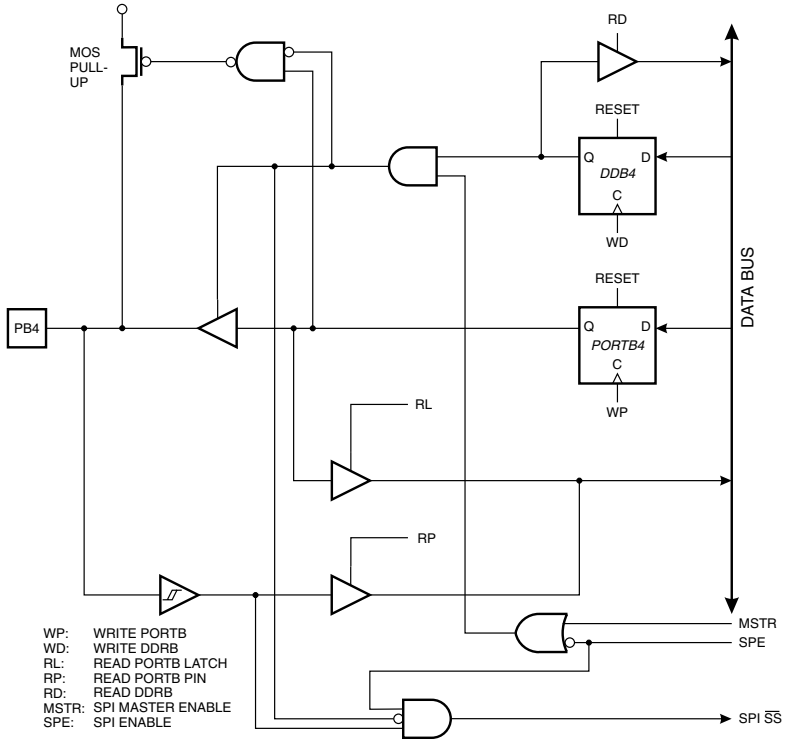


**Figure 49.** Port B Schematic Diagram (Pin PB5)

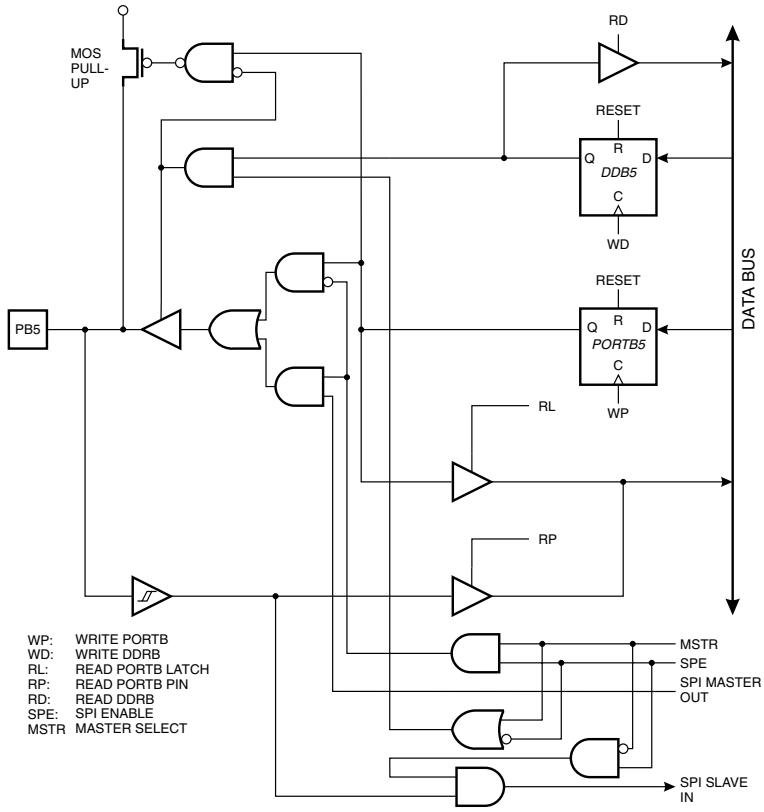



**Figure 50.** Port B Schematic Diagram (Pin PB6)

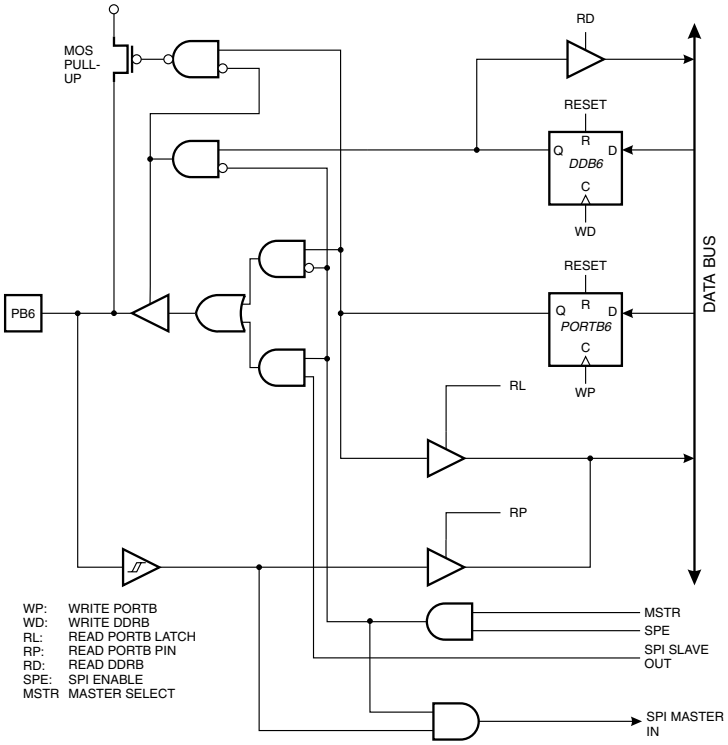
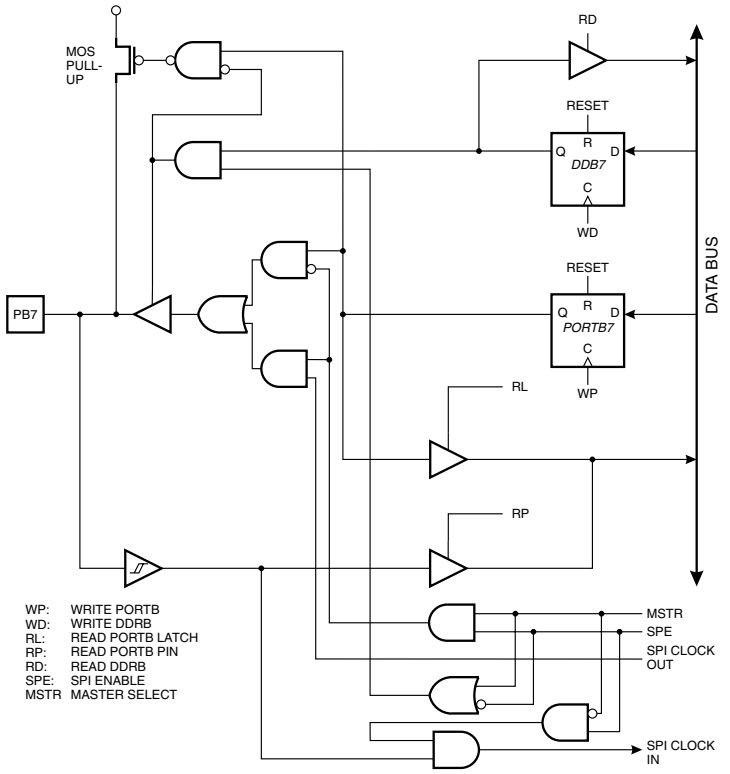


**Figure 51.** Port B Schematic Diagram (Pin PB7)




## Port C

Port C is an 8-bit bi-directional I/O port. Three I/O memory address locations are allocated for the Port C, one each for the Data Register – PORTC, $15($35), Data Direction Register – DDRC, $14($34) and the Port C Input Pins – PINC, $13($33). The Port C Input Pins address is read-only, while the Data Register and the Data Direction Register are read/write.

All port pins have individually selectable pull-up resistors. The Port C output buffers can sink 20 mA and thus drive LED displays directly. When pins PC0 to PC7 are used as inputs and are externally pulled low, they will source current if the internal pull-up resistors are activated.

The Port C pins have alternate functions related to the optional external data SRAM. Port C can be configured to be the high-order address byte during accesses to external data memory. When Port C is set to the alternate function by the SRE (external SRAM enable) bit in the MCUCR (MCU Control Register), the alternate settings override the Data Direction Register.

### Port C Data Register – PORTC

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---|---|---|---|---|---|---|---|---|---|
| $15 ($35) | PORTC7 | PORTC6 | PORTC5 | PORTC4 | PORTC3 | PORTC2 | PORTC1 | PORTC0 | PORTC |
| 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 C Data Direction Register – DDRC

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---|---|---|---|---|---|---|---|---|---|
| $14 ($34) | DDC7 | DDC6 | DDC5 | DDC4 | DDC3 | DDC2 | DDC1 | DDC0 | DDRC |
| 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 C Input Pins Address – PINC

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---|---|---|---|---|---|---|---|---|---|
| $13 ($33) | PINC7 | PINC6 | PINC5 | PINC4 | PINC3 | PINC2 | PINC1 | PINC0 | PINC |
| 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 C Input Pins address (PINC) is not a register; this address enables access to the physical value on each Port C pin. When reading PORTC, the Port C Data Latch is read and when reading PINC, the logical values present on the pins are read.

### Port C as General Digital I/O

All eight pins in Port C have equal functionality when used as digital I/O pins.

PCn, general I/O pin: The DDCn bit in the DDRC register selects the direction of this pin. If DDCn is set (one), PCn is configured as an output pin. If DDCn is cleared (zero), PCn is configured as an input pin. If PORTCn 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, PORTCn has to be cleared (zero) or the pin has to be configured as an output pin. The Port C pins are tri-stated when a reset condition becomes active, even if the clock is not active.

**Table 22.** DDCn Effects on Port C Pins

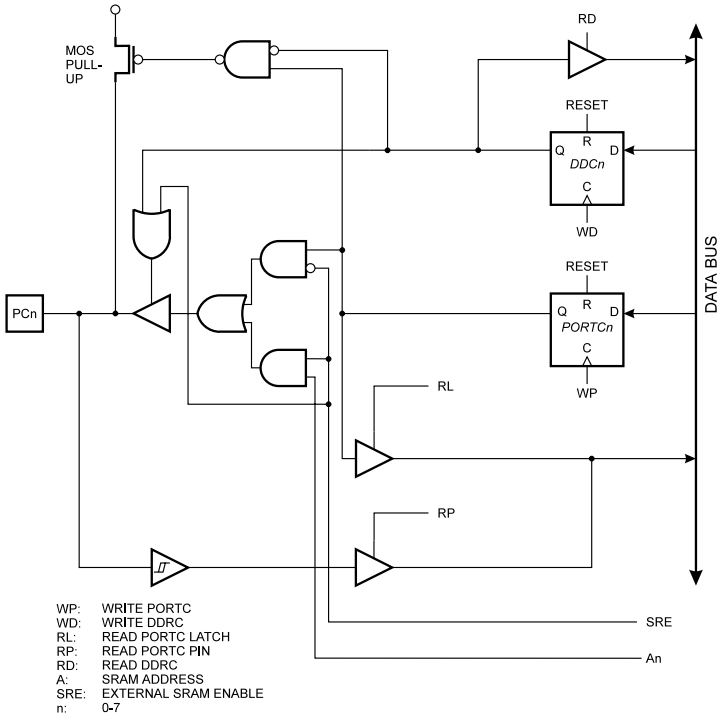
| DDCn | PORTCn | I/O | Pull-up | Comment |
|---|---|---|---|---|
| 0 | 0 | Input | No | Tri-state (high-Z) |
| 0 | 1 | Input | Yes | PCn will source current if ext. pulled low. |
| 1 | 0 | Output | No | Push-pull Zero Output |
| 1 | 1 | Output | No | Push-pull One Output |

Note: n: 7…0, pin number

**Port C Schematics**

Note that all port pins are synchronized. The synchronization latch is, however, not shown in the figure.

**Figure 52.** Port C Schematic Diagram (Pins PC0 - PC7)




## 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 the Port D, one each for the Data Register – PORTD, $12($32), Data Direction Register – DDRD, $11($31) and the Port D Input Pins – PIND, $10($30). 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 20 mA. As inputs, Port D pins that are externally pulled low will source current if the pull-up resistors are activated.

Some Port D pins have alternate functions as shown in Table 23.

**Table 23.** Port D Pin Alternate Functions

| Port Pin | Alternate Function |
|---|---|
| PD0 | RXD (UART Input Line) |
| PD1 | TXD (UART Output Line) |
| PD2 | INT0 (External interrupt 0 Input) |
| PD3 | INT1 (External interrupt 1 Input) |
| PD5 | OC1A (Timer/Counter1 Output CompareA Match Output) |
| PD6 | $\overline{WR}$ (Write Strobe to External Memory) |
| PD7 | $\overline{RD}$ (Read Strobe to External Memory) |

When the pins are used for the alternate function, the DDRD and PORTD registers have to be set according to the alternate function description.

### Port D Data Register – PORTD

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---|---|---|---|---|---|---|---|---|---|
| $12 ($32) | PORTD7 | PORTD6 | PORTD5 | PORTD4 | PORTD3 | PORTD2 | PORTD1 | PORTD0 | PORTD |
| 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 ($31) | DDD7 | DDD6 | DDD5 | DDD4 | DDD3 | DDD2 | DDD1 | DDD0 | DDRD |
| 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 ($30) | PIND7 | PIND6 | PIND5 | PIND4 | PIND3 | PIND2 | PIND1 | PIND0 | PIND |
| 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; 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

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

**Table 24.** DDDn Bits on Port D Pins

| DDDn | PORTDn | I/O | Pull-up | Comment |
|---|---|---|---|---|
| 0 | 0 | Input | No | Tri-state (high-Z) |
| 0 | 1 | Input | Yes | PDn will source current if ext. pulled low. |
| 1 | 0 | Output | No | Push-pull Zero Output |
| 1 | 1 | Output | No | Push-pull One Output |

Note: n: 7,6…0, pin number.

### Alternate Functions of Port D

- **$\overline{RD}$ – Port D, Bit 7**

$\overline{RD}$ is the external data memory read control strobe. See “Interface to External SRAM” on page 60 for detailed information.

- **$\overline{WR}$ – Port D, Bit 6**

$\overline{WR}$ is the external data memory write control strobe. See “Interface to External SRAM” on page 60 for detailed information.

- **OC1A – Port D, Bit 5**

OC1A: Output compare match output. The PD5 pin can serve as an external output when the Timer/Counter1 compare matches. The PD5 pin has to be configured as an output (DDD5 set [one]) to serve this function. See the Timer/Counter1 description for further details and how to enable the output. The OC1A pin is also the output pin for the PWM mode timer function.

- **INT1 – Port D, Bit 3**

INT1: External Interrupt source 1. The PD3 pin can serve as an external interrupt source to the MCU. See the interrupt description for further details and how to enable the source.

- **INT0 – Port D, Bit 2**

INT0: External Interrupt source 0. The PD2 pin can serve as an external interrupt source to the MCU. See the interrupt description for further details and how to enable the source.

- **TXD – Port D, Bit 1**

Transmit Data (data output pin for the UART). When the UART transmitter is enabled, this pin is configured as an output, regardless of the value of DDRD1.

- **RXD – Port D, Bit 0**

Receive Data (data input pin for the UART). When the UART receiver is enabled, this pin is configured as an input, regardless of the value of DDRD0. When the UART forces this pin to be an input, a logical “1” in PORTD0 will turn on the internal pull-up.

**Port D Schematics**

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

**Figure 53.** Port D Schematic Diagram (Pin PD0)

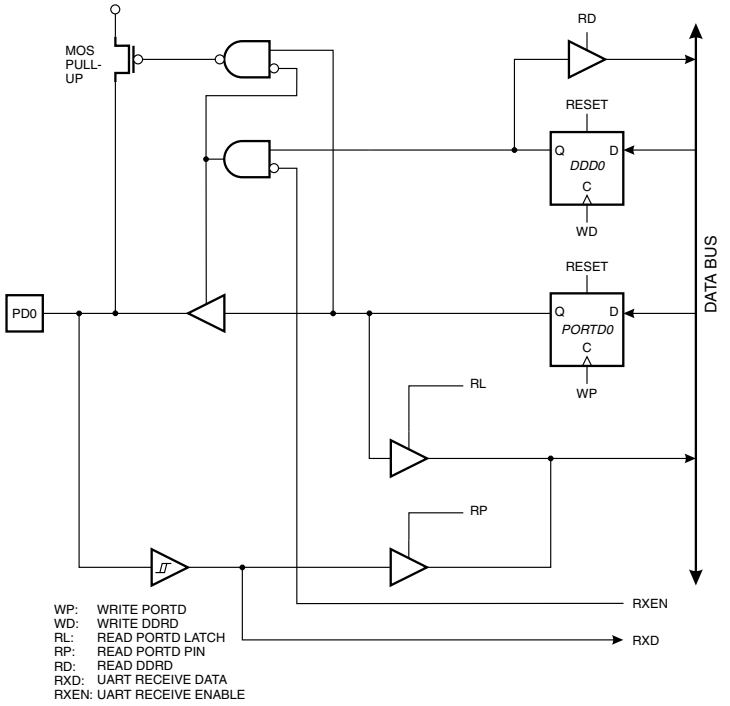


**Figure 54.** Port D Schematic Diagram (Pin PD1)

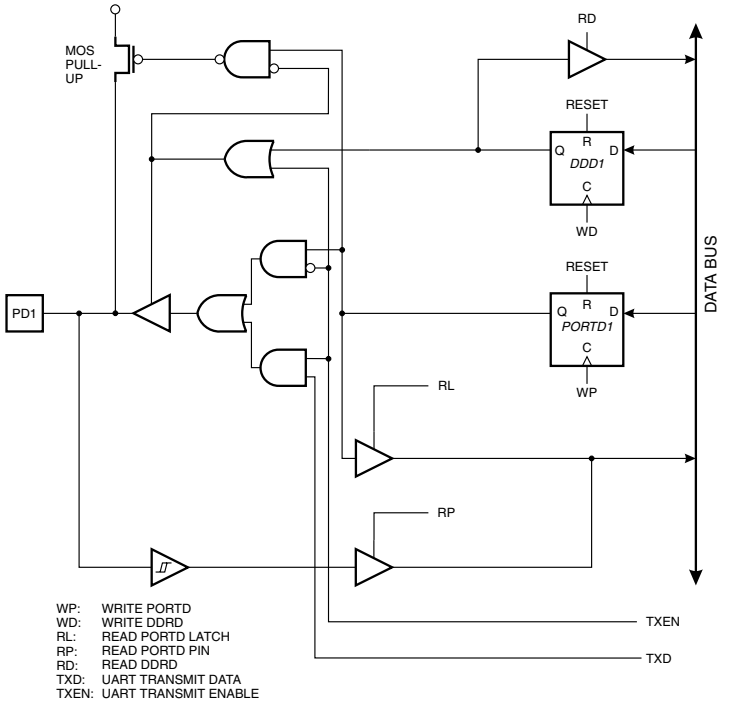



**Figure 55.** Port D Schematic Diagram (Pins PD2 and PD3)

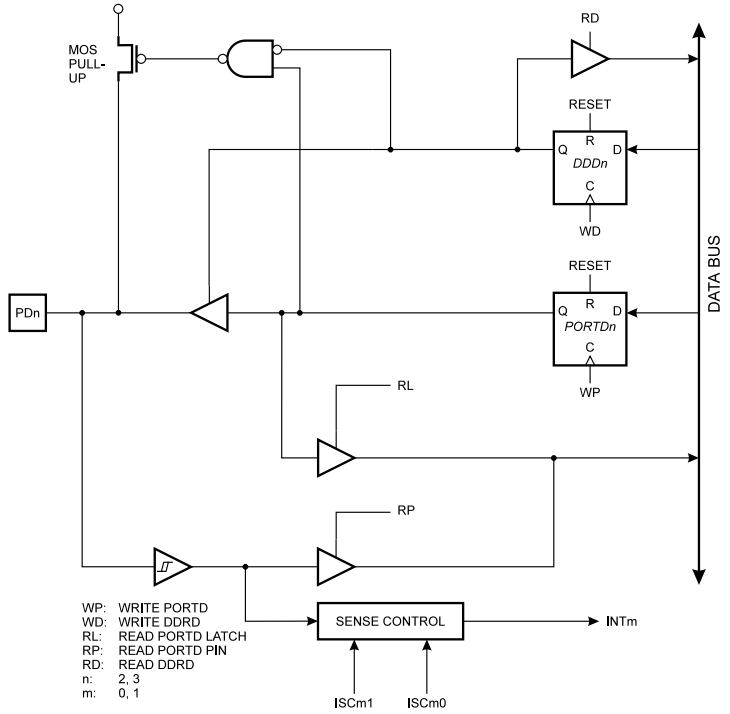


**Figure 56.** Port D Schematic Diagram (Pin PD4)

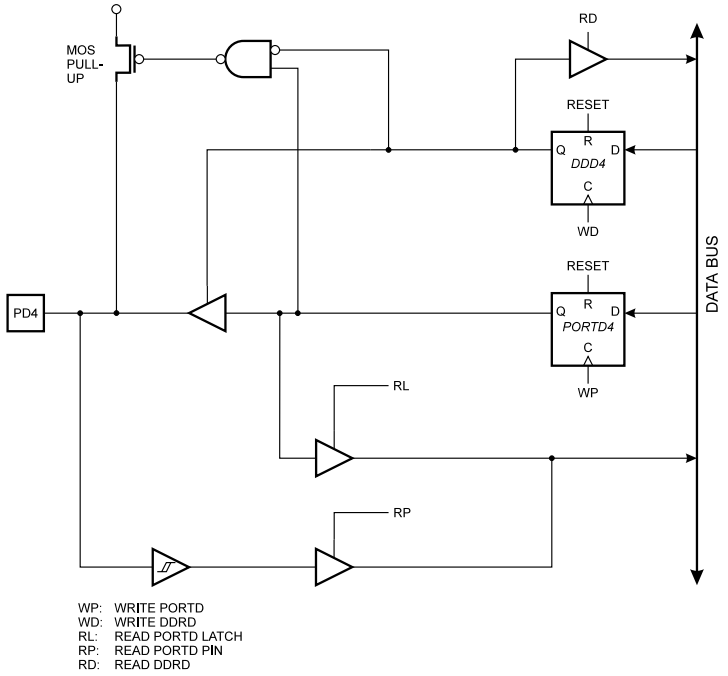



**Figure 57.** Port D Schematic Diagram (Pin PD5)

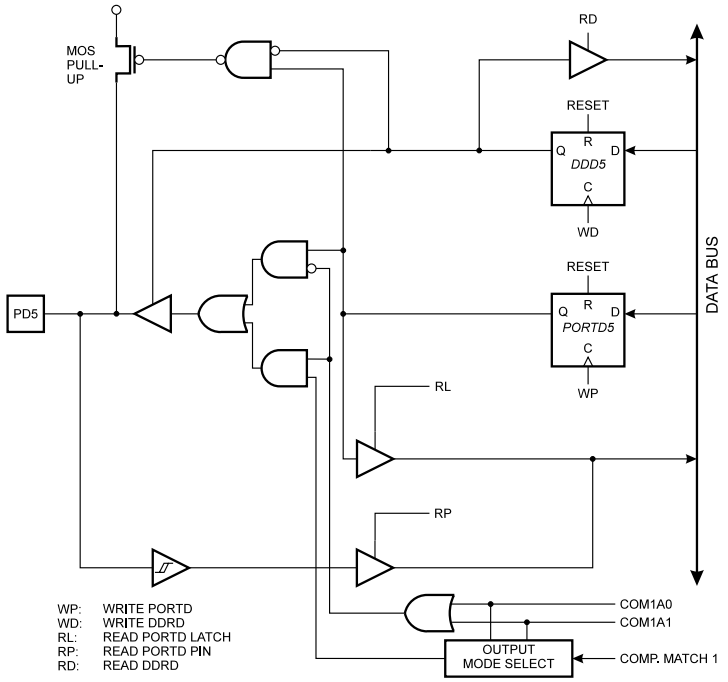


**Figure 58.** Port D Schematic Diagram (Pin PD6)

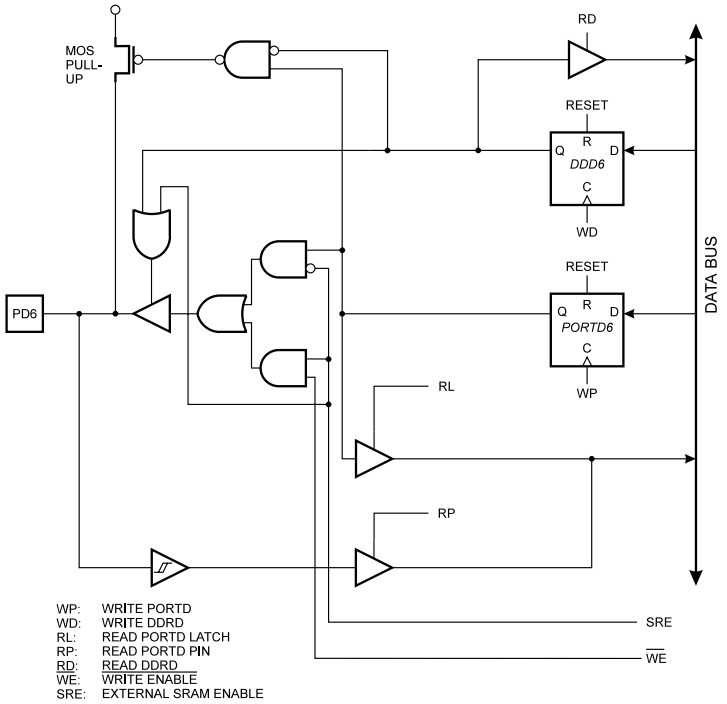
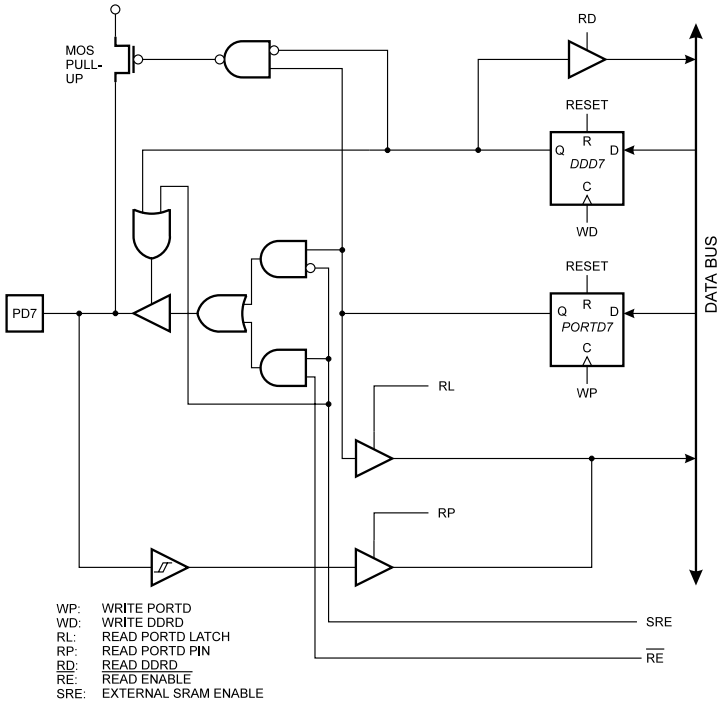



**Figure 59.** Port D Schematic Diagram (Pin PD7)



# Memory Programming

## Program and Data Memory Lock Bits

The AT90S8515 MCU provides two Lock bits that can be left unprogrammed (“1”) or can be programmed (“0”) to obtain the additional features listed in Table 25. The Lock bits can only be erased with the Chip Erase command.

**Table 25.** Lock Bit Protection Modes

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

Note: 1. In Parallel Mode, further programming of the Fuse bits is also disabled. Program the Fuse bits before programming the Lock bits.

## Fuse Bits

The AT90S8515 has two Fuse bits, SPIEN and FSTRT.

- When the SPIEN Fuse is programmed (“0”), Serial Program and Data Downloading is enabled. Default value is programmed (“0”).
- When the FSTRT Fuse is programmed (“0”), the short start-up time is selected. Default value is unprogrammed (“1”). Parts with this bit pre-programmed (“0”) can be delivered on demand.

The Fuse bits are not accessible in Serial Programming Mode. The status of the Fuse bits is not affected by Chip Erase.

## Signature Bytes

All Atmel microcontrollers have a three-byte signature code that identifies the device. This code can be read in both Serial and Parallel mode. The three bytes reside in a separate address space.

For the AT90S8515(1) they are:

1. $000: $1E (indicates manufactured by Atmel)
2. $001: $93 (indicates 8 KB Flash memory)
3. $002: $01 (indicates AT90S8515 device when signature byte $001 is $93)

Note: 1. When both Lock bits are programmed (lock mode 3), the signature bytes cannot be read in Serial Mode. Reading the signature bytes will return: $00, $01 and $02.

## Programming the Flash and EEPROM

Atmel’s AT90S8515 offers 8K bytes of In-System Reprogrammable Flash program memory and 512 bytes of EEPROM data memory.

The AT90S8515 is shipped with the On-chip Flash program and EEPROM data memory arrays in the erased state (i.e., contents = $FF) and ready to be programmed. This device supports a high-voltage (12V) Parallel Programming Mode and a low-voltage Serial Programming Mode. The +12V is used for programming enable only, and no current of significance is drawn by this pin. The Serial Programming Mode provides a convenient way to download program and data into the AT90S8515 inside the user’s system.

The program and data memory arrays on the AT90S8515 are programmed byte-by-byte in either programming mode. For the EEPROM, an auto-erase cycle is provided within

the self-timed write operation in the serial programming mode. During programming, the supply voltage must be in accordance with Table 26.

**Table 26.** Supply Voltage during Programming

| Part | Serial Programming | Parallel Programming |
|---|---|---|
| AT90S8515 | 2.7 - 6.0V | 4.5 - 5.5V |

## Parallel Programming

This section describes how to parallel program and verify Flash program memory, EEPROM data memory, Lock bits and Fuse bits in the AT90S8515.

### Signal Names

In this section, some pins of the AT908515 are referenced by signal names describing their function during parallel programming. See Figure 60 and Table 27. Pins not described in Table 27 are referenced by pin names.

The XA1/XA0 pins determine the action executed when the XTAL1 pin is given a positive pulse. The bit coding are shown in Table 28.

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 29.

**Figure 60.** Parallel Programming

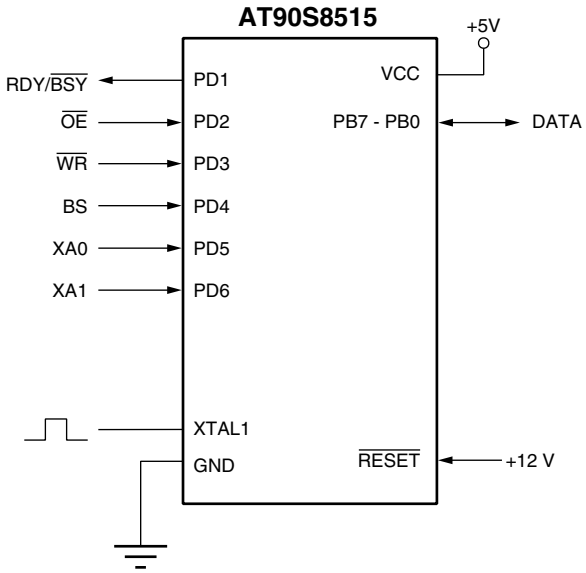


**Table 27.** Pin Name Mapping

| Signal Name in Programming Mode | Pin Name | I/O | Function |
|---|---|---|---|
| RDY/$\overline{BSY}$ | PD1 | O | 0: Device is busy programming, 1: Device is ready for new command |
| $\overline{OE}$ | PD2 | I | Output Enable (Active low) |
| $\overline{WR}$ | PD3 | I | Write Pulse (Active low) |
| BS | PD4 | I | Byte Select (“0” selects low byte, “1” selects high byte) |
| XA0 | PD5 | I | XTAL Action Bit 0 |
| XA1 | PD6 | I | XTAL Action Bit 1 |
| DATA | PB7-0 | I/O | Bi-directional Data Bus (Output when $\overline{OE}$ is low) |

**Table 28.** XA1 and XA0 Coding

| XA1 | XA0 | Action when XTAL1 is Pulsed |
|---|---|---|
| 0 | 0 | Load Flash or EEPROM Address (High or low address byte 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 29.** Command Byte Bit Coding

| Command Byte | Command Executed |
|---|---|
| 1000 0000 | Chip Erase |
| 0100 0000 | Write Fuse Bits |
| 0010 0000 | Write Lock Bits |
| 0001 0000 | Write Flash |
| 0001 0001 | Write EEPROM |
| 0000 1000 | Read Signature Bytes |
| 0000 0100 | Read Lock and Fuse Bits |
| 0000 0010 | Read Flash |
| 0000 0011 | Read EEPROM |

**Enter Programming Mode**

The following algorithm puts the device in Parallel Programming Mode:

1. Apply supply voltage according to Table 26, between $V_{CC}$ and GND.
2. Set the $\overline{RESET}$ and BS pin to “0” and wait at least 100 ns.
3. Apply 11.5 - 12.5V to $\overline{RESET}$. Any activity on BS within 100 ns after +12V has been applied to $\overline{RESET}$ will cause the device to fail entering programming mode.

**Chip Erase**

The Chip Erase command will erase the Flash and EEPROM memories and the Lock bits. The Lock bits are not reset until the Flash and EEPROM have been completely erased. The Fuse bits are not changed. Chip Erase must be performed before the Flash or EEPROM 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 $\overline{WR}$ a $t_{WLWH_CE}$-wide negative pulse to execute Chip Erase. See Table 30 on page 85 for $t_{WLWH_CE}$ value. Chip Erase does not generate any activity on the RDY/$\overline{BSY}$ pin.

**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 - $0F).
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 61 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 62 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 and EEPROM after a Chip Erase.

These considerations also apply to EEPROM programming and Flash, EEPROM and signature byte reading.

**Figure 61.** Programming the Flash Waveforms

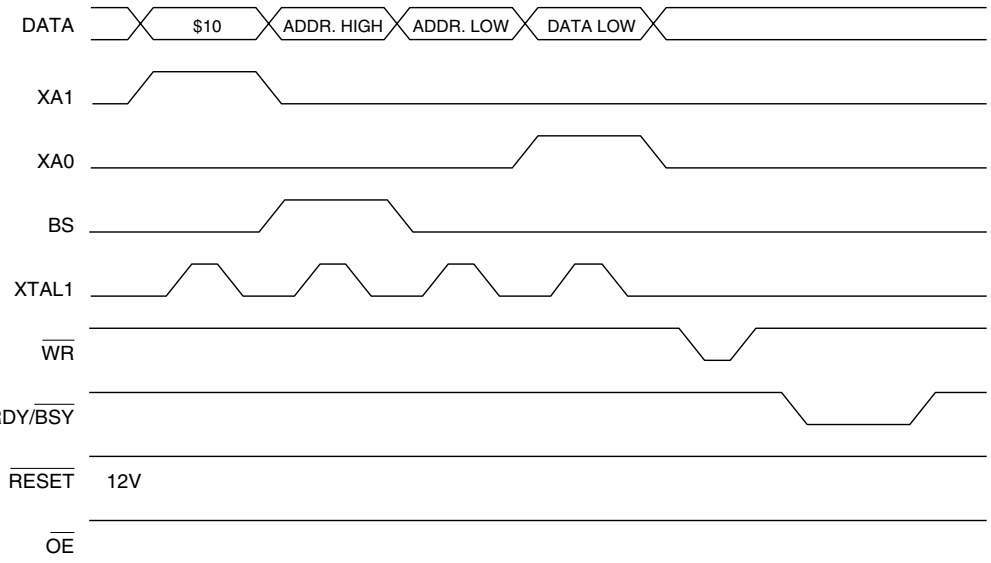
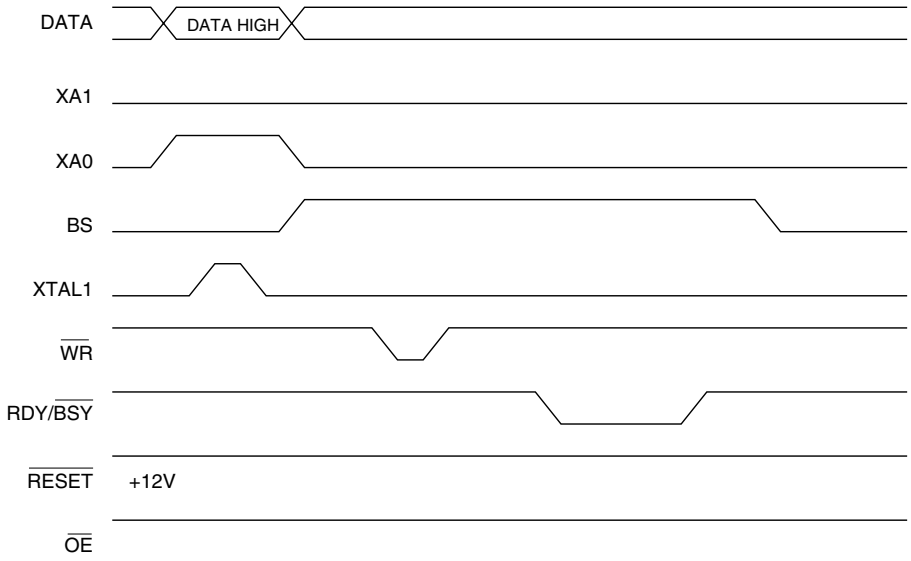


**Figure 62.** Programming the Flash Waveforms (Continued)




**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 - $0F).
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 EEPROM**

The programming algorithm for the EEPROM data memory is as follows (refer to "Programming the Flash" for details on command, address and data loading):

1. A: Load Command "0001 0001".
2. (AT90S8515 only) B: Load Address High Byte ($00 - $01).
3. C: Load Address Low Byte ($00 - $FF).
4. D: Load Data Low Byte ($00 - $FF).
5. E: Write Data Low Byte.

**Reading the EEPROM**

The algorithm for reading the EEPROM memory is as follows (refer to "Programming the Flash" for details on command and address loading):

1. A: Load Command "0000 0011".
2. (AT90S8515 only) B: Load Address High Byte ($00 - $01).
3. C: Load Address Low Byte ($00 - $FF).
4. Set $\overline{OE}$ to "0", and BS to "0". The EEPROM data byte can now be read at DATA.
5. Set $\overline{OE}$ to "1".

**Programming the Fuse Bits**

The algorithm for programming the Fuse bits is as follows (refer to "Programming the Flash" 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 5 = SPIEN Fuse bit

Bit 0 = FSTRT Fuse bit

Bit 7 - 6, 4 - 1 = “1”. These bits are reserved and should be left unprogrammed (“1”).

3. Give $\overline{WR}$ a $t_{WLWH_PFB}$-wide negative pulse to execute the programming, $t_{WLWH_PFB}$ is found in Table 30. Programming the Fuse bits does not generate any activity on the RDY/$\overline{BSY}$ pin.

**Programming the Lock Bits**

The algorithm for programming the Lock bits is as follows (refer to “Programming the Flash” on page 81 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

   Bit 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” on page 81 for details on Command loading):

1. A: Load Command “0000 0100”.
2. Set $\overline{OE}$ to “0”, and BS to “1”. The status of the Fuse and Lock bits can now be read at DATA (“0” means programmed).

   Bit 7 = Lock Bit1

   Bit 6 = Lock Bit2

   Bit 5 = SPIEN Fuse bit

   Bit 0 = FSTRT Fuse bit
3. Set $\overline{OE}$ to “1”.

Observe that BS needs to be set to “1”.

**Reading the Signature Bytes**

The algorithm for reading the signature bytes is as follows (refer to “Programming the Flash” on page 81 for details on command and address loading):

1. A: Load Command “0000 1000”.
2. C: Load Address Low Byte ($00 - $02).

   Set $\overline{OE}$ to “0”, and BS to “0”. The selected signature byte can now be read at DATA.
3. Set $\overline{OE}$ to “1”.

## Parallel Programming Characteristics

**Figure 63.** Parallel Programming Timing

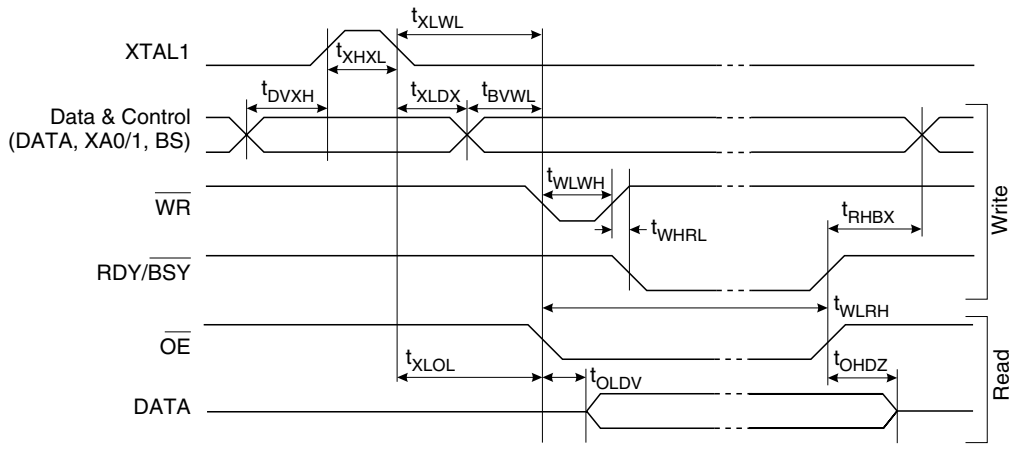



**Table 30.** Parallel Programming Characteristics, $T_A$ = 25°C ± 10%, $V_{CC}$ = 5V ± 10%

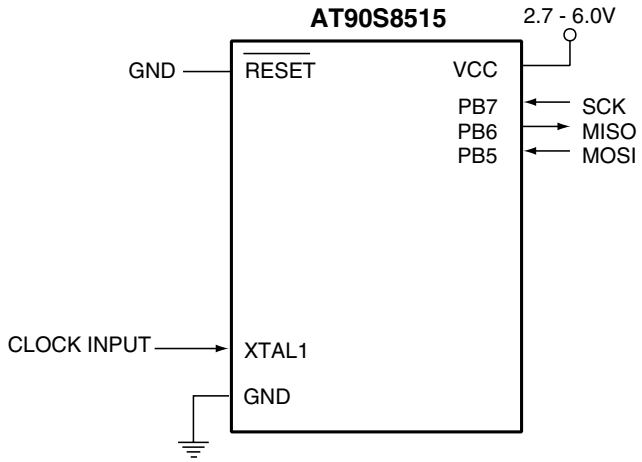
| Symbol | Parameter | Min | Typ | Max | Units |
|---|---|---|---|---|---|
| $V_{PP}$ | Programming Enable Voltage | 11.5 | | 12.5 | V |
| $I_{PP}$ | Programming Enable Current | | | 250.0 | µA |
| $t_{DVXH}$ | Data and Control Setup before XTAL1 High | 67.0 | | | ns |
| $t_{XHXL}$ | XTAL1 Pulse Width High | 67.0 | | | ns |
| $t_{XLDX}$ | Data and Control Hold after XTAL1 Low | 67.0 | | | ns |
| $t_{XLWL}$ | XTAL1 Low to $\overline{WR}$ Low | 67.0 | | | ns |
| $t_{BVWL}$ | BS Valid to $\overline{WR}$ Low | 67.0 | | | ns |
| $t_{RHBX}$ | BS Hold after RDY/$\overline{BSY}$ High | 67.0 | | | ns |
| $t_{WLWH}$ | $\overline{WR}$ Pulse Width Low(1) | 67.0 | | | ns |
| $t_{WHRL}$ | $\overline{WR}$ High to RDY/$\overline{BSY}$ Low(2) | | 20.0 | | ns |
| $t_{WLRH}$ | $\overline{WR}$ Low to RDY/$\overline{BSY}$ High(2) | 0.5 | 0.7 | 0.9 | ms |
| $t_{XLOL}$ | XTAL1 Low to $\overline{OE}$ Low | 67.0 | | | ns |
| $t_{OLDV}$ | $\overline{OE}$ Low to DATA Valid | | 20.0 | | ns |
| $t_{OHDZ}$ | $\overline{OE}$ High to DATA Tri-stated | | | 20.0 | ns |
| $t_{WLWH_CE}$ | $\overline{WR}$ Pulse Width Low for Chip Erase | 5.0 | 10.0 | 15.0 | ms |
| $t_{WLWH_PFB}$ | $\overline{WR}$ Pulse Width Low for Programming the Fuse Bits | 1.0 | 1.5 | 1.8 | ms |

Notes: 1. Use $t_{WLWH_CE}$ for Chip Erase and $t_{WLWH_PFB}$ for programming the Fuse bits.
2. If $t_{WLWH}$ is held longer than $t_{WLRH}$, no RDY/$\overline{BSY}$ pulse will be seen.

## Serial Downloading

Both the program and data memory arrays can be programmed using the SPI bus while $\overline{\text{RESET}}$ is pulled to GND. The serial interface consists of pins SCK, MOSI (input) and MISO (output). See Figure 64. After $\overline{\text{RESET}}$ is set low, the Programming Enable instruction needs to be executed first before program/erase instructions can be executed.

**Figure 64.** Serial Programming and Verify




For the EEPROM, an auto-erase cycle is provided within the self-timed Write instruction and there is no need to first execute the Chip Erase instruction. The Chip Erase instruction turns the content of every memory location in both the program and EEPROM arrays into $FF.

The program and EEPROM memory arrays have separate address spaces: $0000 to $0FFF (AT90S8515) for program memory and $0000 to $01FF (AT90S8515) for EEPROM memory.

Either an external clock is supplied at pin XTAL1 or a crystal needs to be connected across pins XTAL1 and XTAL2. The minimum low and high periods for the serial clock (SCK) input are defined as follows:

Low: > 2 XTAL1 clock cycles

High: > 2 XTAL1 clock cycles

### Serial Programming Algorithm

When writing serial data to the AT90S8515, data is clocked on the rising edge of SCK.

When reading data from the AT90S8515, data is clocked on the falling edge of SCK. See Figure 65, Figure 66 and Table 33 on page 89 for timing details.

To program and verify the AT90S8515 in the Serial Programming Mode, the following sequence is recommended (see 4-byte instruction formats in Table 32):

1. Power-up sequence:

   Apply power between $V_{CC}$ and GND while $\overline{\text{RESET}}$ and SCK are set to “0”. If a crystal is not connected across pins XTAL1 and XTAL2, apply a clock signal to the XTAL1 pin. In some systems, the programmer cannot guarantee that SCK is held low during power-up. In this case, $\overline{\text{RESET}}$ must be given a positive pulse of at least two XTAL1 cycles duration after SCK has been set to “0”.
2. Wait for at least 20 ms and enable serial programming by sending the Programming Enable serial instruction to the MOSI (PB5) pin.
3. The serial programming instructions will not work if the communication is out of synchronization. When in sync, the second byte ($53) will echo back when issu-

ing the third byte of the Programming Enable instruction. Whether the echo is correct or not, all four bytes of the instruction must be transmitted. If the $53 did not echo back, give SCK a positive pulse and issue a new Programming Enable instruction. If the $53 is not seen within 32 attempts, there is no functional device connected.

4. If a Chip Erase is performed (must be done to erase the Flash), wait $t_{WD_ERASE}$ after the instruction, give $\overline{\text{RESET}}$ a positive pulse and start over from step 2. See Table 34 on page 89 for $t_{WD_ERASE}$ value.
5. The Flash or EEPROM array is programmed one byte at a time by supplying the address and data together with the appropriate Write instruction. An EEPROM memory location is first automatically erased before new data is written. Use Data Polling to detect when the next byte in the Flash or EEPROM can be written. If polling is not used, wait $t_{WD_PROG}$ before transmitting the next instruction. See Table 35 on page 89 for $t_{WD_PROG}$ value. In an erased device, no $FFs in the data file(s) need to be programmed.
6. Any memory location can be verified by using the Read instruction that returns the content at the selected address at the serial output MISO (PB6) pin.
7. At the end of the programming session, $\overline{\text{RESET}}$ can be set high to commence normal operation.
8. Power-off sequence (if needed):

   Set XTAL1 to “0” (if a crystal is not used).

   Set $\overline{\text{RESET}}$ to “1”.

   Turn $V_{CC}$ power off.

**Data Polling EEPROM**

When a byte is being programmed into the EEPROM, reading the address location being programmed will give the value P1 until the auto-erase is finished and then the value P2. See Table 31 for P1 and P2 values.

At the time the device is ready for a new EEPROM byte, the programmed value will read correctly. This is used to determine when the next byte can be written. This will not work for the values P1 and P2, so when programming these values, the user will have to wait for at least the prescribed time $t_{WD_PROG}$ before programming the next byte. See Table 34 for $t_{WD_PROG}$ value. As a chip-erased device contains $FF in all locations, programming of addresses that are meant to contain $FF can be skipped. This does not apply if the EEPROM is reprogrammed without first chip-erasing the device.

**Table 31.** Read Back Value during EEPROM Polling

| Part | P1 | P2 |
|---|---|---|
| AT90S8515 | $80 | $7F |

**Data Polling Flash**

When a byte is being programmed into the Flash, reading the address location being programmed will give the value $7F. At the time the device is ready for a new byte, the programmed value will read correctly. This is used to determine when the next byte can be written. This will not work for the value $7F, so when programming this value, the user will have to wait for at least $t_{WD_PROG}$ before programming the next byte. As a chip-erased device contains $FF in all locations, programming of addresses that are meant to contain $FF can be skipped.

**Figure 65.** Serial Programming Waveforms

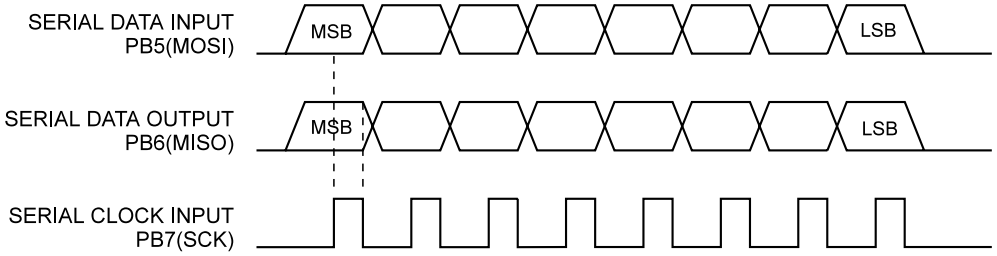



**Table 32.** Serial Programming Instruction Set

| Instruction | Instruction Format | | | | Operation |
|---|---|---|---|---|---|
| | Byte 1 | Byte 2 | Byte 3 | Byte4 | |
| Programming Enable | 1010 1100 | 0101 0011 | xxxx xxxx | xxxx xxxx | Enable serial programming while $\overline{\text{RESET}}$ is low. |
| Chip Erase | 1010 1100 | 100x xxxx | xxxx xxxx | xxxx xxxx | Chip Erase Flash and EEPROM memory arrays. |
| Read Program Memory | 0010 **H**000 | xxxx **aaaa** | **bbbb bbbb** | **oooo oooo** | Read **H** (high or low) data **o** from program memory at word address **a**:**b**. |
| Write Program Memory | 0100 **H**000 | xxxx **aaaa** | **bbbb bbbb** | **iiii iiii** | Write **H** (high or low) data **i** to program memory at word address **a**:**b**. |
| Read EEPROM Memory | 1010 0000 | xxxx xxx**a** | **bbbb bbbb** | **oooo oooo** | Read data **o** from EEPROM memory at address **a**:**b**. |
| Write EEPROM Memory | 1100 0000 | xxxx xxx**a** | **bbbb bbbb** | **iiii iiii** | Write data **i** to EEPROM memory at address **a**:**b**. |
| Write Lock Bits | 1010 1100 | 111x x***21***x | xxxx xxxx | xxxx xxxx | Write Lock bits. Set bits ***1,2*** = “0” to program Lock bits. |
| Read Signature Bytes | 0011 0000 | xxxx xxxx | xxxx xx**bb** | **oooo oooo** | Read signature byte **o** at address **b**.(1) |

Note: 1. The signature bytes are not readable in lock mode 3, i.e., both Lock bits programmed.

**a** = address high bits
**b** = address low bits
**H** = 0 – Low byte, 1 – High Byte
**o** = data out
**i** = data in
x = don’t care
***1*** = Lock bit 1
***2*** = Lock bit 2

## Serial Programming Characteristics

**Figure 66.** Serial Programming Timing

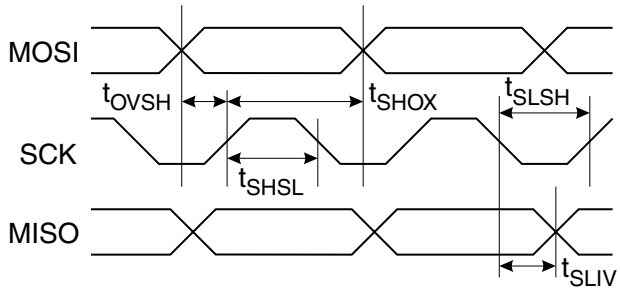



**Table 33.** Serial Programming Characteristics, $T_A$ = -40°C to 85°C, $V_{CC}$ = 2.7V - 6.0V (unless otherwise noted)

| Symbol | Parameter | Min | Typ | Max | Units |
|---|---|---|---|---|---|
| $1/t_{CLCL}$ | Oscillator Frequency ($V_{CC}$ = 2.7 - 4.0V) | 0 | | 4.0 | MHz |
| $t_{CLCL}$ | Oscillator Period ($V_{CC}$ = 2.7 - 4.0V) | 250.0 | | | ns |
| $1/t_{CLCL}$ | Oscillator Frequency ($V_{CC}$ = 4.0 - 6.0V) | 0 | | 8.0 | MHz |
| $t_{CLCL}$ | Oscillator Period ($V_{CC}$ = 4.0 - 6.0V) | 125.0 | | | ns |
| $t_{SHSL}$ | SCK Pulse Width High | 2.0 $t_{CLCL}$ | | | ns |
| $t_{SLSH}$ | SCK Pulse Width Low | 2.0 $t_{CLCL}$ | | | ns |
| $t_{OVSH}$ | MOSI Setup to SCK High | $t_{CLCL}$ | | | ns |
| $t_{SHOX}$ | MOSI Hold after SCK High | 2.0 $t_{CLCL}$ | | | ns |
| $t_{SLIV}$ | SCK Low to MISO Valid | 10.0 | 16.0 | 32.0 | ns |

**Table 34.** Minimum Wait Delay after the Chip Erase Instruction

| Symbol | 3.2V | 3.6V | 4.0V | 5.0V |
|---|---|---|---|---|
| $t_{WD_ERASE}$ | 18 ms | 14 ms | 12 ms | 8 ms |

**Table 35.** Minimum Wait Delay after Writing a Flash or EEPROM Location

| Symbol | 3.2V | 3.6V | 4.0V | 5.0V |
|---|---|---|---|---|
| $t_{WD_PROG}$ | 9 ms | 7 ms | 6 ms | 4 ms |

# Electrical Characteristics

## Absolute Maximum Ratings*

| | |
|---|---|
| Operating Temperature | -55°C to +125°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 |
| Voltage on $\overline{\text{RESET}}$ with Respect to Ground | -1.0V to +13.0V |
| Maximum Operating Voltage | 6.6V |
| DC Current per I/O Pin | 40.0 mA |
| DC Current $V_{CC}$ and GND Pins | 200.0 mA |

*NOTICE: Stresses beyond those 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

$T_A$ = -40°C to 85°C, $V_{CC}$ = 2.7V to 6.0V (unless otherwise noted)

| Symbol | Parameter | Condition | Min | Typ | Max | Units |
|---|---|---|---|---|---|---|
| $V_{IL}$ | Input Low Voltage | (Except XTAL1) | -0.5 | | 0.3 $V_{CC}$[1] | V |
| $V_{IL1}$ | Input Low Voltage | (XTAL1) | -0.5 | | 0.2 $V_{CC}$[1] | V |
| $V_{IH}$ | Input High Voltage | (Except XTAL1, $\overline{\text{RESET}}$) | 0.6 $V_{CC}$[2] | | $V_{CC}$ + 0.5 | V |
| $V_{IH1}$ | Input High Voltage | (XTAL1) | 0.8 $V_{CC}$[2] | | $V_{CC}$ + 0.5 | V |
| $V_{IH2}$ | Input High Voltage | ($\overline{\text{RESET}}$) | 0.9 $V_{CC}$[2] | | $V_{CC}$ + 0.5 | V |
| $V_{OL}$ | Output Low Voltage[3] (Ports A, B, C, D) | $I_{OL}$ = 20 mA, $V_{CC}$ = 5V<br>$I_{OL}$ = 10 mA, $V_{CC}$ = 3V | | | 0.6<br>0.5 | V<br>V |
| $V_{OH}$ | Output High Voltage[4] (Ports A, B, C, D) | $I_{OH}$ = -3 mA, $V_{CC}$ = 5V<br>$I_{OH}$ = -1.5 mA, $V_{CC}$ = 3V | 4.2<br>2.3 | | | V<br>V |
| $I_{IL}$ | Input Leakage Current I/O Pin | $V_{CC}$ = 6V, pin low (absolute value) | | | 8.0 | µA |
| $I_{IH}$ | Input Leakage Current I/O Pin | $V_{CC}$ = 6V, pin high (absolute value) | | | 980.0 | nA |
| RRST | Reset Pull-up Resistor | | 100.0 | | 500.0 | kΩ |
| $R_{I/O}$ | I/O Pin Pull-up Resistor | | 35.0 | | 120.0 | kΩ |
| $I_{CC}$ | Power Supply Current | Active Mode, $V_{CC}$ = 3V, 4 MHz | | | 3.0 | mA |
| | | Idle Mode $V_{CC}$ = 3V, 4 MHz | | | 1.2 | mA |
| | Power-down mode[5] | WDT enabled, $V_{CC}$ = 3V | | 9.0 | 15.0 | µA |
| | | WDT disabled, $V_{CC}$ = 3V | | <1.0 | 2.0 | µA |
| $V_{ACIO}$ | Analog Comparator Input Offset Voltage | $V_{CC}$ = 5V<br>$V_{in}$ = $V_{CC}$/2 | | | 40.0 | mV |
| $I_{ACLK}$ | Analog Comparator Input Leakage Current | $V_{CC}$ = 5V<br>$V_{in}$ = $V_{CC}$/2 | -50.0 | | 50.0 | nA |
| $t_{ACPD}$ | Analog Comparator Propagation Delay | $V_{CC}$ = 2.7V<br>$V_{CC}$ = 4.0V | | 750.0<br>500.0 | | 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 (20 mA at $V_{CC}$ = 5V, 10 mA at $V_{CC}$ = 3V) under steady state conditions (non-transient), the following must be observed:
   1] The sum of all $I_{OL}$, for all ports, should not exceed 200 mA.
   2] The sum of all $I_{OL}$, for ports B0 - B7, D0 - D7 and XTAL2, should not exceed 100 mA.
   3] The sum of all $I_{OL}$, for ports A0 - A7, ALE, OC1B and C0 - C7 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 condition.
4. Although each I/O port can source more than the test conditions (3 mA at $V_{CC}$ = 5V, 1.5 mA at $V_{CC}$ = 3V) under steady state conditions (non-transient), the following must be observed:
   1] The sum of all $I_{OH}$, for all ports, should not exceed 200 mA.
   2] The sum of all $I_{OH}$, for ports B0 - B7, D0 - D7 and XTAL2, should not exceed 100 mA.
   3] The sum of all $I_{OH}$, for ports A0 - A7, ALE, OC1B and C0 - C7 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 condition.
5. Minimum $V_{CC}$ for power-down is 2V.

## External Clock Drive Waveforms

**Figure 67.** External Clock

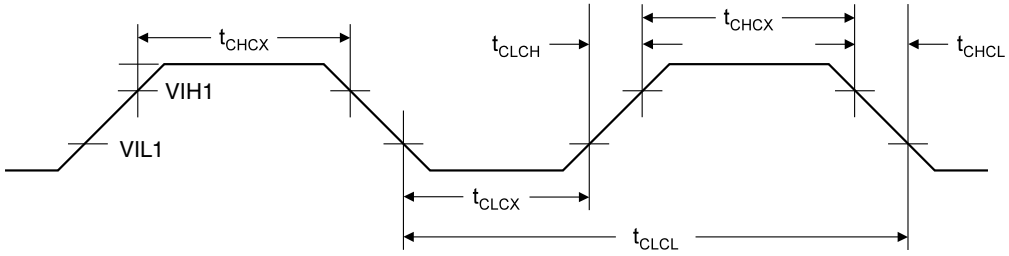
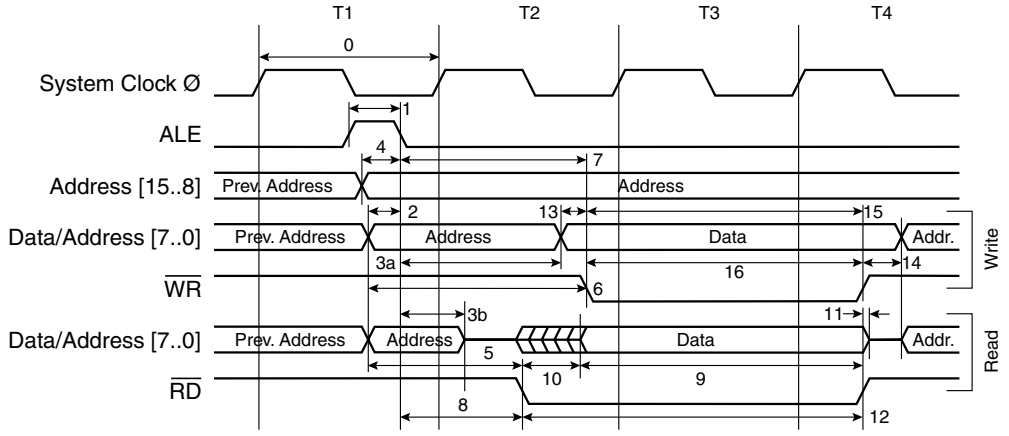



**Table 36.** External Clock Drive

| Symbol | Parameter | $V_{CC}$ = 2.7V to 4.0V | | $V_{CC}$ = 4.0V to 6.0V | | Units |
|---|---|---|---|---|---|---|
| | | Min | Max | Min | Max | |
| $1/t_{CLCL}$ | Oscillator Frequency | 0 | 4.0 | 0 | 8.0 | MHz |
| $t_{CLCL}$ | Clock Period | 250.0 | | 125.0 | | ns |
| $t_{CHCX}$ | High Time | 100.0 | | 50.0 | | ns |
| $t_{CLCX}$ | Low Time | 100.0 | | 50.0 | | ns |
| $t_{CLCH}$ | Rise Time | | 1.6 | | 0.5 | µs |
| $t_{CHCL}$ | Fall Time | | 1.6 | | 0.5 | µs |

Note: See “External Data Memory Timing” for a description of how the duty cycle influences the timing for the external data memory.

**Figure 68.** External RAM Timing




Note: Clock cycle T3 is only present when external SRAM wait state is enabled.

## External Data Memory Timing

**Table 37.** External Data Memory Characteristics, 4.0V - 6.0V, No Wait State

| | | | 8 MHz Oscillator | | Variable Oscillator | | |
|---|---|---|---|---|---|---|---|
| | Symbol | Parameter | Min | Max | Min | Max | Unit |
| 0 | $1/t_{CLCL}$ | Oscillator Frequency | | | 0.0 | 8.0 | MHz |
| 1 | $t_{LHLL}$ | ALE Pulse Width | 32.5 | | $0.5\ t_{CLCL} - 30.0^{(1)}$ | | ns |
| 2 | $t_{AVLL}$ | Address Valid A to ALE Low | 22.5 | | $0.5\ t_{CLCL} - 40.0^{(1)}$ | | ns |
| 3a | $t_{LLAX_ST}$ | Address Hold after ALE Low, ST/STD/STS Instructions | 67.5 | | $0.5\ t_{CLCL} + 5.0^{(2)}$ | | ns |
| 3b | $t_{LLAX_LD}$ | Address Hold after ALE Low, LD/LDD/LDS Instructions | 15.0 | | 15.0 | | ns |
| 4 | $t_{AVLLC}$ | Address Valid C to ALE Low | 22.5 | | $0.5\ t_{CLCL} - 40.0^{(1)}$ | | ns |
| 5 | $t_{AVRL}$ | Address Valid to RD Low | 95.0 | | $1.0\ t_{CLCL} - 30.0$ | | ns |
| 6 | $t_{AVWL}$ | Address Valid to WR Low | 157.5 | | $1.5\ t_{CLCL} - 30.0^{(1)}$ | | ns |
| 7 | $t_{LLWL}$ | ALE Low to WR Low | 105.0 | 145.0 | $1.0\ t_{CLCL} - 20.0$ | $1.0\ t_{CLCL} + 20.0$ | ns |
| 8 | $t_{LLRL}$ | ALE Low to RD Low | 42.5 | 82.5 | $0.5\ t_{CLCL} - 20.0^{(2)}$ | $0.5\ t_{CLCL} + 20.0^{(2)}$ | ns |
| 9 | $t_{DVRH}$ | Data Setup to RD High | 60.0 | | 60.0 | | ns |
| 10 | $t_{RLDV}$ | Read Low to Data Valid | | 70.0 | | $1.0\ t_{CLCL} - 55.0$ | ns |
| 11 | $t_{RHDX}$ | Data Hold after RD High | 0.0 | | 0.0 | | ns |
| 12 | $t_{RLRH}$ | RD Pulse Width | 105.0 | | $1.0\ t_{CLCL} - 20.0$ | | ns |
| 13 | $t_{DVWL}$ | Data Setup to WR Low | 27.5 | | $0.5\ t_{CLCL} - 35.0^{(2)}$ | | ns |
| 14 | $t_{WHDX}$ | Data Hold after WR High | 0.0 | | 0.0 | | ns |
| 15 | $t_{DVWH}$ | Data Valid to WR High | 95.0 | | $1.0\ t_{CLCL} - 30.0$ | | ns |
| 16 | $t_{WLWH}$ | WR Pulse Width | 42.5 | | $0.5\ t_{CLCL} - 20.0^{(1)}$ | | ns |

**Table 38.** External Data Memory Characteristics, 4.0V - 6.0V, One Cycle Wait State

| | | | 8 MHz Oscillator | | Variable Oscillator | | |
|---|---|---|---|---|---|---|---|
| | Symbol | Parameter | Min | Max | Min | Max | Unit |
| 0 | $1/t_{CLCL}$ | Oscillator Frequency | | | 0.0 | 8.0 | MHz |
| 10 | $t_{RLDV}$ | Read Low to Data Valid | | 195.0 | | $2.0\ t_{CLCL} - 55.0$ | ns |
| 12 | $t_{RLRH}$ | RD Pulse Width | 230.0 | | $2.0\ t_{CLCL} - 20.0$ | | ns |
| 15 | $t_{DVWH}$ | Data Valid to WR High | 220.0 | | $2.0\ t_{CLCL} - 30.0$ | | ns |
| 16 | $t_{WLWH}$ | WR Pulse Width | 167.5 | | $1.5\ t_{CLCL} - 20.0^{(2)}$ | | ns |

Notes: 1. This assumes 50% clock duty cycle. The half-period is actually the high time of the external clock, XTAL1.
2. This assumes 50% clock duty cycle. The half-period is actually the low time of the external clock, XTAL1.

**Table 39.** External Data Memory Characteristics, 2.7V - 4.0V, No Wait State

| | | | 4 MHz Oscillator | | Variable Oscillator | | |
|---|---|---|---|---|---|---|---|
| | Symbol | Parameter | Min | Max | Min | Max | Unit |
| 0 | $1/t_{CLCL}$ | Oscillator Frequency | | | 0.0 | 4.0 | MHz |
| 1 | $t_{LHLL}$ | ALE Pulse Width | 70.0 | | 0.5 $t_{CLCL}$ - 55.0(1) | | ns |
| 2 | $t_{AVLL}$ | Address Valid A to ALE Low | 60.0 | | 0.5 $t_{CLCL}$ - 65.0(1) | | ns |
| 3a | $t_{LLAX_ST}$ | Address Hold after ALE Low, ST/STD/STS Instructions | 130.0 | | 0.5 $t_{CLCL}$ + 5.0(2) | | ns |
| 3b | $t_{LLAX_LD}$ | Address Hold after ALE Low, LD/LDD/LDS Instructions | 15.0 | | 15.0 | | ns |
| 4 | $t_{AVLLC}$ | Address Valid C to ALE Low | 60.0 | | 0.5 $t_{CLCL}$ - 65.0(1) | | ns |
| 5 | $t_{AVRL}$ | Address Valid to RD Low | 200.0 | | 1.0 $t_{CLCL}$ - 50.0 | | ns |
| 6 | $t_{AVWL}$ | Address Valid to WR Low | 325.0 | | 1.5 $t_{CLCL}$ - 50.0(1) | | ns |
| 7 | $t_{LLWL}$ | ALE Low to WR Low | 230.0 | 270.0 | 1.0 $t_{CLCL}$ - 20.0 | 1.0 $t_{CLCL}$ + 20.0 | ns |
| 8 | $t_{LLRL}$ | ALE Low to RD Low | 105.0 | 145.0 | 0.5 $t_{CLCL}$ - 20.0(2) | 0.5 $t_{CLCL}$ + 20.0(2) | ns |
| 9 | $t_{DVRH}$ | Data Setup to RD High | 95.0 | | 95.0 | | ns |
| 10 | $t_{RLDV}$ | Read Low to Data Valid | | 170.0 | | 1.0 $t_{CLCL}$ - 80.0 | ns |
| 11 | $t_{RHDX}$ | Data Hold after RD High | 0.0 | | 0.0 | | ns |
| 12 | $t_{RLRH}$ | RD Pulse Width | 230.0 | | 1.0 $t_{CLCL}$ - 20.0 | | ns |
| 13 | $t_{DVWL}$ | Data Setup to WR Low | 70.0 | | 0.5 $t_{CLCL}$ - 55.0(1) | | ns |
| 14 | $t_{WHDX}$ | Data Hold after WR High | 0.0 | | 0.0 | | ns |
| 15 | $t_{DVWH}$ | Data Valid to WR High | 210.0 | | 1.0 $t_{CLCL}$ - 40.0 | | ns |
| 16 | $t_{WLWH}$ | WR Pulse Width | 105.0 | | 0.5 $t_{CLCL}$ - 20.0(2) | | ns |

**Table 40.** External Data Memory Characteristics, 2.7V - 4.0V, One Cycle Wait State

| | | | 4 MHz Oscillator | | Variable Oscillator | | |
|---|---|---|---|---|---|---|---|
| | Symbol | Parameter | Min | Max | Min | Max | Unit |
| 0 | $1/t_{CLCL}$ | Oscillator Frequency | | | 0.0 | 4.0 | MHz |
| 10 | $t_{RLDV}$ | Read Low to Data Valid | | 420.00 | | 2.0 $t_{CLCL}$ - 80.0 | ns |
| 12 | $t_{RLRH}$ | RD Pulse Width | 480.0 | | 2.0 $t_{CLCL}$ - 20.0 | | ns |
| 15 | $t_{DVWH}$ | Data Valid to WR High | 460.0 | | 2.0 $t_{CLCL}$ - 40.0 | | ns |
| 16 | $t_{WLWH}$ | WR Pulse Width | 355.0 | | 1.5 $t_{CLCL}$ - 20.0(2) | | ns |

Notes: 1. This assumes 50% clock duty cycle. The half-period is actually the high time of the external clock, XTAL1.
2. This assumes 50% clock duty cycle. The half-period is actually the low time of the external clock, XTAL1.

## Typical Characteristics

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. ICP is pulled high externally. 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 $C_L \bullet V_{CC} \bullet f$ where $C_L$ = load capacitance, $V_{CC}$ = operating voltage and f = average switching frequency of I/O pin.

The parts are characterized at frequencies higher than test limits. Parts are not guaranteed to function properly at frequencies 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 69.** Active Supply Current vs. Frequency

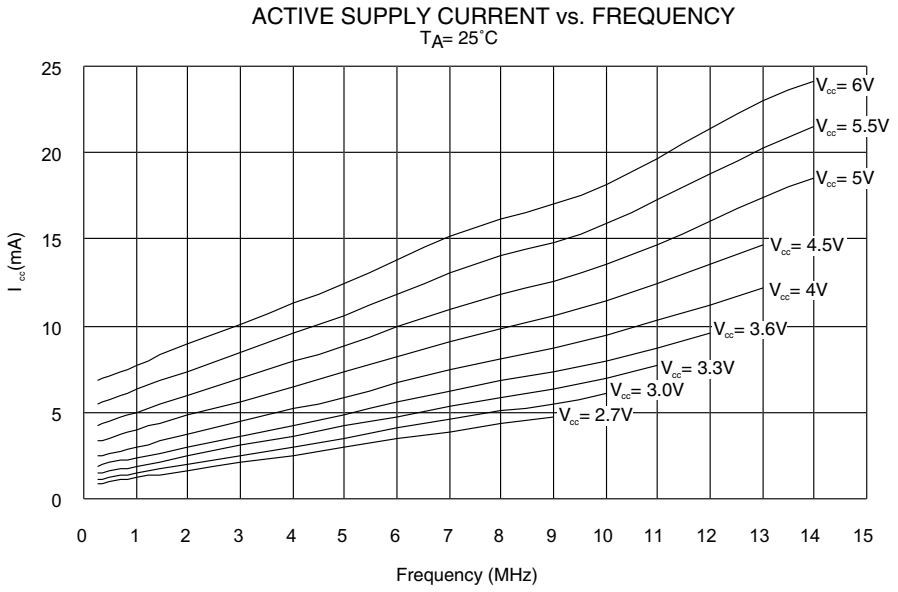


**Figure 70.** Active Supply Current vs. $V_{CC}$

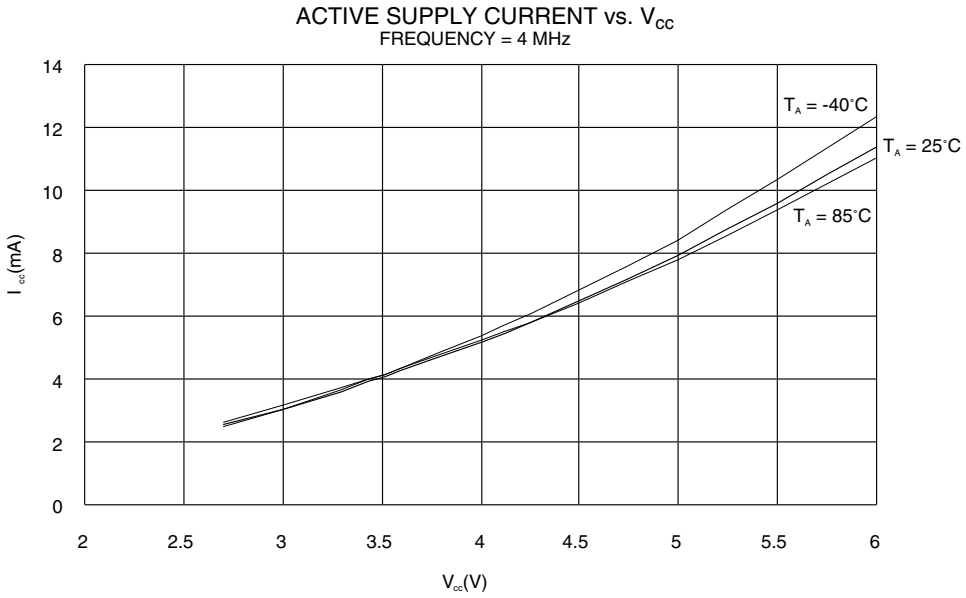



**Figure 71.** Idle Supply Current vs. Frequency

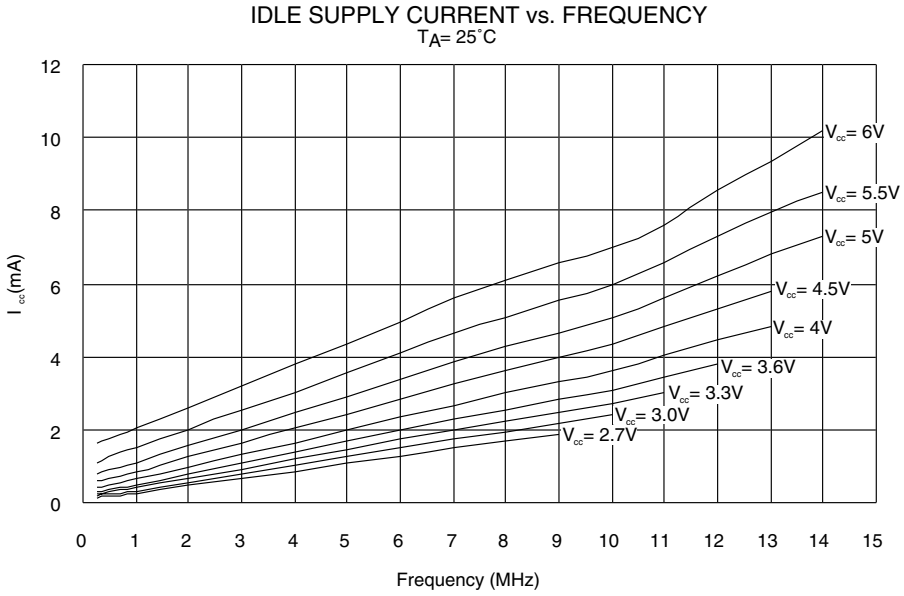


**Figure 72.** Idle Supply Current vs. $V_{CC}$

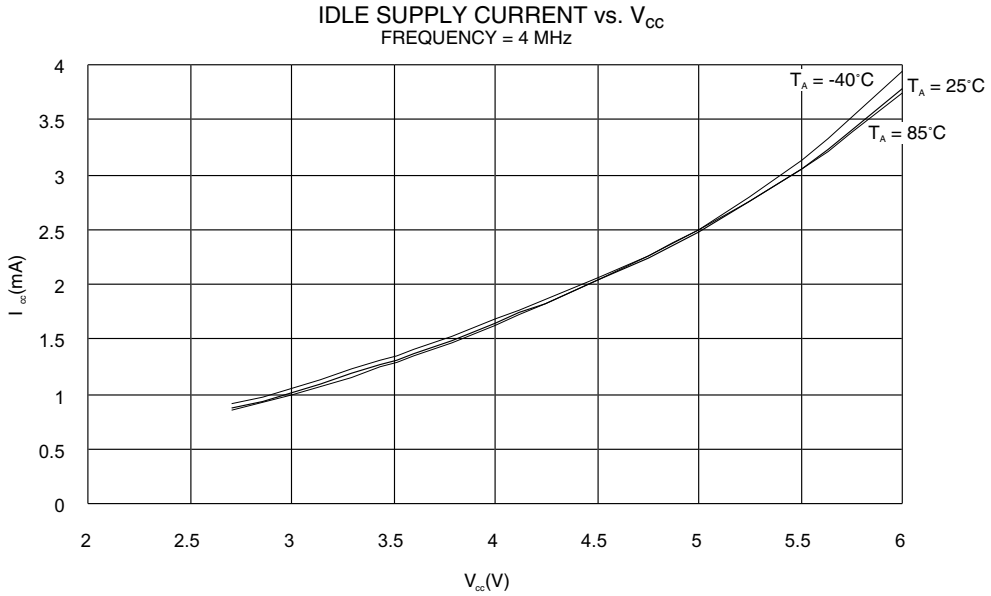



**Figure 73.** Power-down Supply Current vs. $V_{CC}$

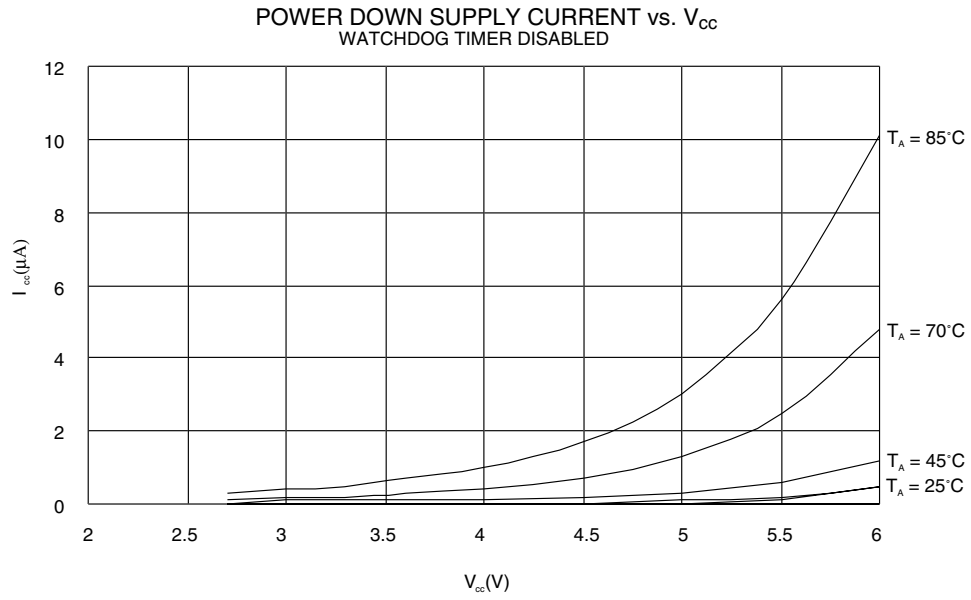


**Figure 74.** Power-down Supply Current vs. $V_{CC}$

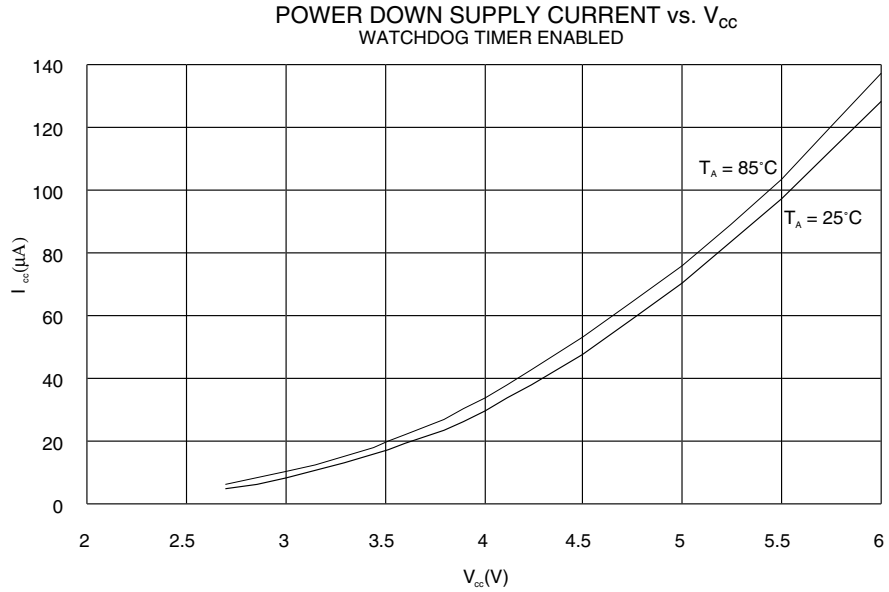
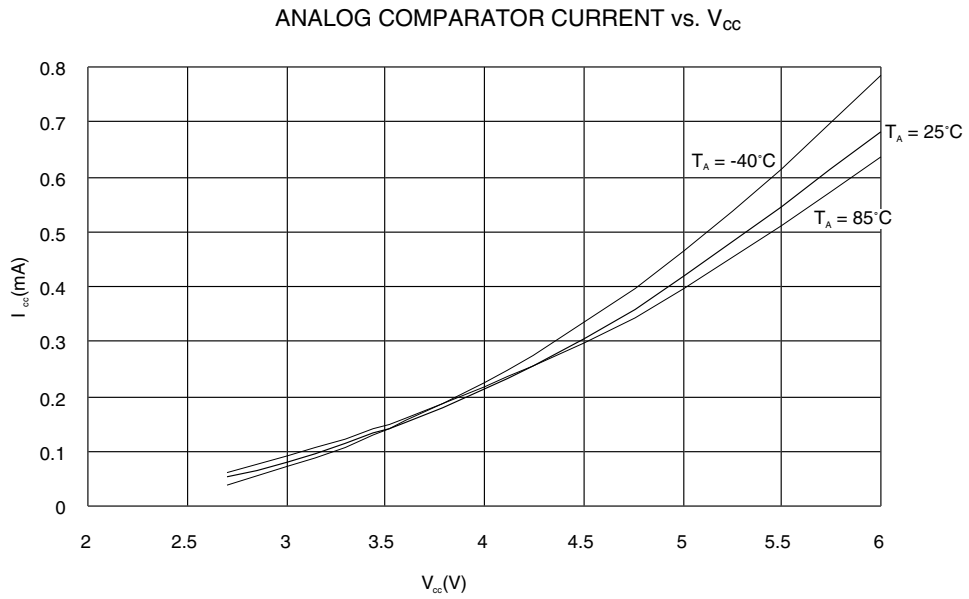



**Figure 75.** Analog Comparator Current vs. $V_{CC}$



Analog Comparator offset voltage is measured as absolute offset.

**Figure 76.** Analog Comparator Offset Voltage vs. Common Mode Voltage

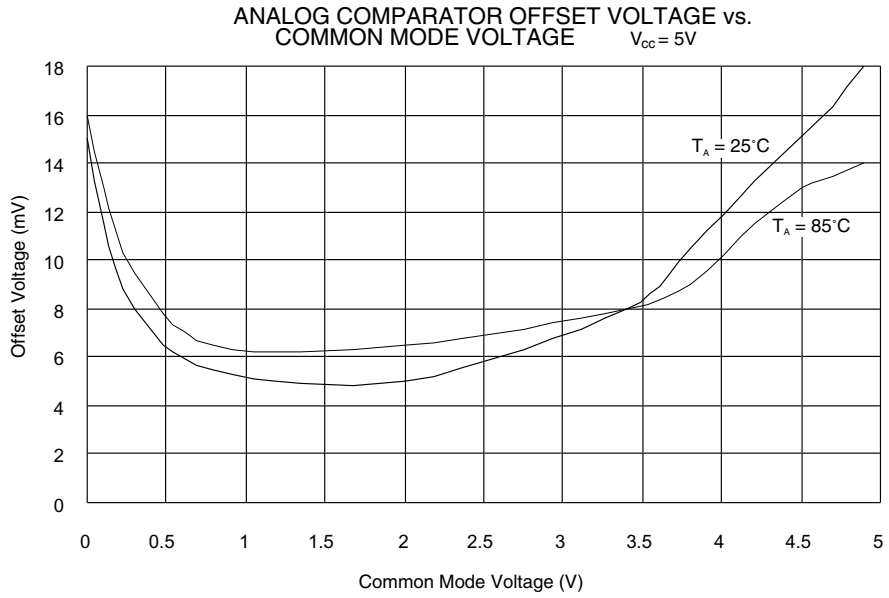



**Figure 77.** Analog Comparator Offset Voltage vs. Common Mode Voltage

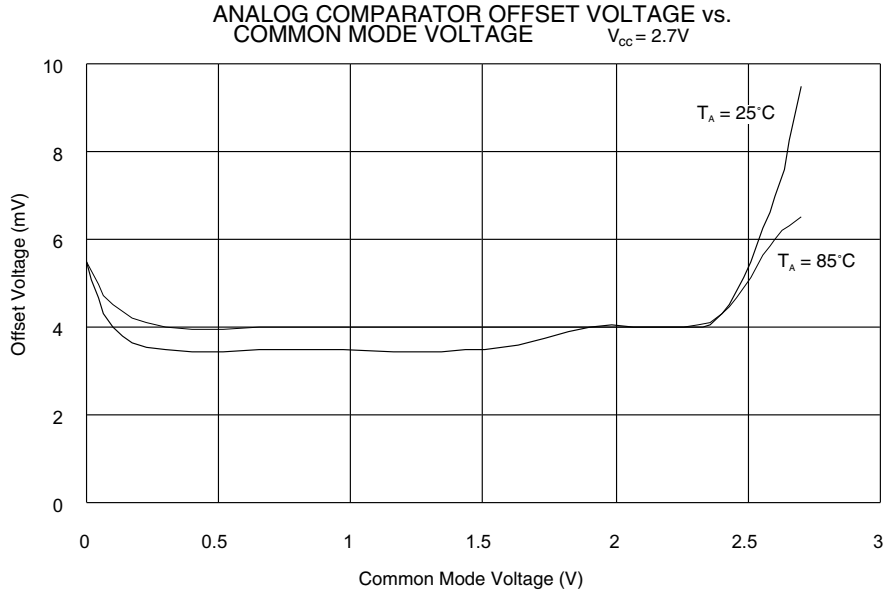


**Figure 78.** Analog Comparator Input Leakage Current

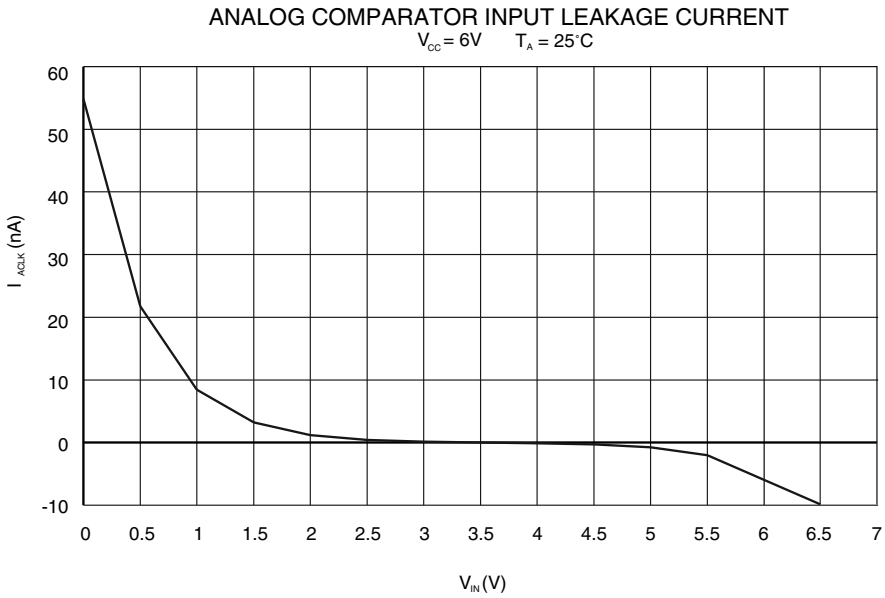
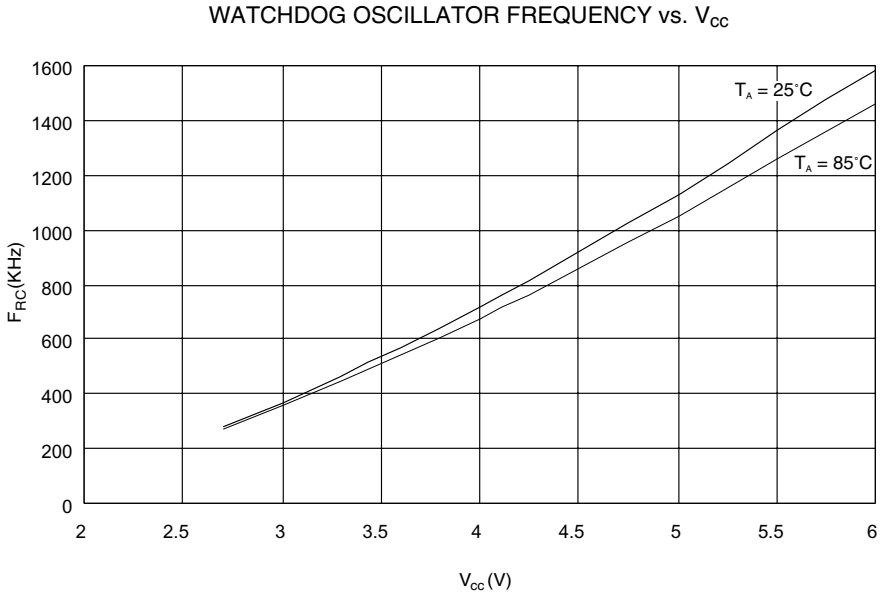



**Figure 79.** Watchdog Oscillator Frequency vs. $V_{CC}$



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

**Figure 80.** Pull-up Resistor Current vs. Input Voltage

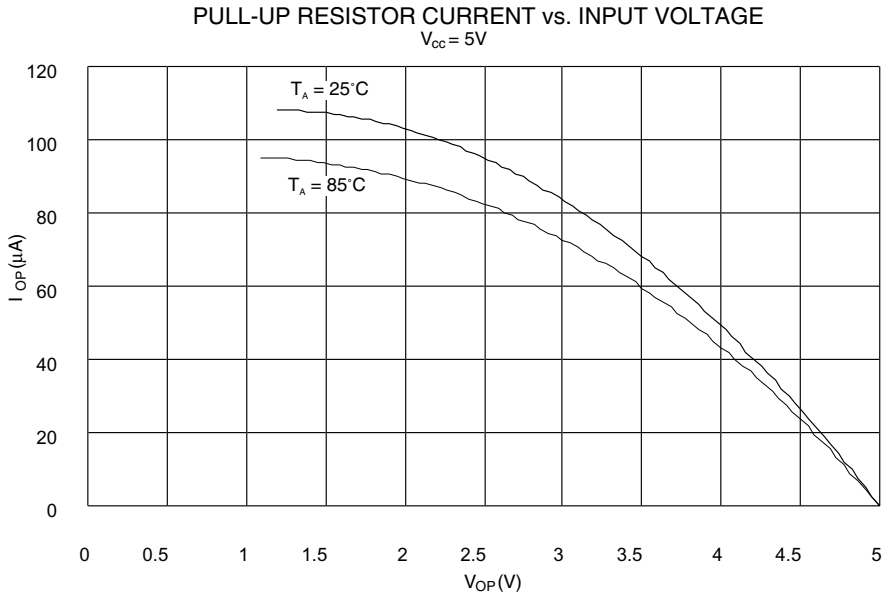



**Figure 81.** Pull-up Resistor Current vs. Input Voltage

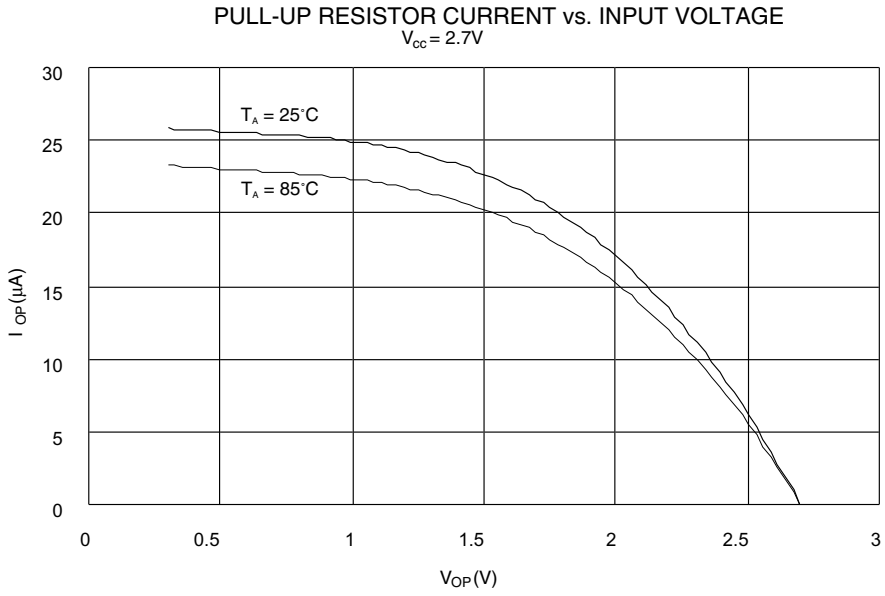


**Figure 82.** I/O Pin Sink Current vs. Output Voltage

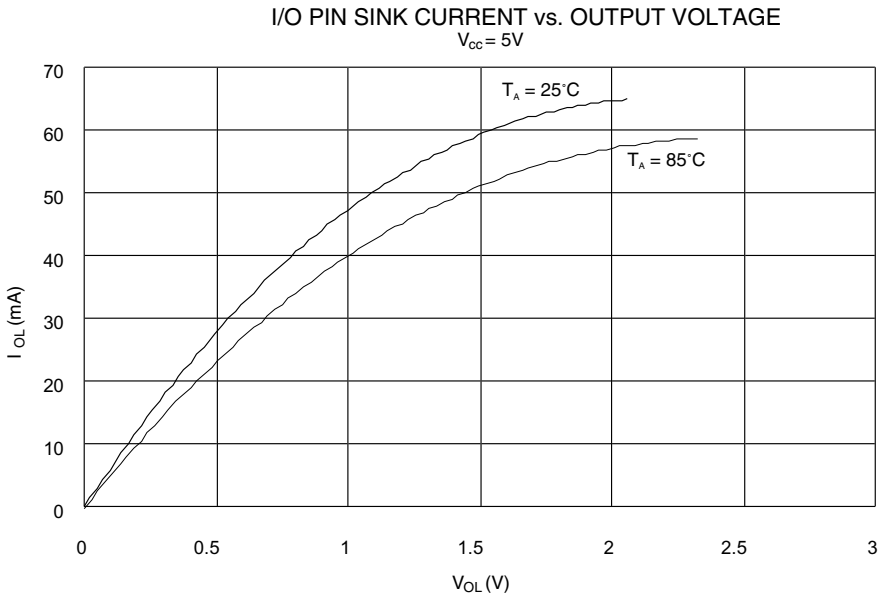



**Figure 83.** I/O Pin Source Current vs. Output Voltage

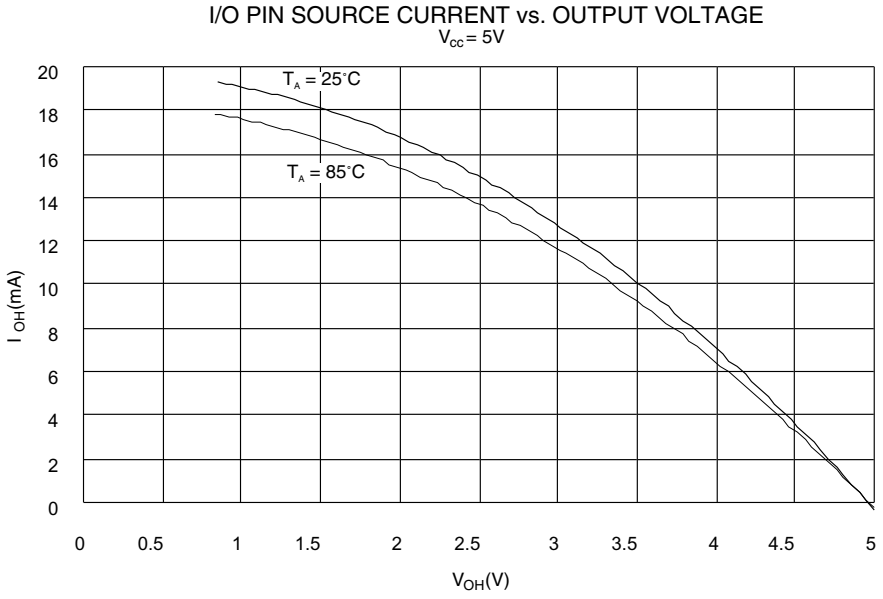


**Figure 84.** I/O Pin Source Current vs. Output Voltage

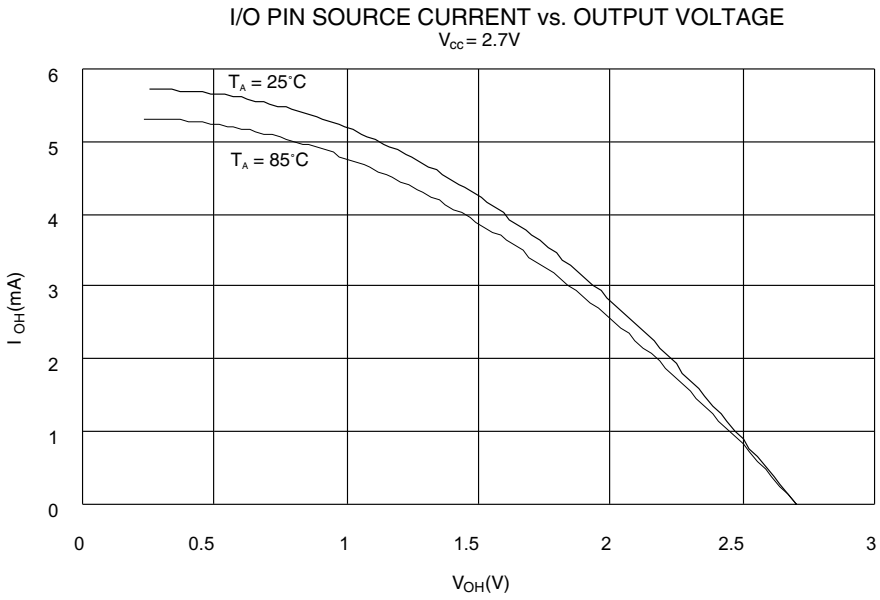



**Figure 85.** I/O Pin Input Threshold Voltage vs. $V_{CC}$

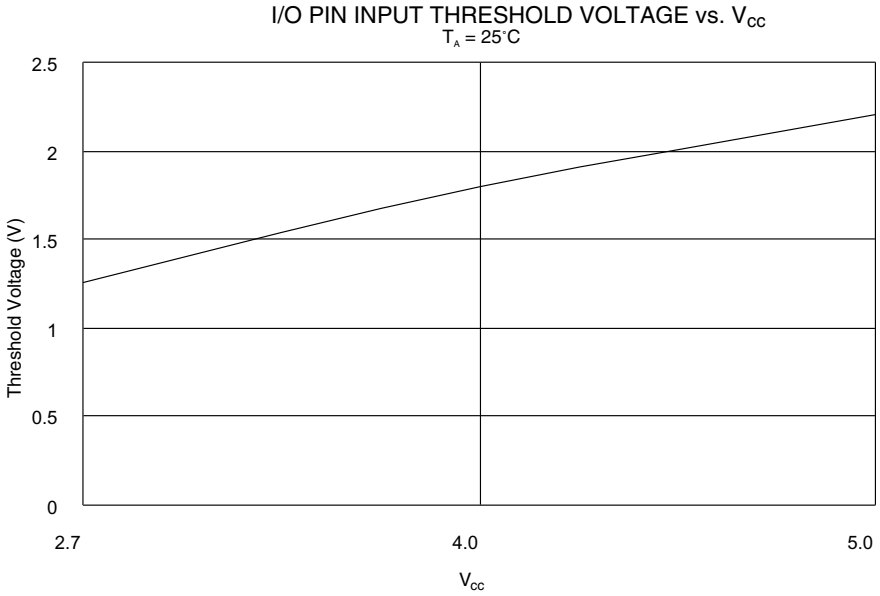


**Figure 86.** I/O Pin Input Hysteresis vs. $V_{CC}$

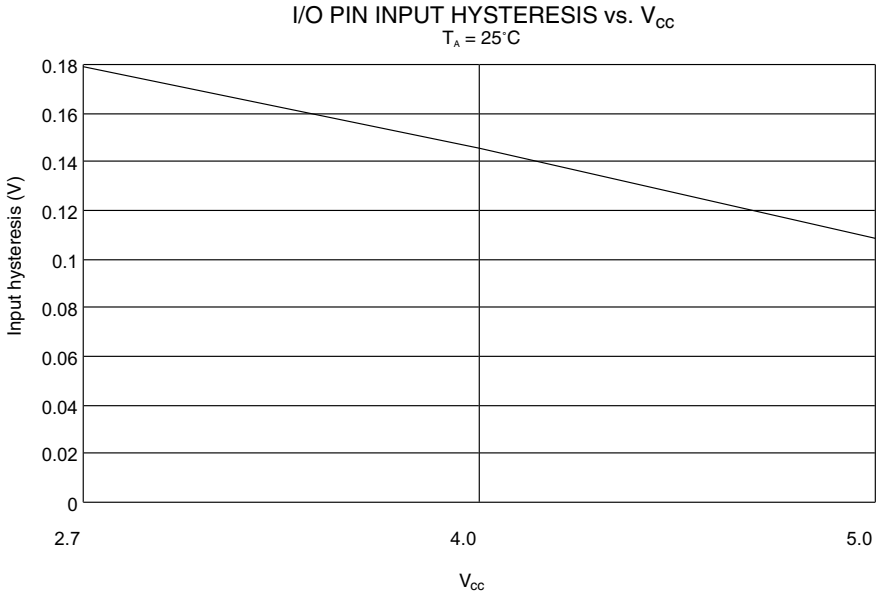
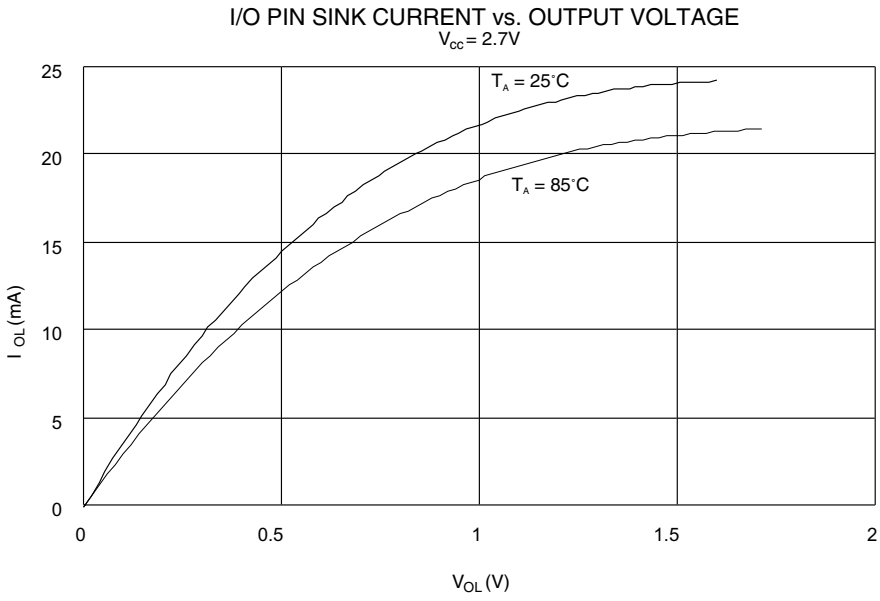



**Figure 87.** I/O Pin Sink Current vs. Output Voltage



## Register Summary

| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Page |
|---|---|---|---|---|---|---|---|---|---|---|
| $3F ($5F) | SREG | I | T | H | S | V | N | Z | C | page 20 |
| $3E ($5E) | SPH | SP15 | SP14 | SP13 | SP12 | SP11 | SP10 | SP9 | SP8 | page 21 |
| $3D ($5D) | SPL | SP7 | SP6 | SP5 | SP4 | SP3 | SP2 | SP1 | SP0 | page 21 |
| $3C ($5C) | Reserved | | | | | | | | | |
| $3B ($5B) | GIMSK | INT1 | INT0 | - | - | - | - | - | - | page 26 |
| $3A ($5A) | GIFR | INTF1 | INTF0 | | | | | | | page 26 |
| $39 ($59) | TIMSK | TOIE1 | OCIE1A | OCIE1B | - | TICIE1 | - | TOIE0 | - | page 27 |
| $38 ($58) | TIFR | TOV1 | OCF1A | OCF1B | - | ICF1 | - | TOV0 | - | page 28 |
| $37 ($57) | Reserved | | | | | | | | | |
| $36 ($56) | Reserved | | | | | | | | | |
| $35 ($55) | MCUCR | SRE | SRW | SE | SM | ISC11 | ISC10 | ISC01 | ISC00 | page 29 |
| $34 ($54) | Reserved | | | | | | | | | |
| $33 ($53) | TCCR0 | - | - | - | - | - | CS02 | CS01 | CS00 | page 33 |
| $32 ($52) | TCNT0 | Timer/Counter0 (8 Bits) | | | | | | | | page 34 |
| ... | Reserved | | | | | | | | | |
| $2F ($4F) | TCCR1A | COM1A1 | COM1A0 | COM1B1 | COM1B0 | - | - | PWM11 | PWM10 | page 36 |
| $2E ($4E) | TCCR1B | ICNC1 | ICES1 | - | - | CTC1 | CS12 | CS11 | CS10 | page 37 |
| $2D ($4D) | TCNT1H | Timer/Counter1 – Counter Register High Byte | | | | | | | | page 38 |
| $2C ($4C) | TCNT1L | Timer/Counter1 – Counter Register Low Byte | | | | | | | | page 38 |
| $2B ($4B) | OCR1AH | Timer/Counter1 – Output Compare Register A High Byte | | | | | | | | page 38 |
| $2A ($4A) | OCR1AL | Timer/Counter1 – Output Compare Register A Low Byte | | | | | | | | page 38 |
| $29 ($49) | OCR1BH | Timer/Counter1 – Output Compare Register B High Byte | | | | | | | | page 39 |
| $28 ($48) | OCR1BL | Timer/Counter1 – Output Compare Register B Low Byte | | | | | | | | page 39 |
| ... | Reserved | | | | | | | | | |
| $25 ($45) | ICR1H | Timer/Counter1 – Input Capture Register High Byte | | | | | | | | page 39 |
| $24 ($44) | ICR1L | Timer/Counter1 – Input Capture Register Low Byte | | | | | | | | page 39 |
| ... | Reserved | | | | | | | | | |
| $21 ($41) | WDTCR | - | - | - | WDTOE | WDE | WDP2 | WDP1 | WDP0 | page 42 |
| $20 ($40) | Reserved | | | | | | | | | |
| $1F ($3F) | EEARH | - | - | - | - | - | - | - | EEAR8 | page 44 |
| $1E ($3E) | EEARL | EEPROM Address Register Low Byte | | | | | | | | page 44 |
| $1D ($3D) | EEDR | EEPROM Data Register | | | | | | | | page 44 |
| $1C ($3C) | EECR | - | - | - | - | - | EEMWE | EEWE | EERE | page 44 |
| $1B ($3B) | PORTA | PORTA7 | PORTA6 | PORTA5 | PORTA4 | PORTA3 | PORTA2 | PORTA1 | PORTA0 | page 63 |
| $1A ($3A) | DDRA | DDA7 | DDA6 | DDA5 | DDA4 | DDA3 | DDA2 | DDA1 | DDA0 | page 63 |
| $19 ($39) | PINA | PINA7 | PINA6 | PINA5 | PINA4 | PINA3 | PINA2 | PINA1 | PINA0 | page 63 |
| $18 ($38) | PORTB | PORTB7 | PORTB6 | PORTB5 | PORTB4 | PORTB3 | PORTB2 | PORTB1 | PORTB0 | page 65 |
| $17 ($37) | DDRB | DDB7 | DDB6 | DDB5 | DDB4 | DDB3 | DDB2 | DDB1 | DDB0 | page 65 |
| $16 ($36) | PINB | PINB7 | PINB6 | PINB5 | PINB4 | PINB3 | PINB2 | PINB1 | PINB0 | page 65 |
| $15 ($35) | PORTC | PORTC7 | PORTC6 | PORTC5 | PORTC4 | PORTC3 | PORTC2 | PORTC1 | PORTC0 | page 70 |
| $14 ($34) | DDRC | DDC7 | DDC6 | DDC5 | DDC4 | DDC3 | DDC2 | DDC1 | DDC0 | page 71 |
| $13 ($33) | PINC | PINC7 | PINC6 | PINC5 | PINC4 | PINC3 | PINC2 | PINC1 | PINC0 | page 71 |
| $12 ($32) | PORTD | PORTD7 | PORTD6 | PORTD5 | PORTD4 | PORTD3 | PORTD2 | PORTD1 | PORTD0 | page 73 |
| $11 ($31) | DDRD | DDD7 | DDD6 | DDD5 | DDD4 | DDD3 | DDD2 | DDD1 | DDD0 | page 73 |
| $10 ($30) | PIND | PIND7 | PIND6 | PIND5 | PIND4 | PIND3 | PIND2 | PIND1 | PIND0 | page 73 |
| $0F ($2F) | SPDR | SPI Data Register | | | | | | | | page 51 |
| $0E ($2E) | SPSR | SPIF | WCOL | - | - | - | - | - | - | page 50 |
| $0D ($2D) | SPCR | SPIE | SPE | DORD | MSTR | CPOL | CPHA | SPR1 | SPR0 | page 49 |
| $0C ($2C) | UDR | UART I/O Data Register | | | | | | | | page 55 |
| $0B ($2B) | USR | RXC | TXC | UDRE | FE | OR | - | - | - | page 55 |
| $0A ($2A) | UCR | RXCIE | TXCIE | UDRIE | RXEN | TXEN | CHR9 | RXB8 | TXB8 | page 56 |
| $09 ($29) | UBRR | UART Baud Rate Register | | | | | | | | page 58 |
| $08 ($28) | ACSR | ACD | - | ACO | ACI | ACIE | ACIC | ACIS1 | ACIS0 | page 59 |
| … | Reserved | | | | | | | | | |
| $00 ($20) | Reserved | | | | | | | | | |

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 “1” 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

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

## Instruction Set Summary (Continued)

| Mnemonic | Operands | Description | Operation | Flags | # Clocks |
|---|---|---|---|---|---|
| **DATA TRANSFER INSTRUCTIONS** | | | | | |
| MOV | Rd, Rr | Move between Registers | Rd ← Rr | None | 1 |
| LDI | Rd, K | Load Immediate | Rd ← K | None | 1 |
| LD | Rd, X | Load Indirect | Rd ← (X) | None | 2 |
| LD | Rd, X+ | Load Indirect and Post-inc. | Rd ← (X), X ← X + 1 | None | 2 |
| LD | Rd, -X | Load Indirect and Pre-dec. | X ← X - 1, Rd ← (X) | None | 2 |
| LD | Rd, Y | Load Indirect | Rd ← (Y) | None | 2 |
| LD | Rd, Y+ | Load Indirect and Post-inc. | Rd ← (Y), Y ← Y + 1 | None | 2 |
| LD | Rd, -Y | Load Indirect and Pre-dec. | Y ← Y - 1, Rd ← (Y) | None | 2 |
| LDD | Rd, Y+q | Load Indirect with Displacement | Rd ← (Y + q) | None | 2 |
| LD | Rd, Z | Load Indirect | Rd ← (Z) | None | 2 |
| LD | Rd, Z+ | Load Indirect and Post-inc. | Rd ← (Z), Z ← Z + 1 | None | 2 |
| LD | Rd, -Z | Load Indirect and Pre-dec. | Z ← Z - 1, Rd ← (Z) | None | 2 |
| LDD | Rd, Z+q | Load Indirect with Displacement | Rd ← (Z + q) | None | 2 |
| LDS | Rd, k | Load Direct from SRAM | Rd ← (k) | None | 2 |
| ST | X, Rr | Store Indirect | (X) ← Rr | None | 2 |
| ST | X+, Rr | Store Indirect and Post-inc. | (X) ← Rr, X ← X + 1 | None | 2 |
| ST | -X, Rr | Store Indirect and Pre-dec. | X ← X - 1, (X) ← Rr | None | 2 |
| ST | Y, Rr | Store Indirect | (Y) ← Rr | None | 2 |
| ST | Y+, Rr | Store Indirect and Post-inc. | (Y) ← Rr, Y ← Y + 1 | None | 2 |
| ST | -Y, Rr | Store Indirect and Pre-dec. | Y ← Y - 1, (Y) ← Rr | None | 2 |
| STD | Y+q, Rr | Store Indirect with Displacement | (Y + q) ← Rr | None | 2 |
| ST | Z, Rr | Store Indirect | (Z) ← Rr | None | 2 |
| ST | Z+, Rr | Store Indirect and Post-inc. | (Z) ← Rr, Z ← Z + 1 | None | 2 |
| ST | -Z, Rr | Store Indirect and Pre-dec. | Z ← Z - 1, (Z) ← Rr | None | 2 |
| STD | Z+q, Rr | Store Indirect with Displacement | (Z + q) ← Rr | None | 2 |
| STS | k, Rr | Store Direct to SRAM | (k) ← Rr | None | 2 |
| LPM | | Load Program Memory | R0 ← (Z) | None | 3 |
| IN | Rd, P | In Port | Rd ← P | None | 1 |
| OUT | P, Rr | Out Port | P ← Rr | None | 1 |
| PUSH | Rr | Push Register on Stack | STACK ← Rr | None | 2 |
| POP | Rd | Pop Register from Stack | Rd ← STACK | None | 2 |
| **BIT AND BIT-TEST INSTRUCTIONS** | | | | | |
| SBI | P, b | Set Bit in I/O Register | I/O(P,b) ← 1 | None | 2 |
| CBI | P, b | Clear Bit in I/O Register | I/O(P,b) ← 0 | None | 2 |
| LSL | Rd | Logical Shift Left | Rd(n+1) ← Rd(n), Rd(0) ← 0 | Z,C,N,V | 1 |
| LSR | Rd | Logical Shift Right | Rd(n) ← Rd(n+1), Rd(7) ← 0 | Z,C,N,V | 1 |
| ROL | Rd | Rotate Left through Carry | Rd(0) ← C, Rd(n+1) ← Rd(n), C ← Rd(7) | Z,C,N,V | 1 |
| ROR | Rd | Rotate Right through Carry | Rd(7) ← C, Rd(n) ← Rd(n+1), C ← Rd(0) | Z,C,N,V | 1 |
| ASR | Rd | Arithmetic Shift Right | Rd(n) ← Rd(n+1), n = 0..6 | Z,C,N,V | 1 |
| SWAP | Rd | Swap Nibbles | Rd(3..0) ← Rd(7..4), Rd(7..4) ← Rd(3..0) | None | 1 |
| BSET | s | Flag Set | SREG(s) ← 1 | SREG(s) | 1 |
| BCLR | s | Flag Clear | SREG(s) ← 0 | SREG(s) | 1 |
| BST | Rr, b | Bit Store from Register to T | T ← Rr(b) | T | 1 |
| BLD | Rd, b | Bit Load from T to Register | Rd(b) ← T | None | 1 |
| SEC | | Set Carry | C ← 1 | C | 1 |
| CLC | | Clear Carry | C ← 0 | C | 1 |
| SEN | | Set Negative Flag | N ← 1 | N | 1 |
| CLN | | Clear Negative Flag | N ← 0 | N | 1 |
| SEZ | | Set Zero Flag | Z ← 1 | Z | 1 |
| CLZ | | Clear Zero Flag | Z ← 0 | Z | 1 |
| SEI | | Global Interrupt Enable | I ← 1 | I | 1 |
| CLI | | Global Interrupt Disable | I ← 0 | I | 1 |
| SES | | Set Signed Test Flag | S ← 1 | S | 1 |
| CLS | | Clear Signed Test Flag | S ← 0 | S | 1 |
| SEV | | Set Two's Complement Overflow | V ← 1 | V | 1 |
| CLV | | Clear Two's Complement Overflow | V ← 0 | V | 1 |
| SET | | Set T in SREG | T ← 1 | T | 1 |
| CLT | | Clear T in SREG | T ← 0 | T | 1 |
| SEH | | Set Half-carry Flag in SREG | H ← 1 | H | 1 |
| CLH | | Clear Half-carry Flag in SREG | H ← 0 | H | 1 |
| NOP | | No Operation | | None | 1 |
| SLEEP | | Sleep | (see specific descr. for Sleep function) | None | 1 |
| WDR | | Watchdog Reset | (see specific descr. for WDR/timer) | None | 1 |

## AT90S8515 Ordering Information

| Speed (MHz) | Power Supply | Ordering Code | Package | Operation Range |
|---|---|---|---|---|
| 4 | 2.7V - 6.0V | AT90S8515-4AC<br>AT90S8515-4JC<br>AT90S8515-4PC | 44A<br>44J<br>40P6 | Commercial<br>(0°C to 70°C) |
| | | AT90S8515-4AI<br>AT90S8515-4JI<br>AT90S8515-4PI | 44A<br>44J<br>40P6 | Industrial<br>(-40°C to 85°C) |
| 8 | 4.0V - 6.0V | AT90S8515-8AC<br>AT90S8515-8JC<br>AT90S8515-8PC | 44A<br>44J<br>40P6 | Commercial<br>(0°C to 70°C) |
| | | AT90S8515-8AI<br>AT90S8515-8JI<br>AT90S8515-8PI | 44A<br>44J<br>40P6 | Industrial<br>(-40°C to 85°C) |

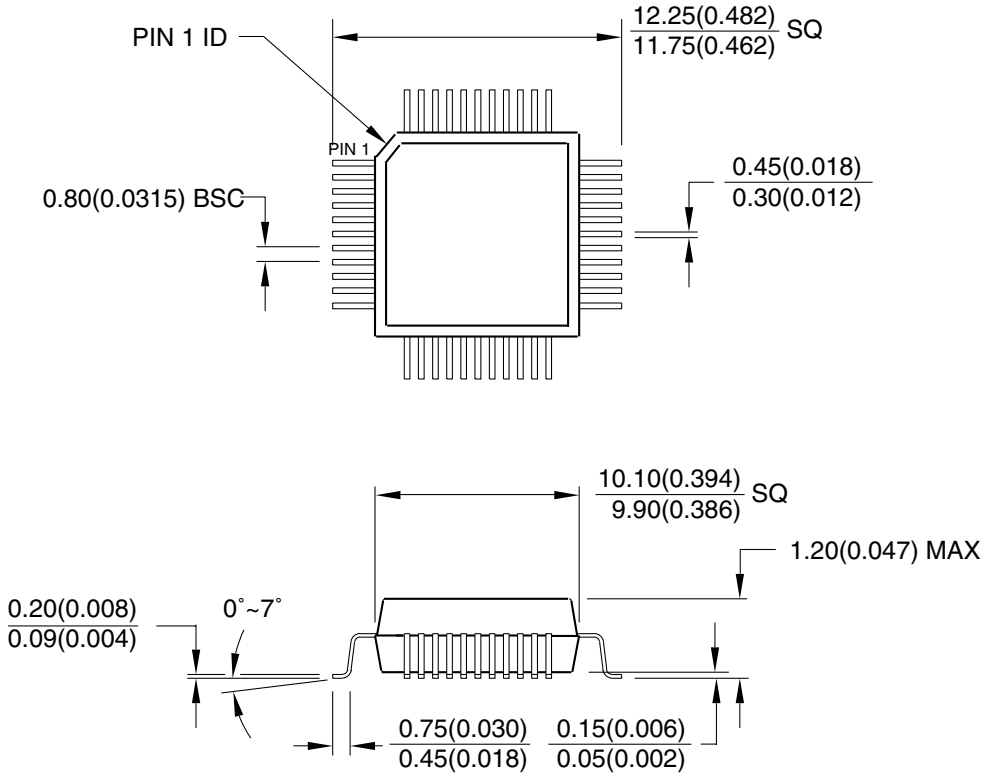
Note: Order AT90S8515A-XXX for devices with the FSTRT Fuse programmed.

| Package Type | |
|---|---|
| **44A** | 44-lead, Thin (1.0 mm) Plastic Gull Wing Quad Flat Package (TQFP) |
| **44J** | 44-lead, Plastic J-leaded Chip Carrier (PLCC) |
| **40P6** | 40-lead, 0.600" Wide, Plastic Dual Inline Package (PDIP) |

## Packaging Information

### 44A

44-lead, Thin (1.0mm) Plastic Quad Flat Package
(TQFP), 10x10mm body, 2.0mm footprint, 0.8mm pitch.
Dimension in Millimeters and (Inches)*
JEDEC STANDARD MS-026 ACB

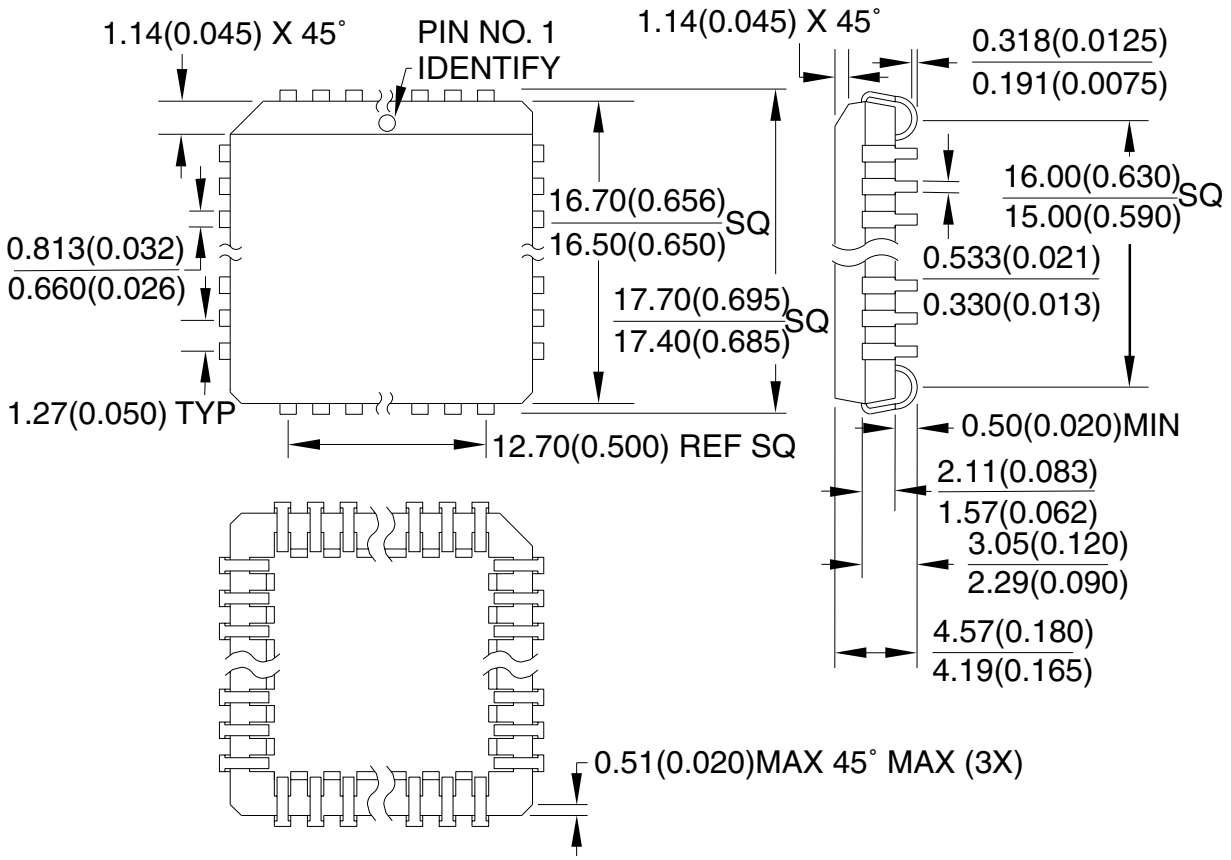



*Controlling dimension: millimetter

REV. A 04/11/2001

**44J**

44J, 44-lead, Plastic J-leaded Chip Carrier (PLCC)
Dimensions in Milimeters and (Inches)*
JEDEC STANDARD MS-018 AC

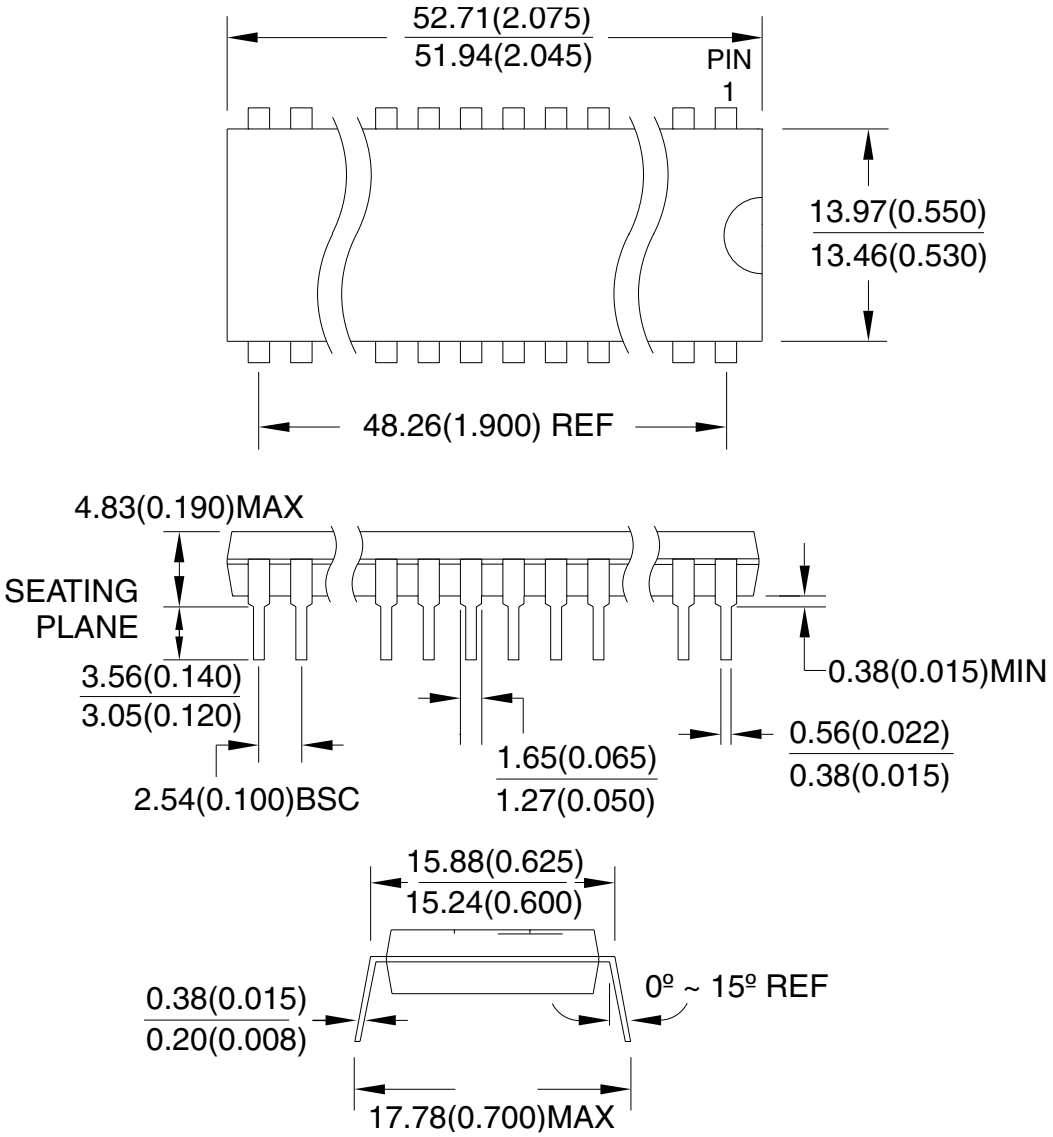



*Controlling dimensions: Inches

REV. A 04/11/2001

**40P6**

40-lead, Plastic Dual Inline
Parkage (PDIP), 0.600" wide
Demension in Millimeters and (Inches)*
JEDEC STANDARD MS-011 AC




*Controlling dimension: Inches

REV. A 04/11/2001




## Atmel Headquarters

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

*Europe*
Atmel SarL
Route des Arsenaux 41
Casa Postale 80
CH-1705 Fribourg
Switzerland
TEL (41) 26-426-5555
FAX (41) 26-426-5500

*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 Product Operations

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

*Atmel Grenoble*
Avenue de Rochepleine
BP 123
38521 Saint-Egreve Cedex, France
TEL (33) 4-7658-3000
FAX (33) 4-7658-3480

*Atmel Heilbronn*
Theresienstrasse 2
POB 3535
D-74025 Heilbronn, Germany
TEL (49) 71 31 67 25 94
FAX (49) 71 31 67 24 23

*Atmel Nantes*
La Chantrerie
BP 70602
44306 Nantes Cedex 3, France
TEL (33) 0 2 40 18 18 18
FAX (33) 0 2 40 18 19 60

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

*Atmel Smart Card ICs*
Scottish Enterprise Technology Park
East Kilbride, Scotland G75 0QR
TEL (44) 1355-357-000
FAX (44) 1355-242-743

*e-mail*
literature@atmel.com

*Web Site*
http://www.atmel.com

*BBS*
1-(408) 436-4309



**© Atmel Corporation 2001.**
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.

ATMEL® and AVR® are the registered trademarks of Atmel.

Other terms and product names may be the trademarks of others.

Printed on recycled paper.



0841G–09/01/xM

**ООО «НИОКРсистемс» -** это оперативные поставки широкого спектра электронных компонентов отечественного и импортного производства напрямую от производителей и с крупнейших мировых складов. Реализуемая нашей компанией продукция насчитывает более полумиллиона наименований.

Благодаря этому наша компания предлагает к поставке практически не ограниченный ассортимент компонентов как оптовыми, мелкооптовыми партиями, так и в розницу.

Благодаря развитой сети поставщиков, помогаем в поиске и приобретении экзотичных или снятых с производства компонентов.

## Наша компания это:

- Гарантия качества поставляемой продукции
- Широкий ассортимент
- Минимальные сроки поставок
- Техническая поддержка
- Подбор комплектации
- Индивидуальный подход
- Гибкое ценообразование
- Работаем по 275 ФЗ

ООО «НИОКР СИТЕМС»
Москва, 125362, Россия, ул. Вишневая, д. 9, к. 1, Блок 104 а и 104 в

Телефон: 8 (495) 268-14-82
Email: n@nsistems.ru

ИНН: 7735154786
ОГРН: 1167746717709