

## Features

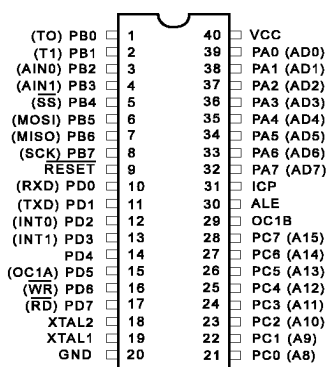
- Utilizes the *AVR*<sup>®</sup> Enhanced RISC Architecture
- *AVR* - High Performance and Low Power RISC Architecture
- 120 Powerful Instructions - Most Single Clock Cycle Execution
- 8K bytes of In-System Reprogrammable Downloadable Flash
  - SPI Serial Interface for Program Downloading
  - Endurance: 1,000 Write/Erase Cycles
- 512 bytes EEPROM
  - Endurance: 100,000 Write/Erase Cycles
- 512 bytes Internal SRAM
- 32 x 8 General Purpose Working Registers
- 32 Programmable I/O Lines
- Programmable Serial UART
- SPI Serial Interface
- V<sub>CC</sub>: 2.7 - 6.0V
- Fully Static Operation, 0 - 20 MHz
- Instruction Cycle Time: 50 ns @ 20 MHz
- One 8-Bit Timer/Counter with Separate Prescaler
- One 16-Bit Timer/Counter with Separate Prescaler and Compare and Capture Modes
- Dual PWM
- External and Internal Interrupt Sources
- Programmable Watchdog Timer with On-Chip Oscillator
- On-Chip Analog Comparator
- Low Power Idle and Power Down Modes
- Programming Lock for Software Security

## Description

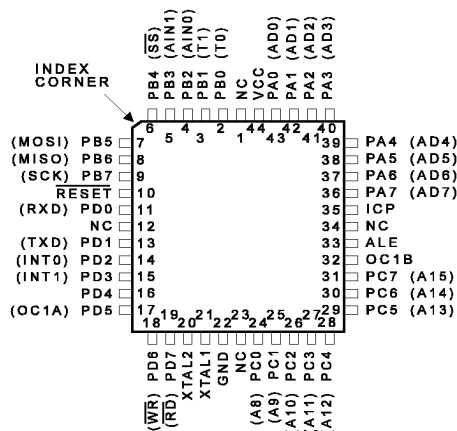
The AT90S8515 is a low-power CMOS 8-bit microcontroller based on the *AVR*<sup>®</sup> enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the AT90S8515 achieves throughputs approaching 1 MIPS per MHz allowing the system designer to optimize power consumption versus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

## Pin Configurations

PDIP



PLCC



8-Bit **AVR**<sup>®</sup>  
Microcontroller  
with 8K bytes  
Downloadable  
Flash

Preliminary

0841A



## Block Diagram

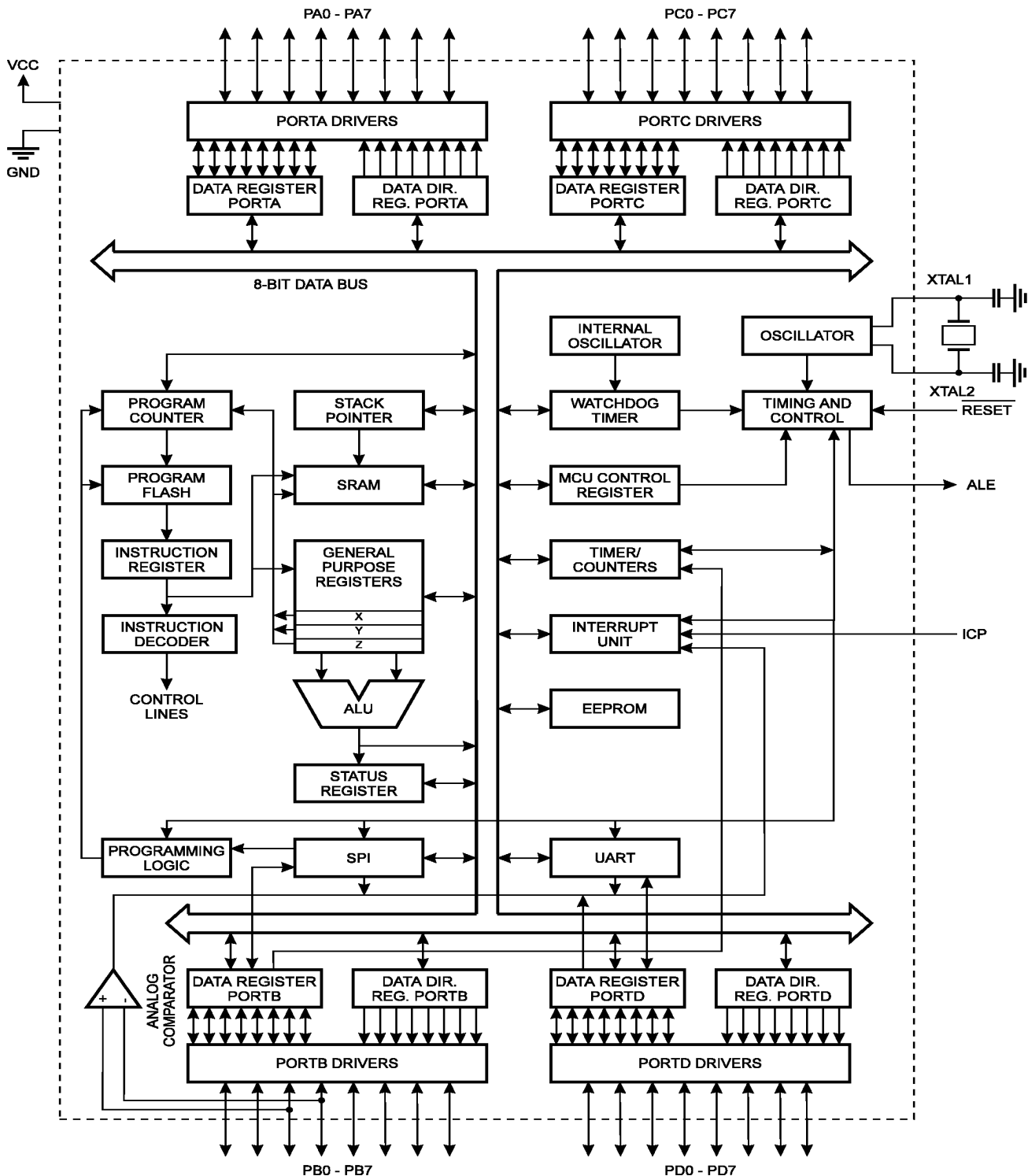


Figure 1. The AT90S8515 Block Diagram

## **Description (Continued)**

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 Downloadable 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 interrupt or hardware reset.

The device is manufactured using Atmel's high density non-volatile memory technology. The on-chip Downloadable 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 Downloadable 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 bidirectional I/O port. Port pins can provide internal pullups (selected for each bit). The Port A output buffers can sink 20mA and can drive LED displays directly. When pins PA0 to PA7 are used as inputs and are externally pulled low, they will source current ( $I_{IL}$ ) if the internal pullups are activated.

Port A serves as Multiplexed Address/Data input/output when using external SRAM.

### Port B (PB7..PB0)

Port B is an 8-bit bidirectional I/O pins with internal pullups. The Port B output buffers can sink 20 mA. As inputs, Port B pins that are externally pulled low will source current ( $I_{IL}$ ) if the pullups are activated.

Port B also serves the functions of various special features of the AT90S8515 as listed on Page 5-62.

### Port C (PC7..PC0)

Port C is an 8-bit bidirectional I/O port with internal pullups. The Port C output buffers can sink 20 mA. As inputs, Port C pins that are externally pulled low will source current ( $I_{IL}$ ) if the pullups are activated.

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

### Port D (PD7..PD0)

Port D is an 8-bit bidirectional I/O port with internal pullups. The Port D output buffers can sink 20 mA. As inputs, Port D pins that are externally pulled low will source current ( $I_{IL}$ ) if the pullups are activated.

Port D also serves the functions of various special features of the AT90S8515 as listed on Page 5-68.

### RESET

Reset input. A low on this pin for two machine cycles while the oscillator is running resets the device.

### 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 which can be configured for use as an on-chip oscillator, as shown in Figure 2. Either a quartz crystal or a ceramic resonator may be used. To drive the device from an external clock source, XTAL2 should be left unconnected while XTAL1 is driven as shown in Figure 3.

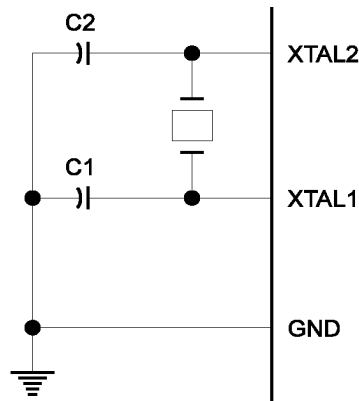


Figure 2. Oscillator Connections

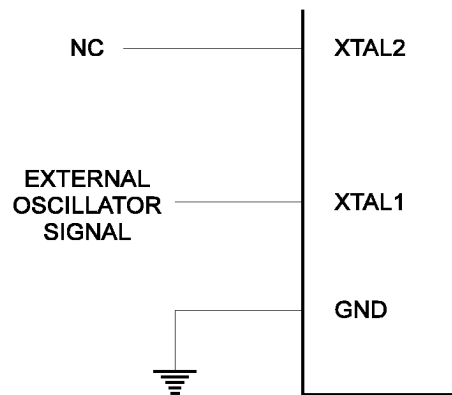


Figure 3. External Clock Drive Configuration

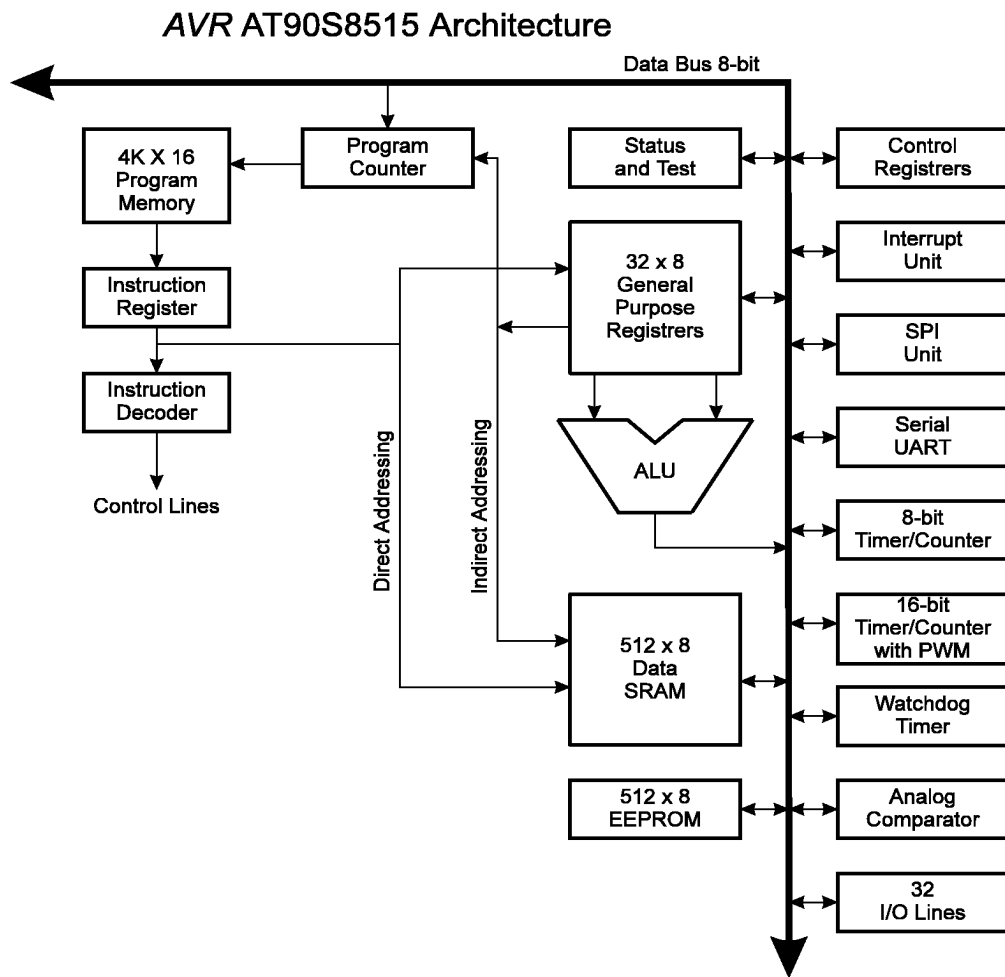
## AT90S8515 AVR Enhanced RISC Microcontroller CPU

The AT90S8515 AVR RISC microcontroller is upward compatible with the AVR Enhanced RISC Architecture. The programs written for the AT90S8515 MCU are fully compatible with the range of AVR 8-bit MCUs (AT90Sxxxx) with respect to source code and clock cycles for execution.

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



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

The ALU supports arithmetic and logic functions between registers or between a constant and a register. Single register operations are also executed in the ALU. Figure 4 shows the AT90S8515 AVR Enhanced 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 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.

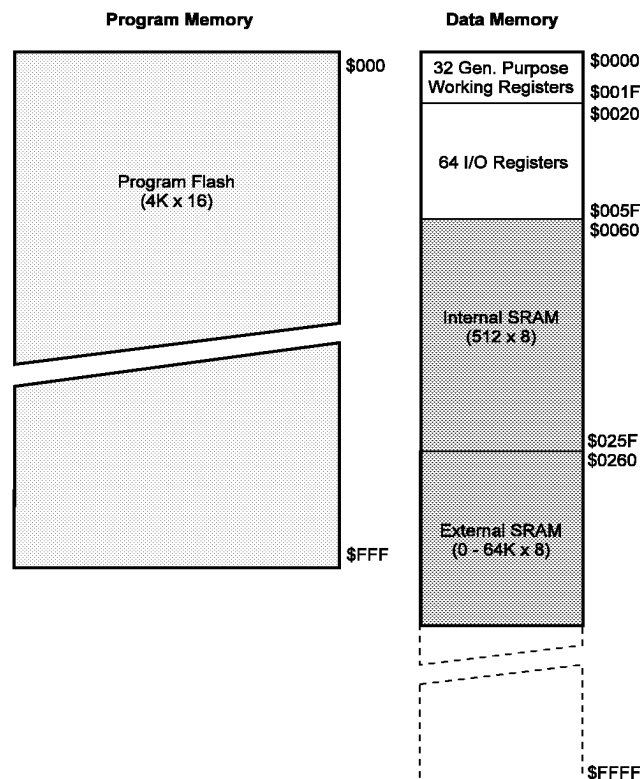
The AVR uses a Harvard architecture concept - with separate memories and buses for program and data. The program memory is executed with a single level pipelining. While one instruction is being executed, the next instruction is pre-fetched from the program memory. This concept enables instructions to be executed in every clock cycle. The program memory is in-system downloadable 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 bytes 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 5. Memory Maps**

A flexible interrupt module has its control registers in the I/O space with an additional global interrupt enable bit in the status register. All the different interrupts have a separate interrupt vector in the interrupt vector table at the beginning of the program memory. The different interrupts have priority in accordance with their interrupt vector position. The lower the interrupt address vector the higher priority.



## The ALU - Arithmetic Logic Unit

The high-performance *AVR* ALU operates in direct connection with all the 32 general purpose working registers. Within a single clock cycle, ALU operations between registers in the register file are executed. The ALU operations are divided into three main categories - arithmetic, logic and bit-functions. Some microcontrollers in the *AVR* product family feature a hardware multiplier in the arithmetic part of the ALU.

## The Downloadable Flash Program Memory

The AT90S8515 contains 8K bytes on-chip downloadable 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.

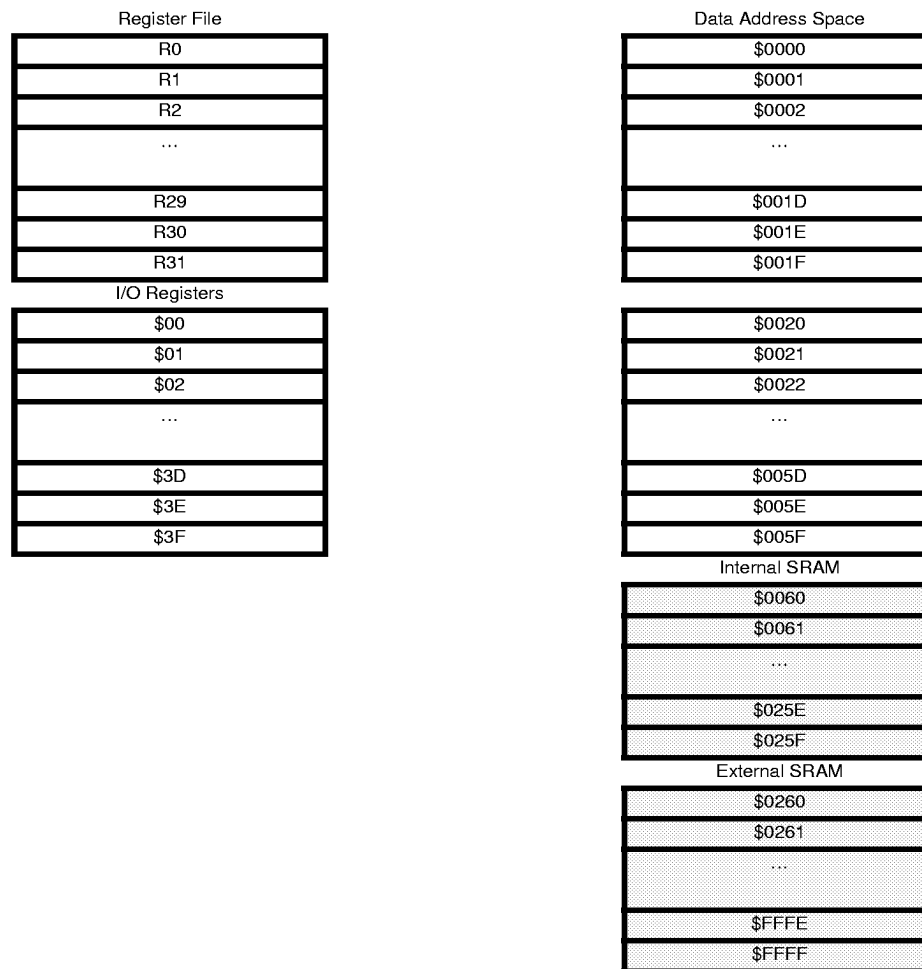
See Page 5-78 for a detailed description on Flash data downloading.

Constant tables must be allocated within the address 0-4K (see the LPM - Load Program Memory instruction description).

See Page 5-14 for the different program memory addressing modes.

## The SRAM Data Memory - Internal and External

The following figure 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 exceeds 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 (RD and WR) are inactive during the whole access cycle. External SRAM operation is enabled by setting the SRE bit in the MCUCR register. See Page 5-31 for details.

Accessing external SRAM takes one additional clock cycle per byte compared to the internal SRAM. This applies to commands LD, ST, LDS, STS, PUSH, POP. If the stack is placed in the external SRAM, interrupts, subroutine calls and returns will require two more clock cycles. When the external SRAM is used with wait state, all external SRAM access takes four clock cycles extra.

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 a 63 address locations reach 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.

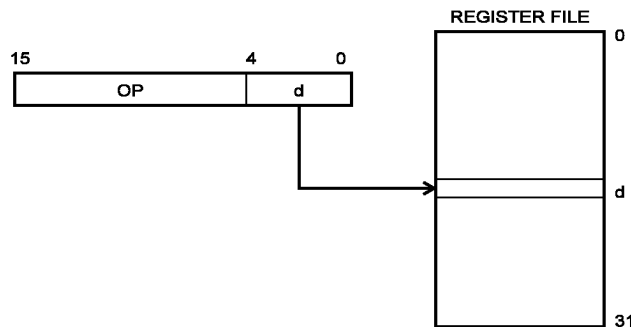
The 32 general purpose working registers, 64 I/O registers, the 512 bytes of internal data SRAM, and the 64K bytes of optional external data SRAM in the AT90S8515 are all accessible through all these addressing modes.

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

## The Program and Data Addressing Modes

The AT90S8515 AVR Enhanced 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.

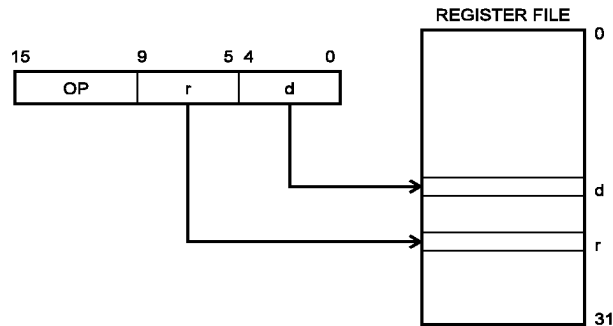
### REGISTER DIRECT, SINGLE REGISTER RD



**Figure 9.** Direct Single Register Addressing

The operand is contained in register d (Rd).

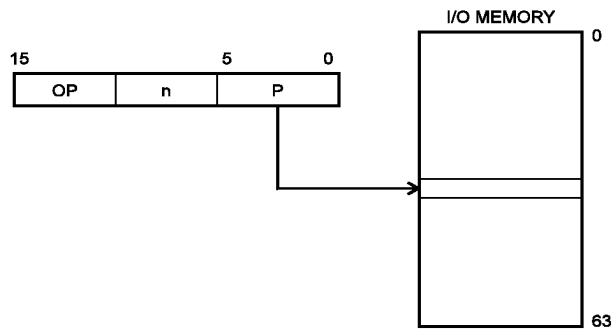
## REGISTER DIRECT, TWO REGISTERS RD AND RR



**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).

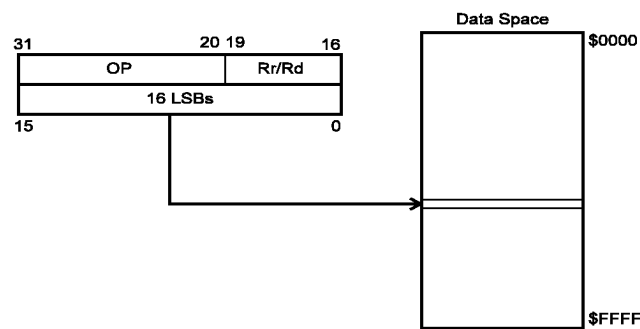
## I/O DIRECT



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

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

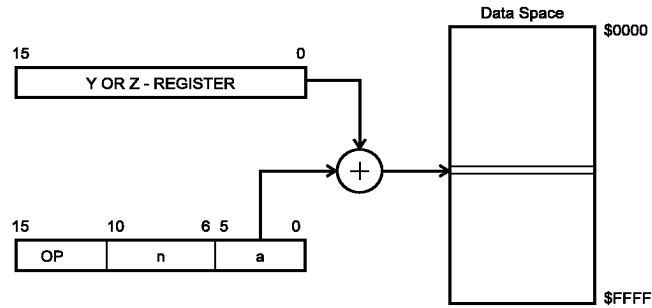
## DATA DIRECT



**Figure 12.** Direct Data Addressing

A 16-bit Data Address is contained in the 16 LSBs of a two-word instruction. Rd/Rr specify the destination or source register.

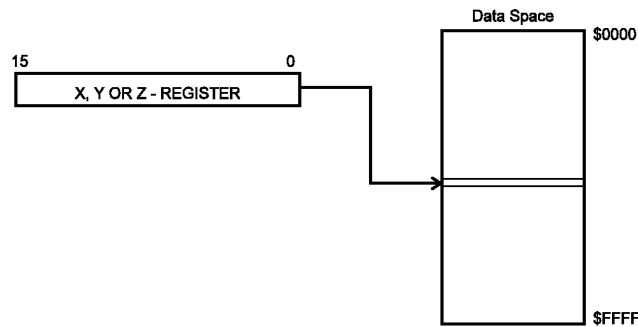
## DATA INDIRECT WITH DISPLACEMENT



**Figure 13.** Data Indirect with Displacement

Operand address is the result of the Y or Z-register contents added to the address contained in 6 bits of the instruction word.

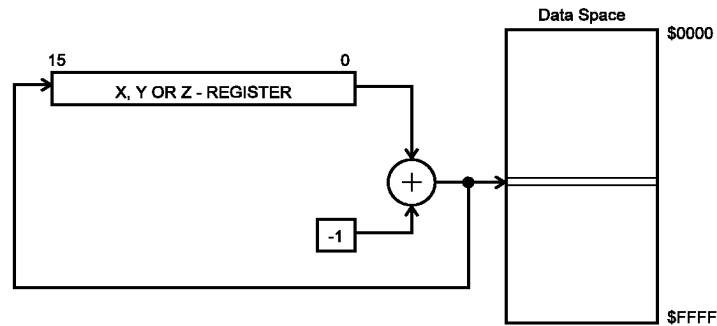
## DATA INDIRECT



**Figure 14.** Data Indirect Addressing

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

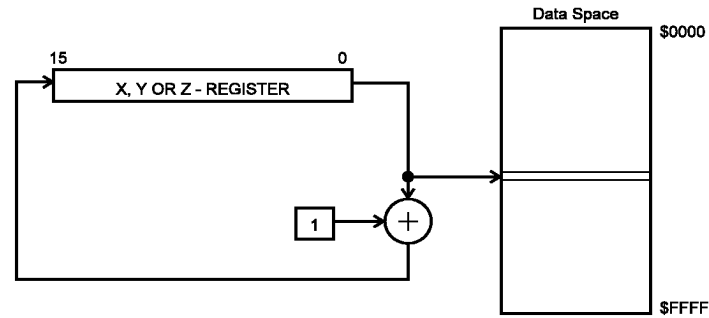
## DATA INDIRECT WITH PRE-DECREMENT



**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.

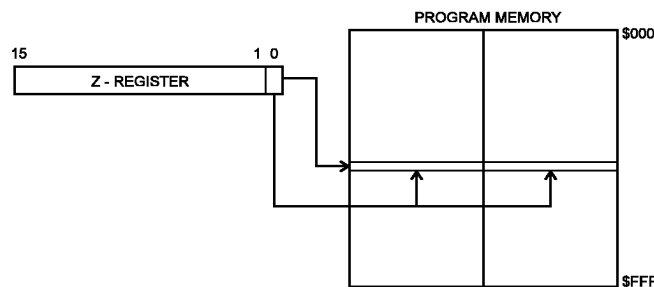
## DATA INDIRECT WITH POST-INCREMENT



**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.

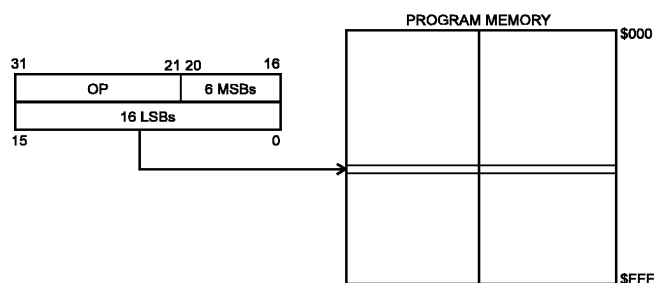
## CONSTANT ADDRESSING USING THE LPM INSTRUCTION



**Figure 17.** Code Memory Constant Addressing

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

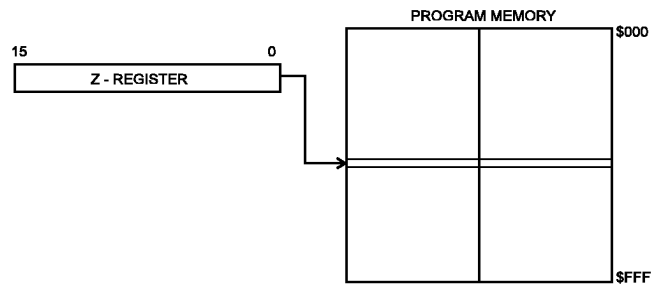
## DIRECT PROGRAM ADDRESS, JMP AND CALL



**Figure 18.** Direct Program Memory Addressing

Program execution continues at the address immediate in the instruction words.

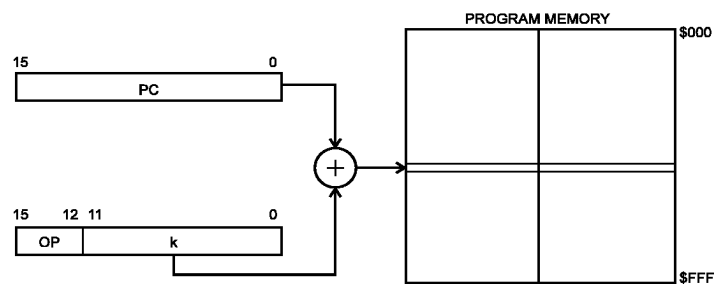
## INDIRECT PROGRAM ADDRESSING, IJMP AND ICALL



**Figure 19.** 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 20.** Relative Program Memory Addressing

Program execution continues at address  $PC + k$ . The relative address  $k$  is  $\pm 2K$ .

## The 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 5-44 specifying the EEPROM address registers, the EEPROM data register, and the EEPROM control register.

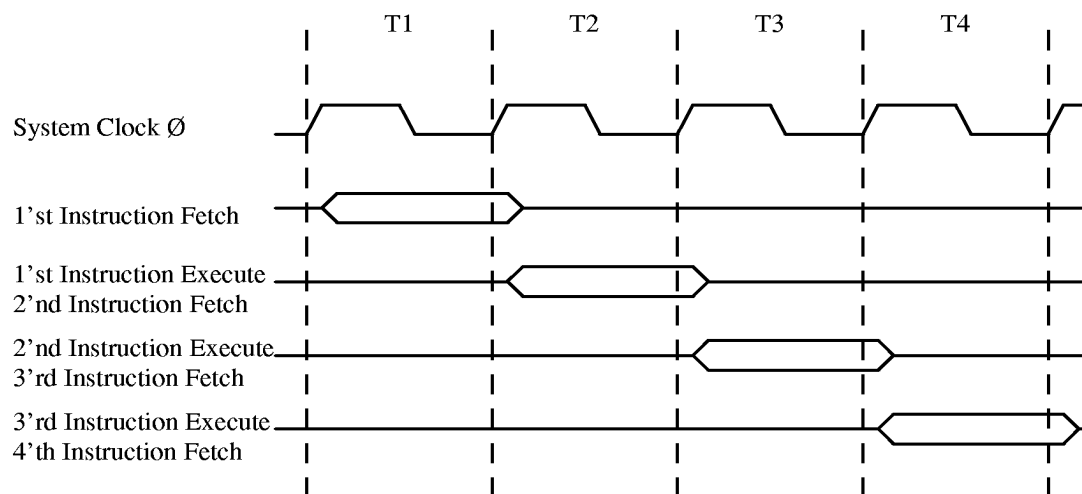
For the SPI data downloading, see Page 5-78 for a detailed description.

## Memory Access Times and Instruction Execution Timing

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

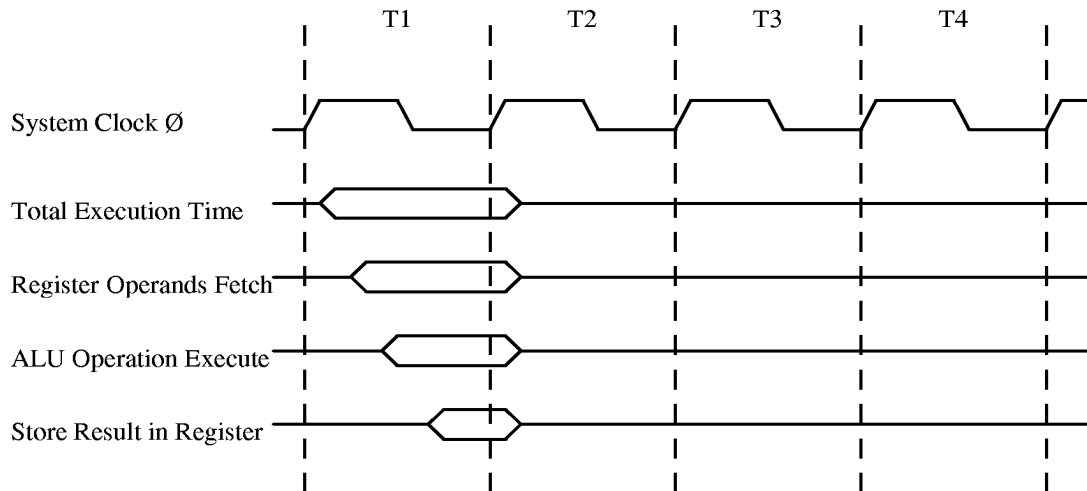
The AVR CPU is driven by the System Clock  $\phi$ , directly generated from the external clock crystal for the chip. No internal clock division is used.

Figure 21 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.



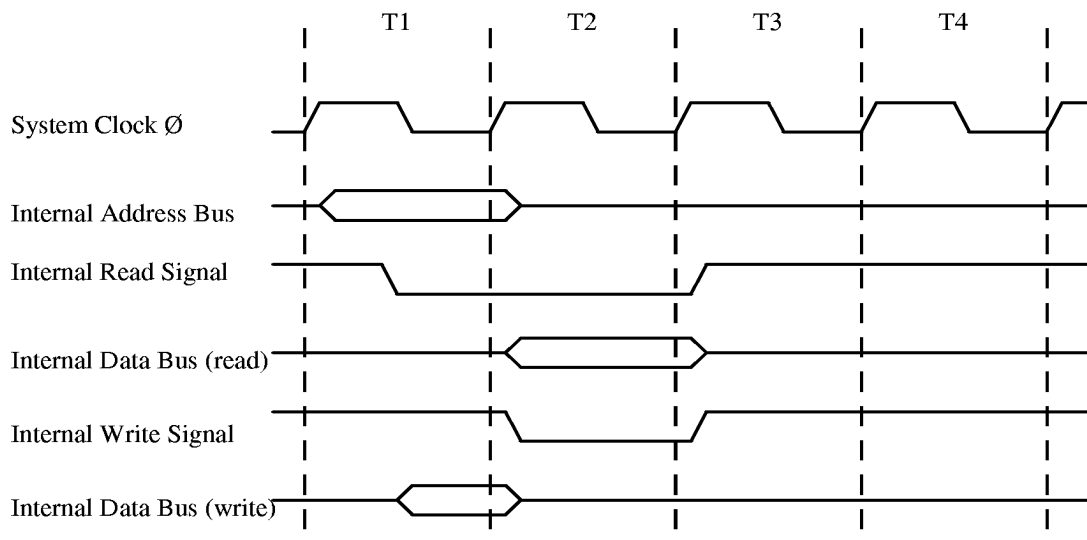
**Figure 21.** The Parallel Instruction Fetches and Instruction Executions

Figure 22 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.



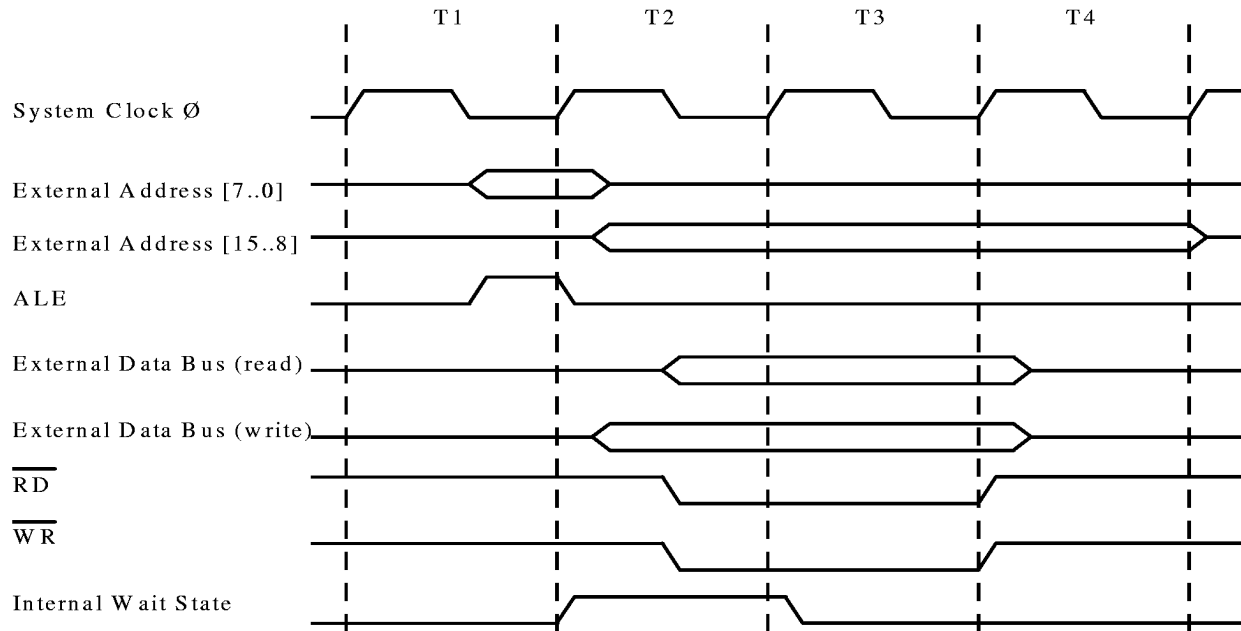
**Figure 22.** Single Cycle ALU Operation

The internal data SRAM access is performed in two System Clock cycles as described in Figure 23.



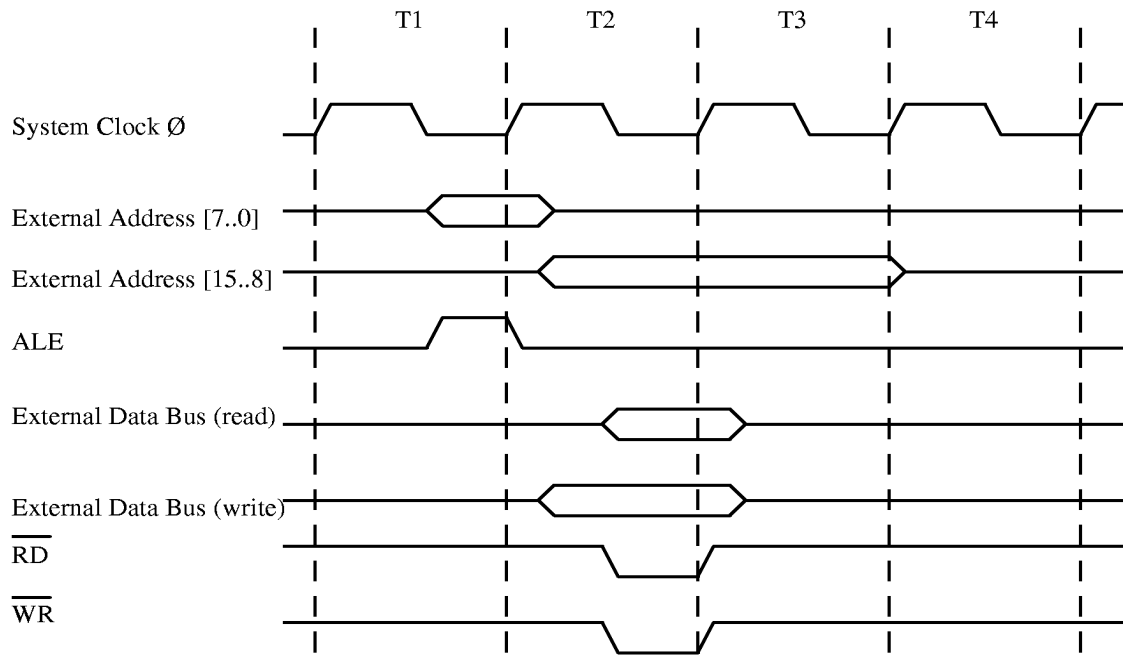
**Figure 23.** On-Chip Data SRAM Access Cycles

The external data SRAM access is performed in two System Clock cycles as described in Figure 23 .



**Figure 24.** External Data SRAM Memory Cycles without Wait State

The external data SRAM memory access cycle with the Wait State bit enabled (Wait State active) is shown in Figure 25.



**Figure 25.** External Data SRAM Memory Cycles with Wait State



## I/O Memory

The I/O space definition of the AT90S8515 is shown in the following table:

**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) | WDTCSR | Watchdog Timer Control Register                    |
| \$1F (\$3E) | EEARH  | EEPROM Address Register High Byte                  |
| \$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                              |
| \$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 the different AT90S8515 I/Os and peripherals are placed in the I/O space. The different I/O locations are accessed by the IN and OUT instructions transferring data between the 32 general purpose working registers and the I/O space. I/O registers within the address range \$00 - \$1F are directly bit-accessible using the SBI and CBI instructions. In these registers, the value of single bits can be checked by using the SBIS and SBIC instructions. Refer to the instruction set chapter for more details.

When using the I/O specific commands, IN, OUT, SBIS and SBIC, 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.

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

## THE 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 the interrupt mask registers - GIMSK and TIMSK. If the global interrupt enable register is cleared (zero), none of the interrupts are enabled independent of the GIMSK and TIMSK values. The I-bit is cleared by hardware after an interrupt has occurred, and is set by the RETI instruction to enable subsequent interrupts.

### Bit 6 - T : Bit Copy Storage:

The bit copy instructions BLD (Bit Load) and BST (Bit Store) use the T bit as source and destination for the operated bit. A bit from a register in the register file can be copied into T by the BST instruction, and a bit in T can be copied into a bit in a register in the register file by the BLD instruction.

### Bit 5 - H : Half Carry Flag:

The half carry flag H indicates a half carry in some arithmetic operations. See the Instruction Set Description for detailed information.

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

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

### Bit 3 - V : Two's Complement Overflow Flag:

The two's complement overflow flag V supports two's complement 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.



## THE 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 is decremented by one when data is pushed onto the Stack with the PUSH instruction, and it is decremented by two when data is pushed onto the Stack with subroutine CALL and interrupt. The Stack Pointer is incremented by one when data is popped from the Stack with the POP instruction, and it is incremented by two when data is popped from the Stack with return from subroutine RET or return from interrupt IRET.

## 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 which 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 is 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        | Hardware Pin 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 Handle                |          |
| \$001   |        | rjmp EXT_INT0 ; IRQ0 Handle              |          |
| \$002   |        | rjmp EXT_INT1 ; IRQ1 Handle              |          |
| \$003   |        | rjmp TIM1_CAPT ; Timer1 capture Handle   |          |
| \$004   |        | rjmp TIM1_COMPA ; Timer1 compareA Handle |          |
| \$005   |        | rjmp TIM1_OVF ; Timer1 overflow Handle   |          |
| \$006   |        | rjmp TIM1_OVF ; Timer1 overflow Handle   |          |
| \$007   |        | rjmp TIM0_OVF ; Timer0 overflow Handle   |          |
| \$008   |        | rjmp SPI_HANDLE ; SPI TX Handle          |          |
| \$009   |        | rjmp UART_RXC ; UART RX Complete Handle  |          |
| \$00a   |        | rjmp UART_DRE ; UDR Empty Handle         |          |
| \$00b   |        | rjmp UART_TXC ; UART TX Complete Handle  |          |
| \$00c   |        | rjmp ANA_COMP ; Analog Comparator Handle |          |
| ;       |        |                                          |          |
| \$00d   | MAIN:  | <instr> xxx ; Main program start         |          |
| ...     | ...    | ...                                      | ...      |

## RESET SOURCES

The AT90S8515 has three sources of reset:

- Power-On Reset. The MCU is reset when a supply voltage is applied to the VCC and GND pins.
- External Reset. The MCU is reset when a low level is present on the RESET pin for more than two XTAL cycles
- Watchdog Reset. The MCU is reset when the Watchdog timer period expires and the Watchdog is enabled.

During reset, all I/O registers are then set to their initial values, and the program starts execution from address \$000. The instruction placed in address \$000 must be an RJMP - relative jump - instruction to the reset handling routine. If the program never enables an interrupt source, the interrupt vectors are not used, and regular program code can be placed at these locations. The circuit diagram in Figure 26 shows the reset logic. Table 3 defines the timing and electrical parameters of the reset circuitry.

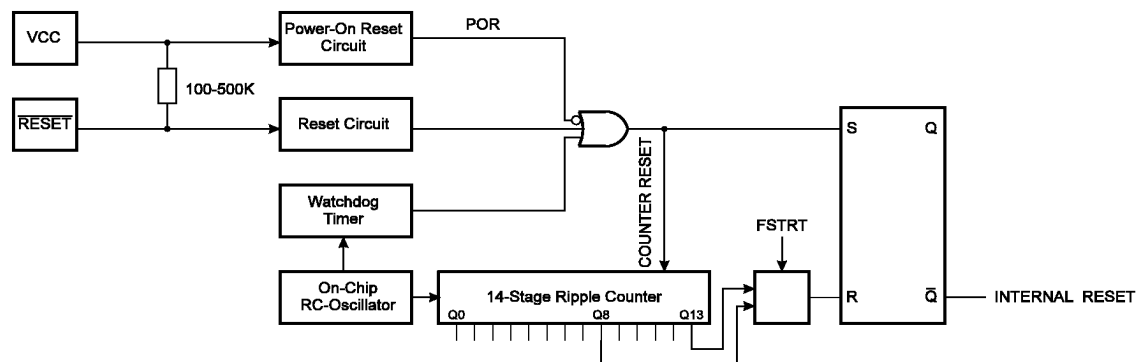


Figure 26. Reset Logic



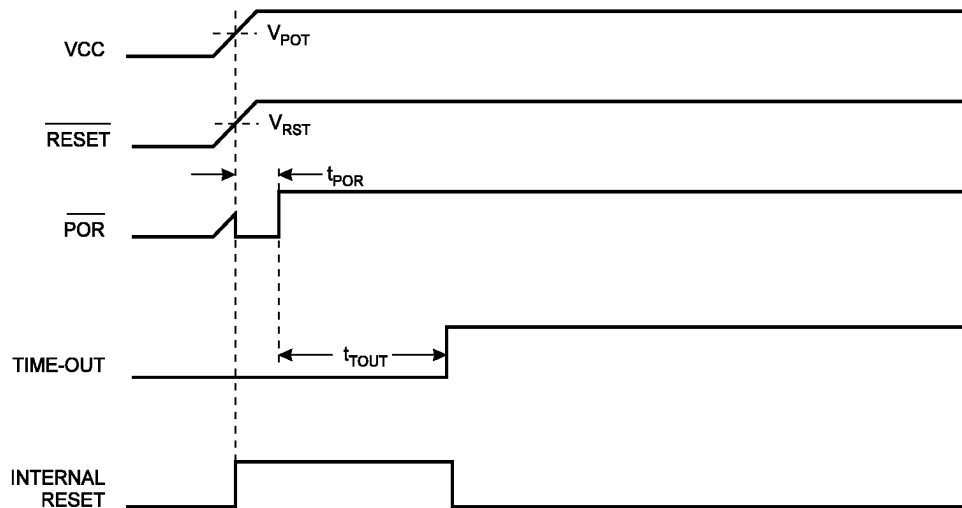
**Table 3.** Reset Characteristics ( $V_{CC} = 5.0V$ )

| Symbol     | Parameter                                         | Min | Typ        | Max | Units |
|------------|---------------------------------------------------|-----|------------|-----|-------|
| $V_{POT}$  | Power-On Reset Threshold Voltage                  | 1.8 | 2          | 2.2 | V     |
| $V_{RST}$  | RESET Pin Threshold Voltage                       |     | $V_{CC}/2$ |     | V     |
| $t_{POR}$  | Power-On Reset Period                             | 2   | 3          | 4   | ms    |
| $t_{TOUT}$ | Reset Delay Time-Out Period<br>FSTRT Unprogrammed | 11  | 16         | 21  | ms    |
| $t_{TOUT}$ | Reset Delay Time-Out Period<br>FSTRT Programmed   | 1.0 | 1.1        | 1.2 | ms    |

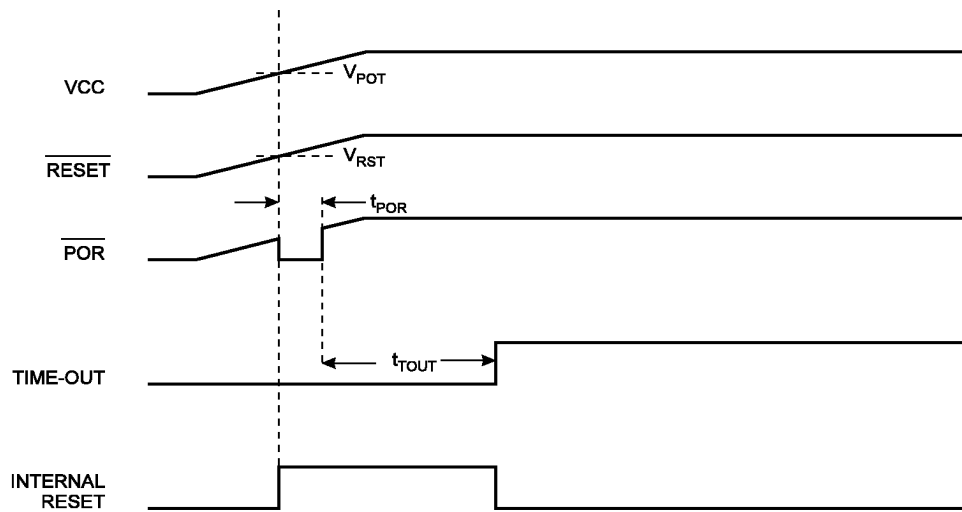
### POWER-ON RESET

A Power-On Reset (POR) circuit ensures that the device is not started until  $V_{CC}$  has reached a safe level. As shown in Figure 26, 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 27 and Figure 28). The total reset period is the Power-On Reset period -  $t_{POR}$  + the Delay Time-out period -  $t_{TOUT}$ . 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.

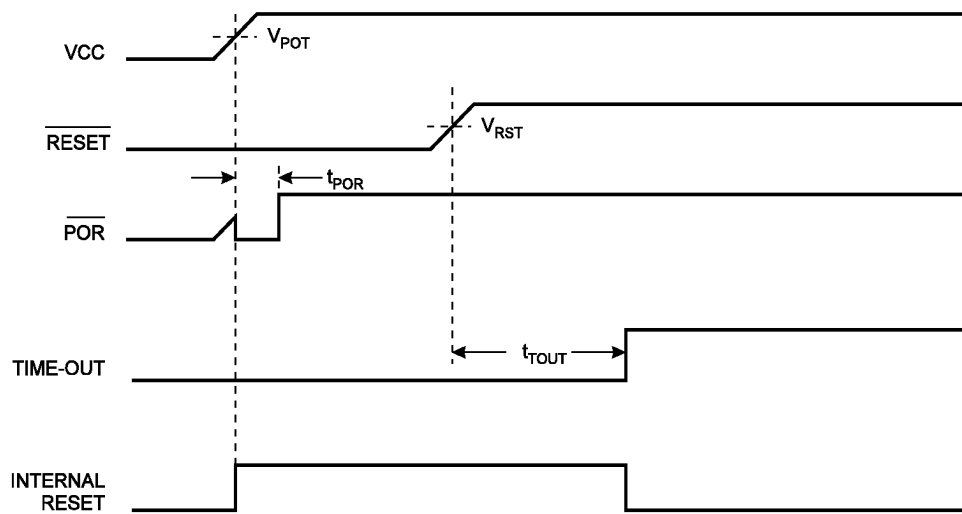
As the pin is pulled high by an on-chip resistor, the pin can be left unconnected if no external reset is required. Connecting to  $V_{CC}$  will have the same effect. By holding the pin low for a period after  $V_{CC}$  has been applied, the Power-On Reset period can be extended. Refer to Figure 29 for a timing example on this.



**Figure 27.** MCU Start-Up, RESET Tied to  $V_{CC}$  or Unconnected. Rapidly Rising  $V_{CC}$



**Figure 28.** MCU Start-Up, RESET Tied to V<sub>CC</sub> or Unconnected. Slowly Rising V<sub>CC</sub>

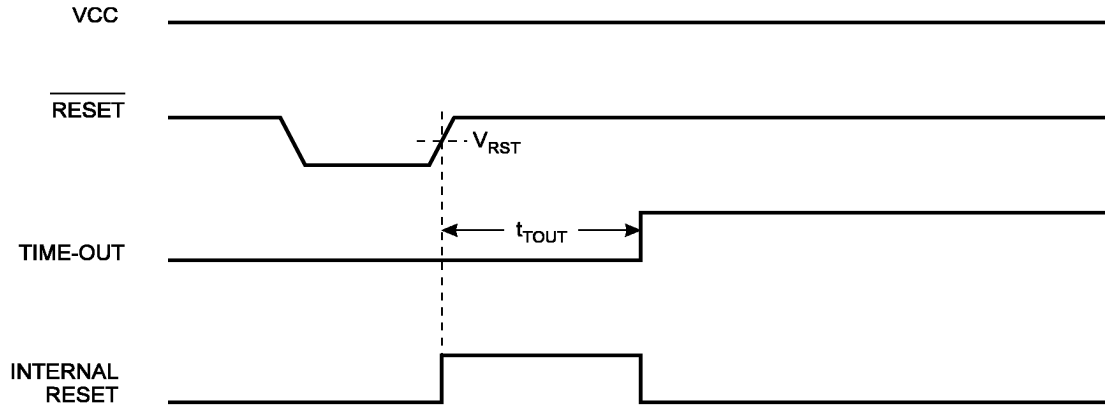


**Figure 29.** MCU Start-Up, RESET Controlled Externally



## EXTERNAL RESET

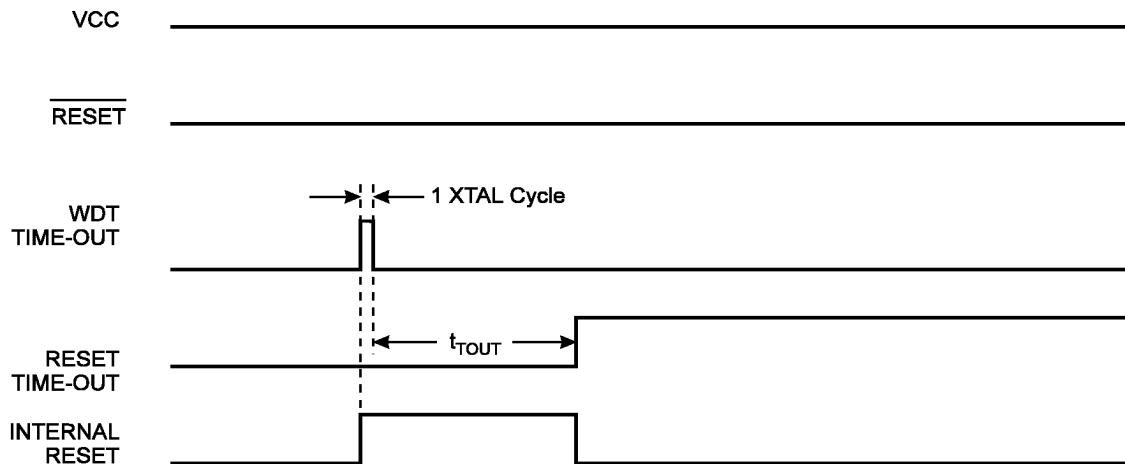
An external reset is generated by a low level on the **RESET** pin. The **RESET** pin must be held low for at least two crystal clock cycles. When reaches the Reset Threshold Voltage -  $V_{RST}$  on its positive edge, the delay timer starts the MCU after the Time-out period  $t_{TOUT}$  has expired.



**Figure 30.** 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 5-43 for details on operation of the Watchdog.



**Figure 31.** 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 must set (one) the I-bit to enable interrupts.

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

## THE 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 activated. The Interrupt Sense Control1 bits 1/0 (ISC11 and ISC10) in the MCU general Control Register (MCUCR) defines whether the external interrupt is activated on rising or falling edge of the INT1 pin or level sensed. Activity on the pin will cause an interrupt request even if INT1 is configured as an output. The corresponding interrupt of External Interrupt Request 1 is executed from 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 activated. The Interrupt Sense Control0 bits 1/0 (ISC01 and ISC00) in the MCU general Control Register (MCUCR) defines whether the external interrupt is activated on rising or falling edge of the INT0 pin or level sensed. Activity on the pin will cause an interrupt request even if INT0 is configured as an output. The corresponding interrupt of External Interrupt Request 0 is executed from 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.

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

### Bit 6 - INTF0 : External Interrupt Flag0:

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

### Bits 5..0 - Res : Reserved bits:

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

## THE 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. The Overflow Flag (Timer/Counter1) is set (one) in the Timer/Counter Interrupt Flag Register - TIFR. When Timer/Counter1 is in PWM mode, the Timer Overflow flag is set when the counter changes counting direction at \$0000.



#### **Bit 6 - OCIE1A :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. The CompareA Flag in Timer/Counter1 is set (one) 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. The CompareB Flag in Timer/Counter1 is set (one) 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. The Input Capture Flag in Timer/Counter1 is set (one) 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 \$008) is executed if an overflow in Timer/Counter0 occurs. The Overflow Flag (Timer0) is set (one) 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.

### **THE 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 logic one to the flag. When the I-bit in SREG, and 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 logic one to the flag. When the I-bit in SREG, and OCIE1A (Timer/Counter1 Compare match InterruptA Enable), and the OCF1A are set (one), the Timer/Counter1 Compare 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 logic one to the flag. When the I-bit in SREG, and OCIE1B (Timer/Counter1 Compare match InterruptB Enable), and the OCF1B are set (one), the Timer/Counter1 Compare 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 logic one to the flag.

## Bit 2 - Res : Reserved bit:

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

## Bit 1 - TOV0 : Timer/Counter0 Overflow Flag:

The bit TOV0 is set (one) when an overflow occurs in Timer/Counter0. TOV0 is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, TOV0 is cleared by writing a logic one to the flag. When the SREG I-bit, and 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 4 clock cycles minimum. After the 4 clock cycles 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 a relative jump to the interrupt routine, and this jump takes 2 clock cycles. If an interrupt occurs during execution of a multi-cycle instruction, this instruction is completed before the interrupt is served.

A return from an interrupt handling routine (same as for a subroutine call routine) takes 4 clock cycles. During these 4 clock cycles, the Program Counter (2 bytes) is popped back from the Stack, and the Stack Pointer is incremented by 2. When AVR exits from an interrupt, it will always return to the main program and execute one more instruction before any pending interrupt is served.

Note that the Status Register - SREG - is not handled by the AVR hardware, neither for interrupts nor for 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.

## 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), WR and 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 "The SRAM Data Memory - Internal and External" for 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 two cycle scheme. See Figure 24 External Data SRAM Memory Cycles without Wait State and Figure 25: External Data SRAM Memory Cycles with Wait State.

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

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

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

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

#### **Bit 3, 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 is set. The level and edges on the external INT1 pin that activate the interrupt are defined in the following table:

**Table 4.** 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.  |

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

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

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

**Table 5.** 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.  |

Note: When changing the ISC10/ISC00 bits, INT0 must be disabled by clearing its Interrupt Enable bit in the GIMSK Register. Otherwise an interrupt can occur when the bits are changed.

## **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.

Note that if a *level* triggered interrupt is used for wake-up from power down, the low level must be held for a time longer than the oscillator start-up time of 16 ms. Otherwise, the interrupt flag may return to zero before the MCU starts executing.



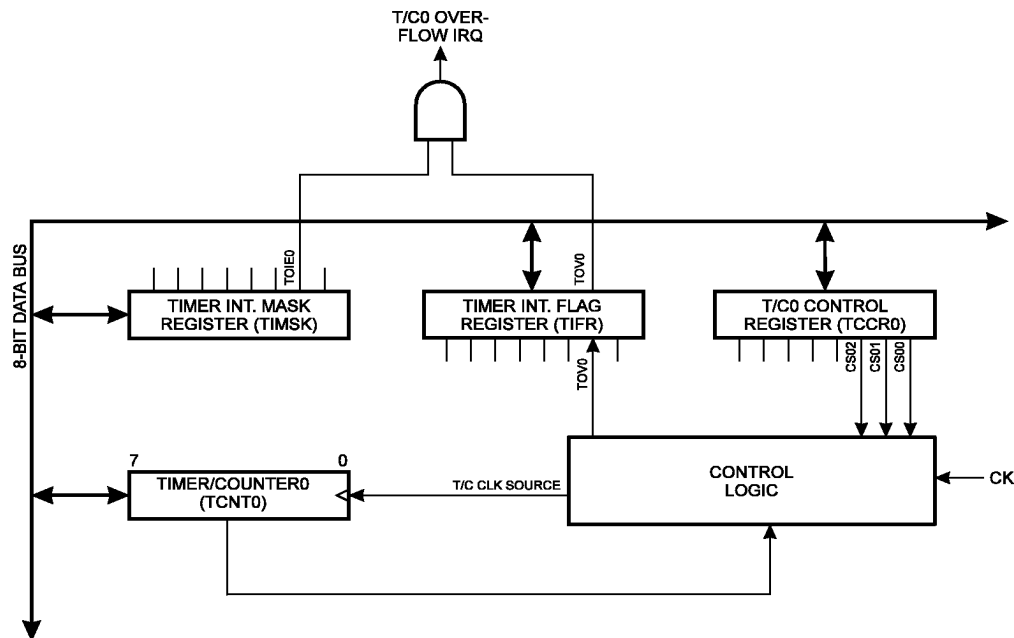
## The 8-Bit Timer/Counter0

Figure 33 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 Interrupt 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 33.** Timer/Counter0 Block Diagram

### THE 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 7,6 - Res : Reserved bits:**

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

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

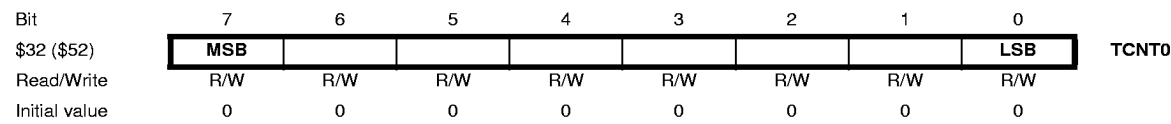
The Clock Select0 bits 2,1 and 0 define the prescaling source of Timer0.

Table 6. 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, the corresponding setup must be performed in the actual data direction control register (cleared to zero gives an input pin).

THE TIMER COUNTER 0 - TCNT0

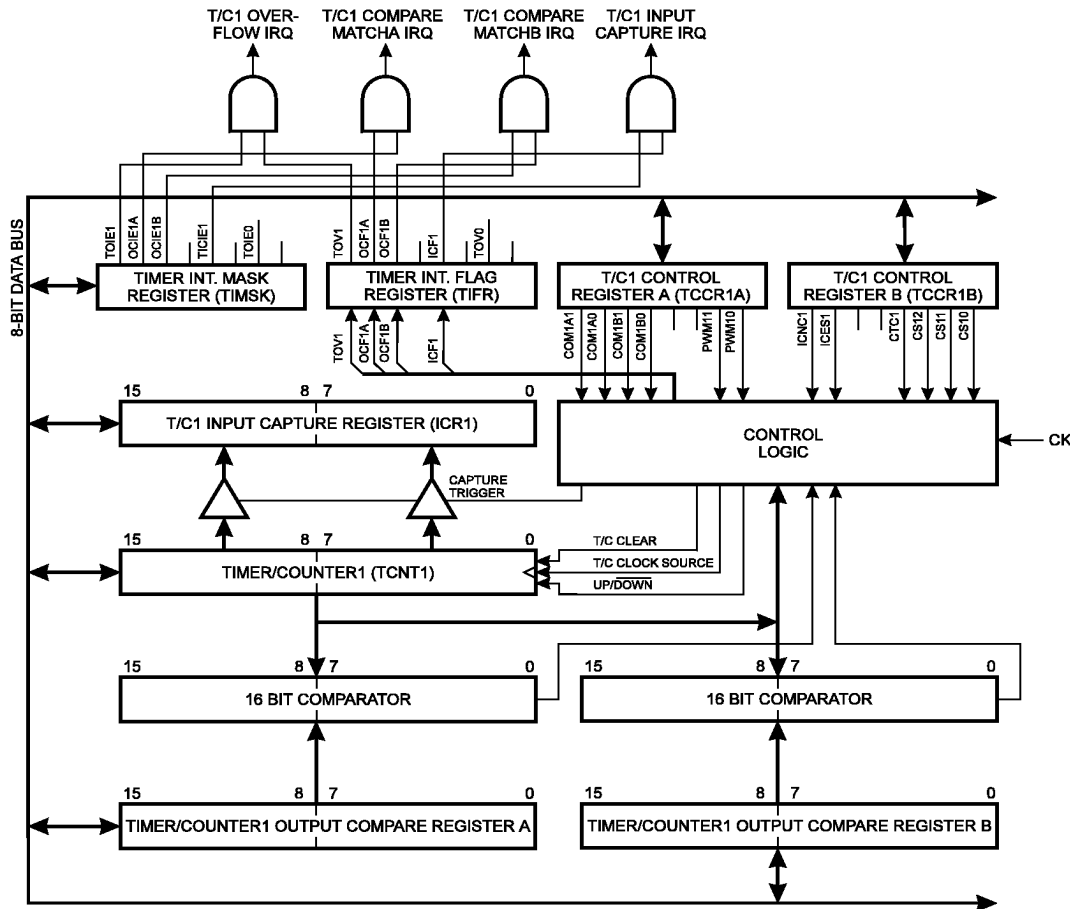


The Timer/Counter0 is realized as an up-counter with read and write access. If the Timer/Counter0 is written and a clock source is present, the Timer/Counter0 continues counting in the clock cycle following the write operation.



## The 16-Bit Timer/Counter1

Figure 34 shows the block diagram for Timer/Counter1.



**Figure 34.** 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) and 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 makes 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 a 8, 9 or 10-bit Pulse With Modulator. In this mode the counter and the OCR1A/OCR1B registers serve as a dual glitch-free stand-alone PWM with centered pulses. Refer to Page 5-41 for a detailed description on 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 the section, "The Analog Comparator", for details on this. The ICP pin logic is shown in Figure 35.

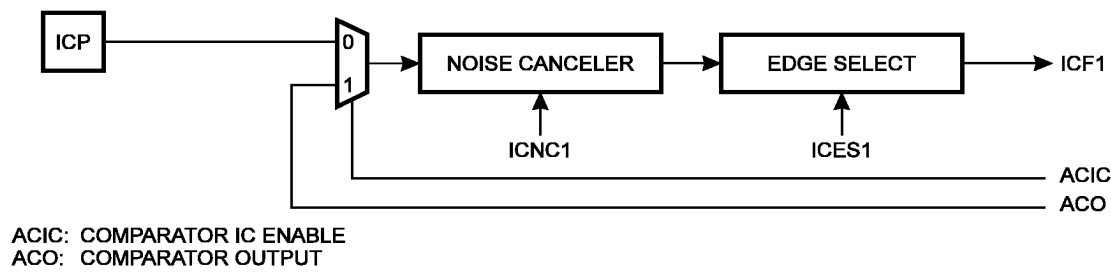


Figure 35. ICP Pin Schematic Diagram

The Timer/Counter1 input capture noise canceler block diagram is shown in Figure 36.

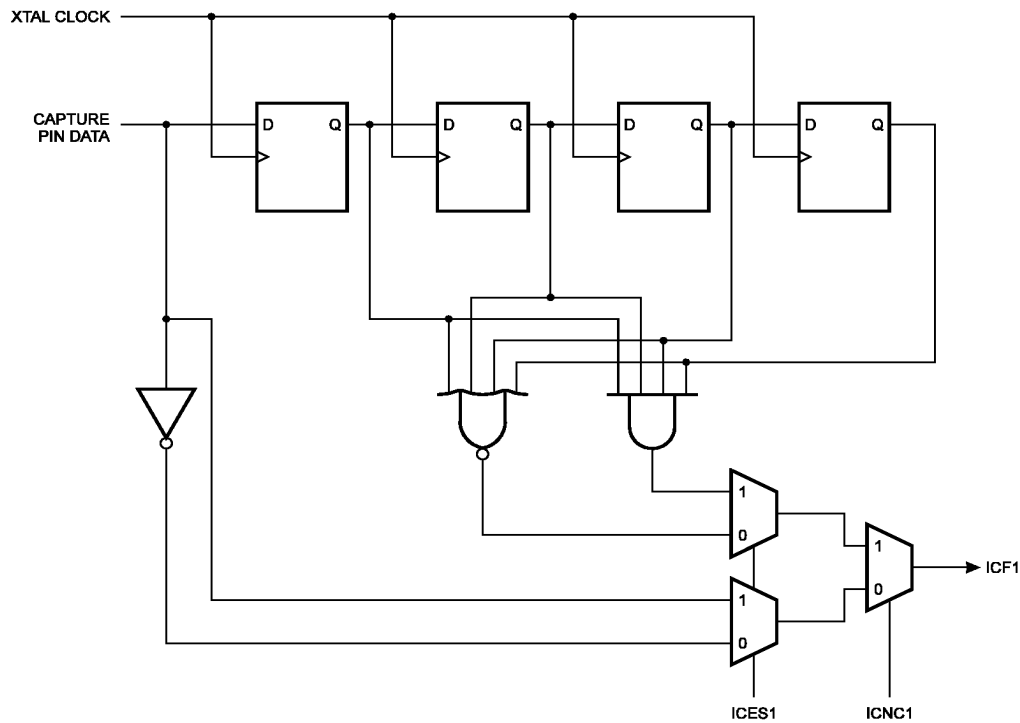


Figure 36. The Input Capture Noise Canceler

If the noise canceler function is enabled, the actual trigger condition for the capture event is monitored over 4 samples before the capture is activated. The input pin signal is sampled at XTAL clock frequency.

THE 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. Since this is an alternative function to an I/O port, the corresponding direction control bit must be set (one) to control an output pin. The control configuration is shown in Table 7.

#### 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. Since this is an alternative function to an I/O port, the corresponding direction control bit must be set (one) to control an output pin. The following control configuration is given:

**Table 7.** 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).               |

X = A or B

In PWM mode, these bits have a different function. Refer to Table 11 for a detailed description.

When changing the COM1X1/COM1X0 bits, Output Compare Interrupts 1 must be disabled by clearing their Interrupt Enable bits in the TIMSK Register. Otherwise an interrupt can occur when the bits are changed.

#### 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:

These bits select PWM operation of Timer/Counter1 as specified in Table 8. This mode is described on Page 5-41.

**Table 8.** 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              |

#### THE 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 measures on the ICP - input capture pin, 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 Input Capture Register - ICR1 - on the rising edge of the input capture pin - 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, the Timer/Counter1 continues counting until it is stopped, cleared, wraps around (overflow) or changes direction. In PWM mode, this bit has no effect.

## Bits 2,1,0 - CS12, CS11, CS10 : Clock Select1, bit 2,1 and 0:

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

**Table 9.** 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, the corresponding setup must be performed in the actual direction control register (cleared to zero gives an input pin).

## THE 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).

### • 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)   | <b>MSB</b> |     |     |     |     |     |     |            | <b>OCR1AH</b><br><b>OCR1AL</b> |
| \$2A (\$4A)   |            |     |     |     |     |     |     | <b>LSB</b> |                                |
|               | 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)   | <b>MSB</b> |     |     |     |     |     |     |            | <b>OCR1BH</b><br><b>OCR1BL</b> |
| \$28 (\$48)   |            |     |     |     |     |     |     | <b>LSB</b> |                                |
|               | 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 register.

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 TIMER/COUNTER1 INPUT CAPTURE REGISTER - ICR1H AND ICR1L

| Bit           | 15         | 14 | 13 | 12 | 11 | 10 | 9 | 8          |                              |
|---------------|------------|----|----|----|----|----|---|------------|------------------------------|
| \$25 (\$45)   | <b>MSB</b> |    |    |    |    |    |   |            | <b>ICR1H</b><br><b>ICR1L</b> |
| \$24 (\$44)   |            |    |    |    |    |    |   | <b>LSB</b> |                              |
|               | 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.

## TIMER/COUNTER1 IN PWM MODE

When the PWM mode is selected, Timer/Counter1 and 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 10) , when 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 11 for details.

**Table 10.** Timer TOP Values and PWM Frequency

| PWM Resolution | Timer TOP value | Frequency      |
|----------------|-----------------|----------------|
| 8-bit          | \$00FF (255)    | $f_{TC1}/510$  |
| 9-bit          | \$01FF (511)    | $f_{TC1}/1022$ |
| 10-bit         | \$03FF(1023)    | $f_{TC1}/2046$ |

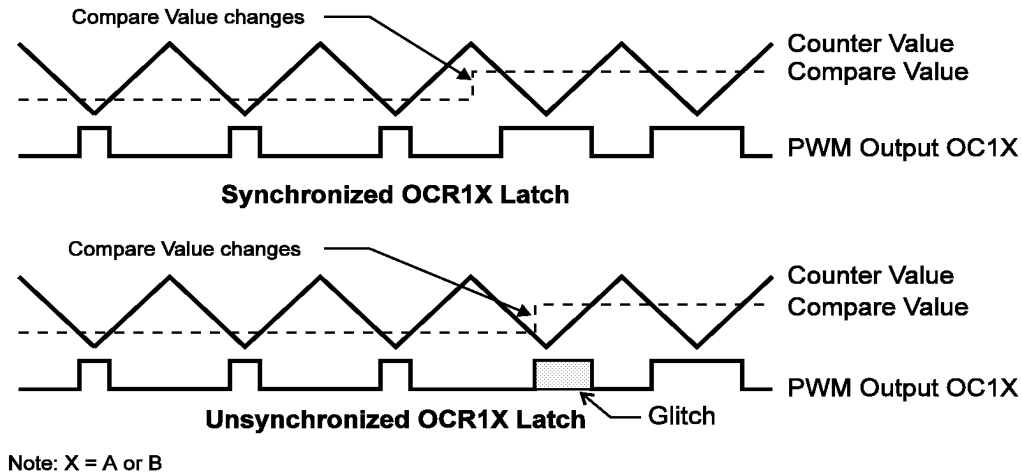
**Table 11.** Compare1 Mode Select in PWM Mode

| COM1X1 | COM1X0 | Effect on OCX1                                                                               |
|--------|--------|----------------------------------------------------------------------------------------------|
| 0      | 0      | Not connected                                                                                |
| 0      | 1      | Not connected                                                                                |
| 1      | 0      | Cleared on compare match, upcounting. Set on compare match, downcounting (non-inverted PWM). |
| 1      | 1      | Cleared on compare match, downcounting. Set on compare match, upcounting (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 37 for an example.



**Figure 37.** Effects on Unsynchronized OCR1 Latching

When OCR1 contains \$0000 or TOP, the output OC1A/OC1B is held low or high according to the settings of COM1A1/COM1A0 or COM1B1/COM1B0. This is shown in Table 12:

**Table 12.** 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 changes direction at \$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 does also apply to the Timer Output Compare1 flags and interrupts.

## The Watchdog Timer

The Watchdog Timer is clocked from a separate on-chip oscillator which runs at 1MHz. 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 from 16 to 2048 ms. 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 5-28.

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.

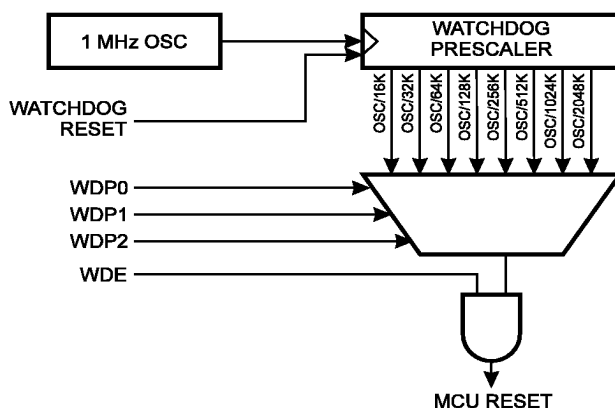


Figure 38. Watchdog Timer

### THE WATCHDOG TIMER CONTROL REGISTER - WDTCR

| Bit           | 7 | 6 | 5 | 4      | 3   | 2    | 1    | 0    |       |
|---------------|---|---|---|--------|-----|------|------|------|-------|
| \$21 (\$41)   | - | - | - | WDTTOE | 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 - WDTTOE : Watch Dog Turn-Off Enable:

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

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

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

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



#### Bits 2..0 - WDP2, WDP1, WDP0 : Watch Dog 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 Timeout Periods are shown in Table 13.

**Table 13.** Watch Dog Timer Prescale Select (Typical Values at  $V_{CC} = 5V$ )

| WDP2 | WDP1 | WDP0 | Timeout Period |
|------|------|------|----------------|
| 0    | 0    | 0    | 16 ms          |
| 0    | 0    | 1    | 32 ms          |
| 0    | 1    | 0    | 64 ms          |
| 0    | 1    | 1    | 128 ms         |
| 1    | 0    | 0    | 256 ms         |
| 1    | 0    | 1    | 512 ms         |
| 1    | 1    | 0    | 1024 ms        |
| 1    | 1    | 1    | 2048 ms        |

## 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 - 4ms, depending on the  $V_{CC}$  voltages. A self-timing function, however, lets the user software detect when the next byte can be written. An EEPROM brown-out detection prevents writing to the EEPROM if  $V_{CC}$  is below a certain level.

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

### THE EEPROM ADDRESS REGISTER - EEARH AND EEARL

|               |       |       |       |       |       |       |       |       |       |
|---------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| Bit           | 15    | 14    | 13    | 12    | 11    | 10    | 9     | 8     |       |
| \$1F (\$3F)   | -     | -     | -     | -     | -     | -     | -     | EEAR9 | EEARH |
| \$1E (\$3E)   | EEAR7 | EEAR6 | EEAR5 | EEAR4 | EEAR3 | EEAR2 | EEAR1 | EEAR0 | EEARL |
|               | 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 | 8     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |       |
|               | 8     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |       |

The EEPROM Address Registers - EEARH and EEARL specify the EEPROM address in the 512 bytes EEPROM space. The EEPROM data bytes are addressed linearly between 0 and 512.

### THE 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.

## THE 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    |      |

### **Bit 7..3 - Res : Reserved bits:**

These bits are reserved bits in the AT90S2313 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 one 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 one to the EEMWE bit in EECR
5. Within four clock cycles after setting EEMWE, write a logical one 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.

### **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 two 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.



## The Serial Peripheral Interface - SPI

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

- Full-Duplex, 3-Wire Synchronous Data Transfer
- Master or Slave Operation
- 5 Mbit/s Bit Frequency (max.)
- LSB First or MSB First Data Transfer
- Four Programmable Bit Rates
- End of Transmission Interrupt Flag
- Write Collision Flag Protection
- Wakeup from Idle Mode (Slave Mode Only)

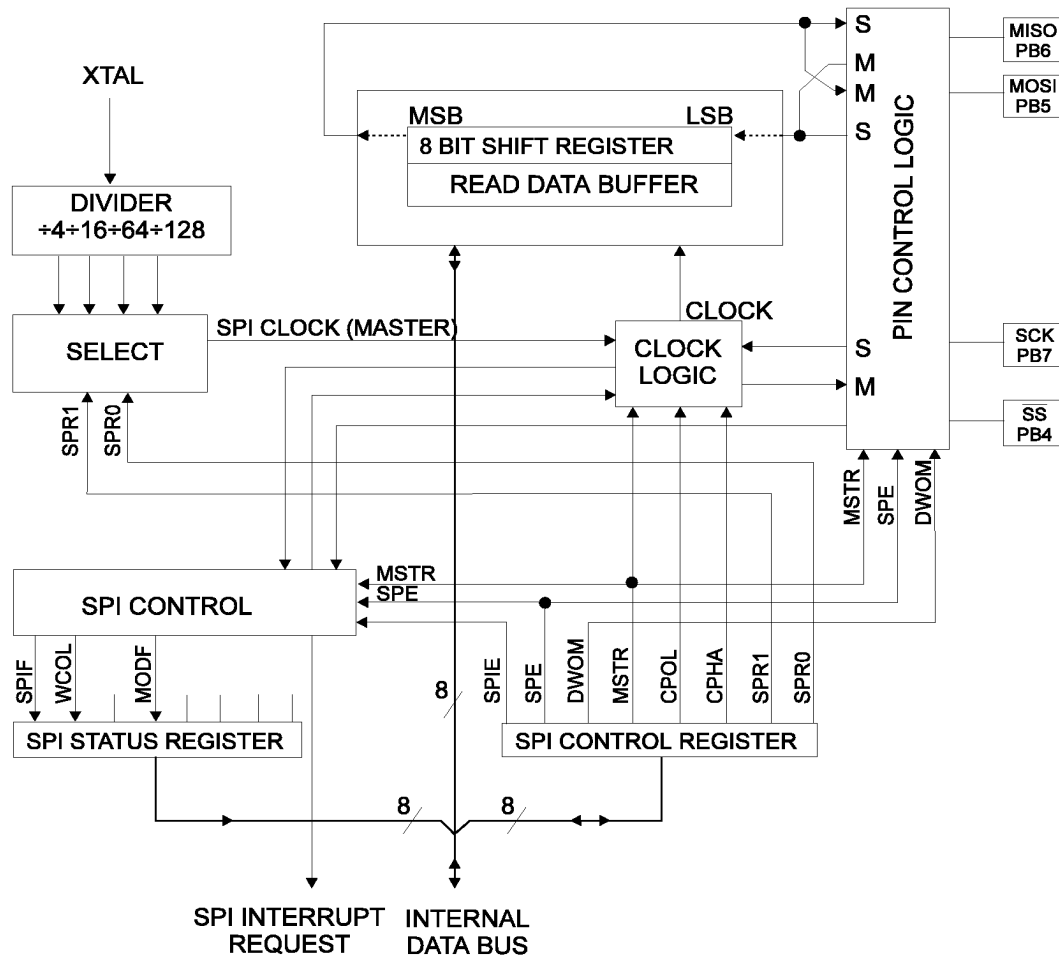


Figure 39. SPI Block Diagram

The interconnection between master and slave CPUs with SPI is shown in Figure 40. 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 becomes set, an interrupt is requested. The Slave Select input, PB4(SS), is set low to select an individual SPI device as a slave. 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 40. 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.

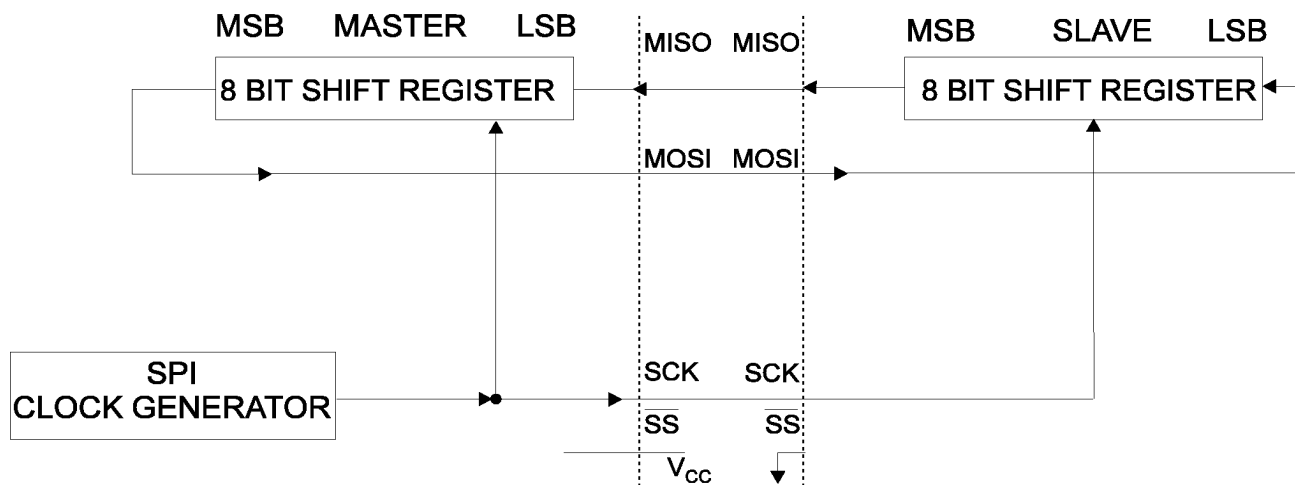


Figure 40. SPI Master-Slave Interconnection

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

When the SPI is enabled, the data direction of the MOSI, MISO, SCK and  $\overline{SS}$  pins is overridden according to the following table:

Table 14. 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                |

## 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, in master mode, the  $\overline{SS}$  pin is input, and is driven low by peripheral circuitry, the SPI system interprets this as that another master selects the SPI as a slave and will start sending 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, the interrupt routine will be executed.

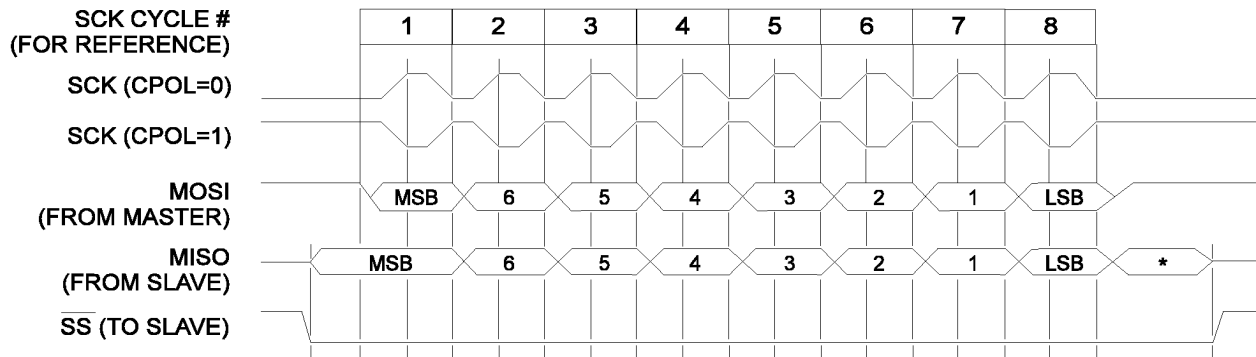


Thus, when interrupt-driven SPI transmittal is used in master mode, and there exists a possibility that  $\overline{SS}$  is driven low, the interrupt should always check that the MSTR bit is still set. Once the MSTR bit has been cleared by a slave select, it must be set by the user.

When the SPI is configured as a slave, the  $\overline{SS}$  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 low, all pins are inputs, and the SPI is passive, which means that it will not receive incoming data.

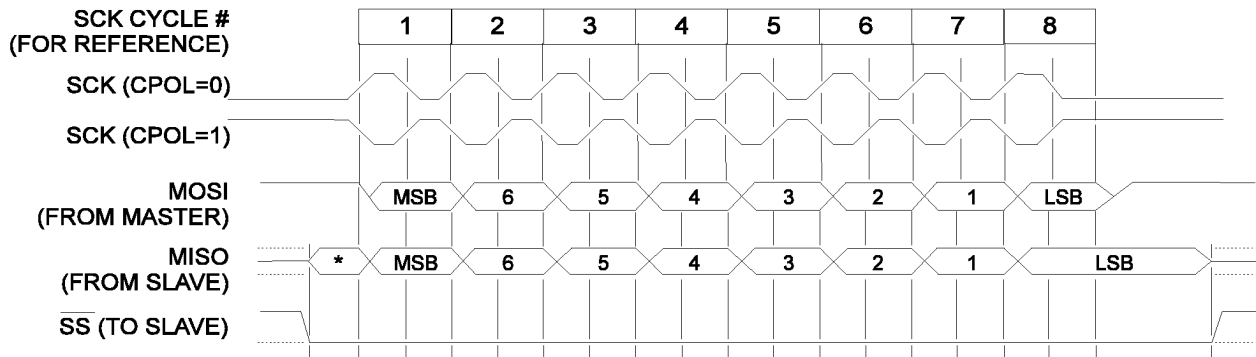
## Data Modes

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 41 and Figure 42.



\* Not defined but normally MSB of character just received

**Figure 41.** SPI Transfer Format with CPHA = 0



\* Not defined but normally LSB of previously transmitted character

**Figure 42.** SPI Transfer Format with CPHA = 1

## THE SPI CONTROL REGISTER - SPCR

| Bit           | 7           | 6          | 5           | 4           | 3           | 2           | 1           | 0           |      |
|---------------|-------------|------------|-------------|-------------|-------------|-------------|-------------|-------------|------|
| \$0D (\$2D)   | <b>SPIE</b> | <b>SPE</b> | <b>DORD</b> | <b>MSTR</b> | <b>CPOL</b> | <b>CPHA</b> | <b>SPR1</b> | <b>SPR0</b> | 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           | 1           | 0           | 0           |      |

### Bit 7 - SPIE : SPI Interrupt Enable:

This bit causes setting of the SPIF bit in the SPSR register to execute the SPI interrupt provided that 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 41 and Figure 42 for additional information.

### Bit 2 - CPHA : Clock PHase:

Refer to Figure 41 or Figure 42 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 SPR2 have no effect on the slave. The relationship between SCK and the Oscillator Clock frequency  $f_{cl}$  is shown in the following table:

**Table 15.** 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    | $f_{cl} / 128$ |

## THE SPI STATUS REGISTER - SPSR

| Bit           | 7           | 6           | 5 | 4 | 3 | 2 | 1 | 0 |      |
|---------------|-------------|-------------|---|---|---|---|---|---|------|
| \$0E (\$2E)   | <b>SPIF</b> | <b>WCOL</b> | - | - | - | - | - | - | 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 with SPIF set (one), then 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. During data transfer, the result of reading the SPDR register may be incorrect, and writing to it will have no effect. The WCOL bit (and the SPIF bit) are cleared (zero) by first reading the SPI Status Register with WCOL set (one), and then accessing the SPI Data Register.

#### **Bit 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 5-78 for serial programming and verification.

### **THE SPI DATA REGISTER - SPDR**

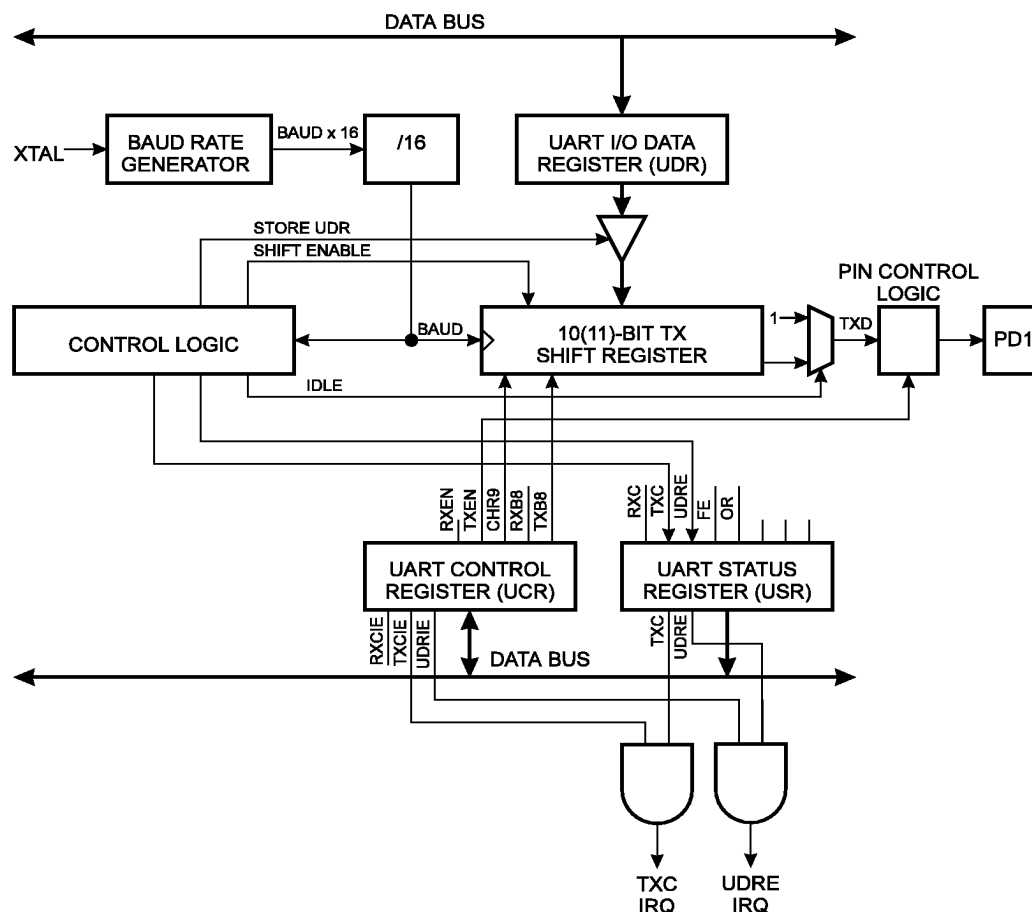
|               |     |     |     |     |     |     |     |     |      |
|---------------|-----|-----|-----|-----|-----|-----|-----|-----|------|
| Bit           | 7   | 6   | 5   | 4   | 3   | 2   | 1   | 0   |      |
| \$0F (\$2F)   | MSB |     |     |     |     |     |     | LSB | SPDR |
| Read/Write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

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.

## **The UART**

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

- Baud rate generator generates any baud rate
- 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



### Figure 43. 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.

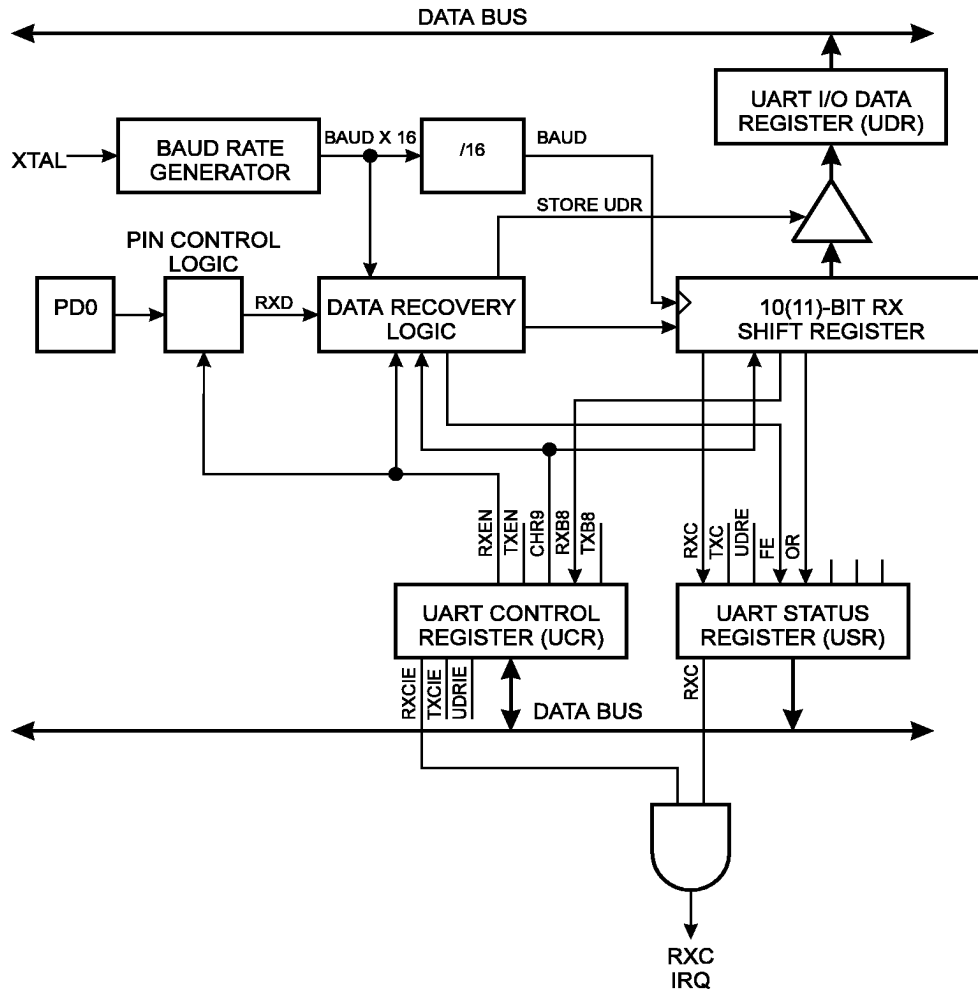
If the 10(11)-bit Transmitter shift register is empty or when, data is transferred from UDR to the shift register. At this time the UDRE (UART Data Register Empty) bit in the UART Status Register, USR, is set. When this bit is set (one), the UART is ready to receive the next character. At the same time 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). By clearing this bit (zero), the PD1 pin can be used for general I/O. When TXEN is set, the UART Transmitter will be connected to the PD1 pin regardless of the setting of the DDD1 bit in DDRB.

## Data Reception

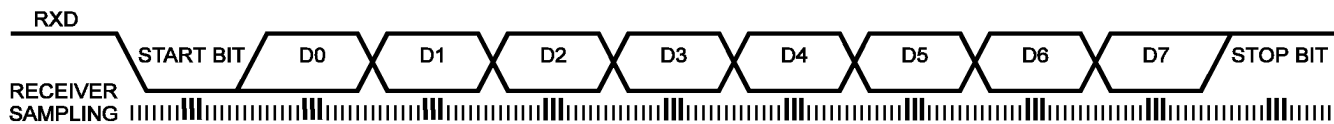
Figure 44 shows a block diagram of the UART Receiver



**Figure 44.** 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 zero 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 ones, 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 45.



**Figure 45.** Sampling Received Data

When the stop bit enters the receiver, the majority of the three samples must be one to accept the stop bit. If two or more samples are logical zeros, 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 UCR is set. This means that the last data byte shifted into to 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.

By clearing the RXEN bit in the UCR register, 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 the PD0 pin regardless of the setting of the DDD0 bit in DDRB.

## UART Control

### THE 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.

### THE 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   | 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 one 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.

### **THE UART CONTROL REGISTER - UCR**

| Bit           | 7            | 6            | 5            | 4           | 3           | 2           | 1           | 0           |     |
|---------------|--------------|--------------|--------------|-------------|-------------|-------------|-------------|-------------|-----|
| \$0A (\$2A)   | <b>RXCIE</b> | <b>TXCIE</b> | <b>UDRIE</b> | <b>RXEN</b> | <b>TXEN</b> | <b>CHR9</b> | <b>RXB8</b> | <b>TXB8</b> | 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           | 0           | 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 TXC, 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 bit long plus start and stop bits. The 9th bit is read and written by using the RXB8 and TXB8 bits in UCR, respectively. The 9th 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 9th data bit of the received character.

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

When CHR9 is set (one), TXB8 is the 9th data bit in the character to be transmitted.

## **THE BAUD RATE GENERATOR**

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

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

- BAUD = Baud-Rate
- fck= 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 16. UBRR values which yield an actual baud rate differing less than 2% from the target baud rate, are bolded in the table.



**Table 16.** 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    |
| 57600     | UBRR= 0  | 7.8    | UBRR= 1    | 0.0    | UBRR= 1  | 7.8    | UBRR= 2    | 12.5   |
| 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    |
| 57600     | UBRR= 3    | 12.5   | UBRR= 3    | 0.0    | UBRR= 3   | 7.8    | UBRR= 4   | 0.0    |
| 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    |
| 57600     | UBRR= 7    | 0.0    | UBRR= 8   | 3.7    | UBRR= 9   | 0.0    | UBRR= 11   | 0.0    |
| 115200    | UBRR= 3    | 0.0    | UBRR= 3   | 7.8    | UBRR= 4   | 0.0    | UBRR= 5    | 0.0    |

| Baud Rate | 14.746 MHz | %Error | 16 MHz    | %Error | 18.432 MHz | %Error | 20 MHz    | %Error |
|-----------|------------|--------|-----------|--------|------------|--------|-----------|--------|
| 2400      | UBRR= 383  | -      | UBRR= 416 | -      | UBRR= 479  | -      | UBRR= 520 | -      |
| 4800      | UBRR= 191  | 0.0    | UBRR= 207 | 0.2    | UBRR= 239  | 0.0    | UBRR= 259 | -      |
| 9600      | UBRR= 95   | 0.0    | UBRR= 103 | 0.2    | UBRR= 119  | 0.0    | UBRR= 129 | 0.2    |
| 14400     | UBRR= 63   | 0.0    | UBRR= 68  | 0.6    | UBRR= 79   | 0.0    | UBRR= 86  | 0.2    |
| 19200     | UBRR= 47   | 0.0    | UBRR= 51  | 0.2    | UBRR= 59   | 0.0    | UBRR= 64  | 0.2    |
| 28800     | UBRR= 31   | 0.0    | UBRR= 34  | 0.8    | UBRR= 39   | 0.0    | UBRR= 42  | 0.9    |
| 57600     | UBRR= 15   | 0.0    | UBRR= 16  | 2.1    | UBRR= 19   | 0.0    | UBRR= 21  | 1.4    |

**THE 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 which specifies the UART Baud Rate according to the equation on the previous page.

## The Analog Comparator

The analog comparator compares the input values on the positive pin PB2 (AIN0) and negative pin PB3 (AIN1). When the voltage on the positive pin PB2 (AIN0) is higher than the voltage on the negative pin 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 46.

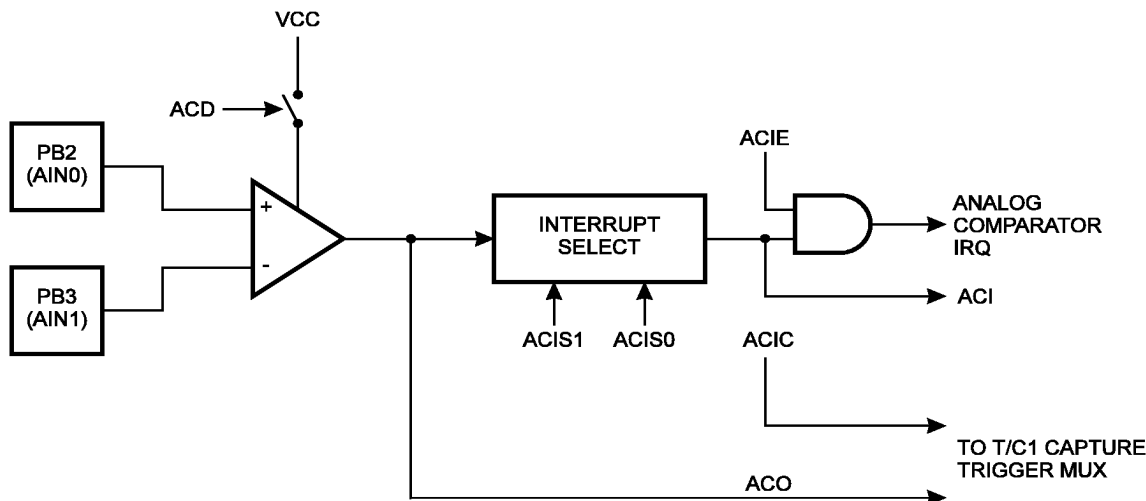


Figure 46. Analog Comparator Block Diagram

### THE 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 | 0   | 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. It is most commonly used if power consumption during Idle Mode is critical, and wake-up from the analog comparator is not required. 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 logic one to the flag.

#### Bit 3 - ACIE : Analog Comparator Interrupt Enable:

When the ACIE bit is set (one) and the I-bit in the Status Register is set (one), the analog comparator interrupt is activated. When cleared (zero), the interrupt is disabled.



#### **Bit 2 - 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 that trigger the Analog Comparator interrupt. The different settings are shown in Table 17.

**Table 17. 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.

## **I/O-Ports**

### **Port A**

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

Three data 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 pullups. The PORT A output buffers can sink 20mA and thus drive LED displays directly. When pins PA0 to PA7 are used as inputs and are externally pulled low, they will source current ( $I_{IL}$ ) if the internal pullups 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 pullups.

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.

## THE 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      |       |

## THE 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    |      |

## THE 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 | Hi-Z  | Hi-Z  | Hi-Z  | Hi-Z  | Hi-Z  | Hi-Z  | Hi-Z  | Hi-Z  |      |

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

## PORTA AS GENERAL DIGITAL I/O

All 8 bits in PORT A are equal 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 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.

**Table 18.** DDAn Effects on PORT A Pins

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

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



## PORT A SCHEMATICS

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

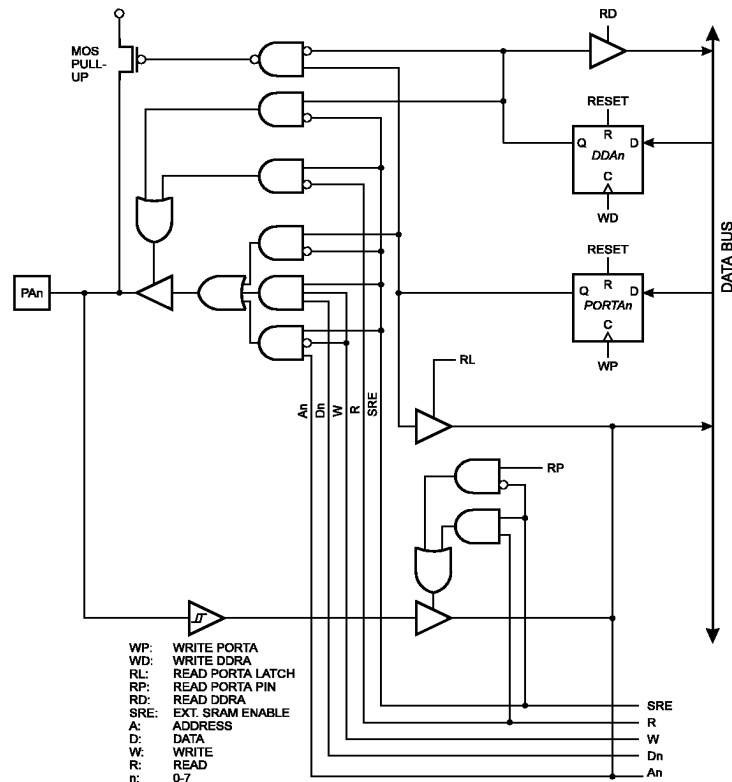


Figure 47. PORTA Schematic Diagrams (Pins PA0 - PA7)

## Port B

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

Three data 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 pullups. The Port B output buffers can sink 20mA and thus drive LED displays directly. When pins PB0 to PB7 are used as inputs and are externally pulled low, they will source current ( $I_{IL}$ ) if the internal pullups are activated.

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

**Table 19.** Port B Pins 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 register has to be set according to the alternate function description.

## THE PORT B DATA REGISTER - PORTB

|               |               |               |               |               |               |               |               |               |              |
|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|--------------|
| Bit           | 7             | 6             | 5             | 4             | 3             | 2             | 1             | 0             |              |
| \$18 (\$38)   | <b>PORTB7</b> | <b>PORTB6</b> | <b>PORTB5</b> | <b>PORTB4</b> | <b>PORTB3</b> | <b>PORTB2</b> | <b>PORTB1</b> | <b>PORTB0</b> | <b>PORTB</b> |
| Read/Write    | R/W           | R/W           | R/W           | R/W           | R/W           | R/W           | R/W           | R/W           |              |
| Initial value | 0             | 0             | 0             | 0             | 0             | 0             | 0             | 0             |              |

## THE PORT B DATA DIRECTION REGISTER - DDRB

|               |             |             |             |             |             |             |             |             |             |
|---------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| Bit           | 7           | 6           | 5           | 4           | 3           | 2           | 1           | 0           |             |
| \$17 (\$37)   | <b>DDB7</b> | <b>DDB6</b> | <b>DDB5</b> | <b>DDB4</b> | <b>DDB3</b> | <b>DDB2</b> | <b>DDB1</b> | <b>DDB0</b> | <b>DDRB</b> |
| Read/Write    | R/W         | R/W         | R/W         | R/W         | R/W         | R/W         | R/W         | R/W         |             |
| Initial value | 0           | 0           | 0           | 0           | 0           | 0           | 0           | 0           |             |

## THE PORT B INPUT PINS ADDRESS - PINB

|               |              |              |              |              |              |              |              |              |             |
|---------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|-------------|
| Bit           | 7            | 6            | 5            | 4            | 3            | 2            | 1            | 0            |             |
| \$16 (\$36)   | <b>PINB7</b> | <b>PINB6</b> | <b>PINB5</b> | <b>PINB4</b> | <b>PINB3</b> | <b>PINB2</b> | <b>PINB1</b> | <b>PINB0</b> | <b>PINB</b> |
| Read/Write    | R            | R            | R            | R            | R            | R            | R            | R            |             |
| Initial value | Hi-Z         | Hi-Z         | Hi-Z         | Hi-Z         | Hi-Z         | Hi-Z         | Hi-Z         | Hi-Z         |             |

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

## PORTB AS GENERAL DIGITAL I/O

All 8 bits in port B are equal 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 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.



**Table 20.** DDBn Effects on Port B Pins

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

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

### ALTERNATE FUNCTIONS OF PORTB

The alternate pin configuration is as follows:

#### **SCK - PORTB, 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 - PORTB, 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 - PORTB, 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 - PORTB, Bit 4:**

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 DDB5. 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 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.

#### **AIN1 - PORTB, 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 - PORTB, 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 - PORTB, Bit 1:**

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

#### **T0 - PORTB, 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.

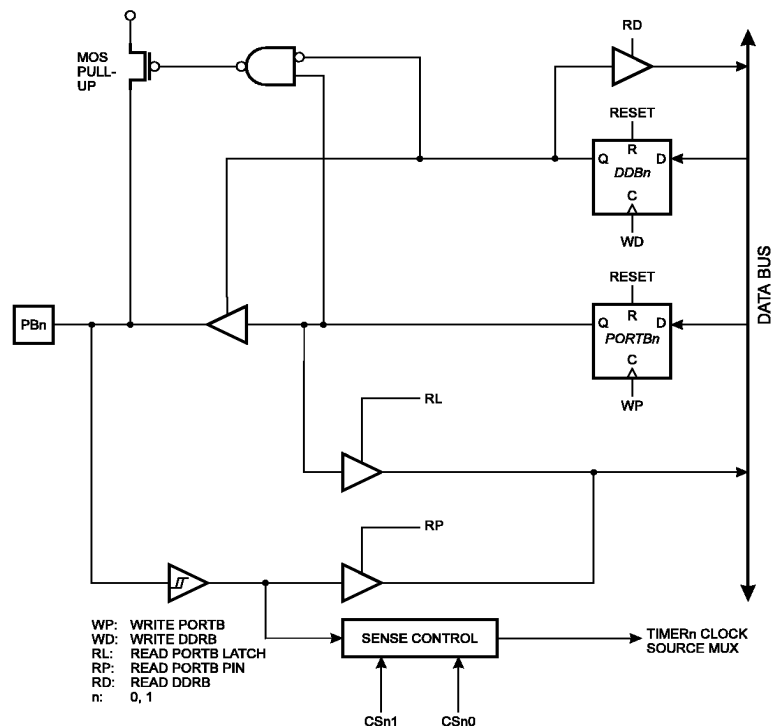


Figure 48. PORTB Schematic Diagram (Pins PB0 and PB1)

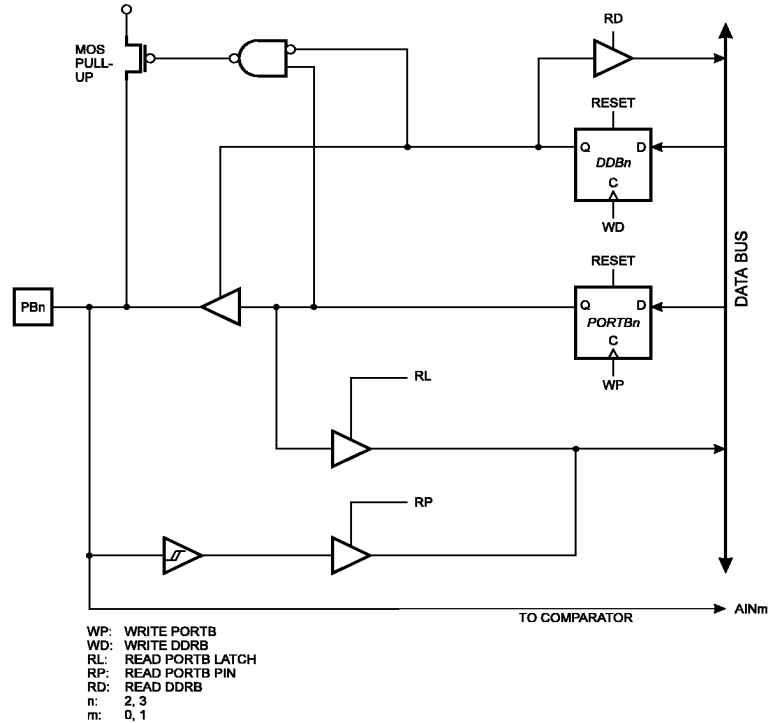
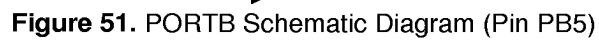
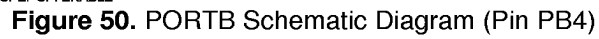
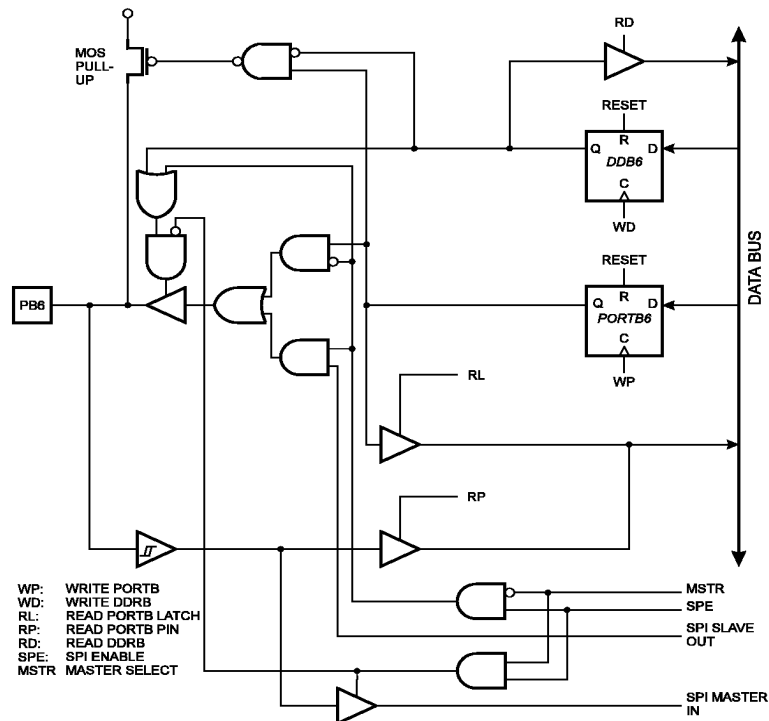
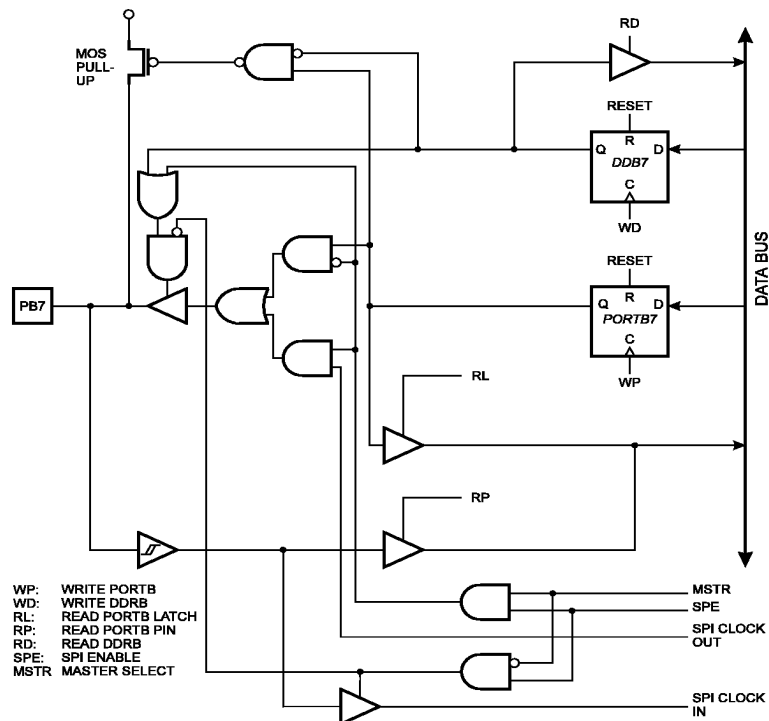


Figure 49. PORTB Schematic Diagram (Pins PB2 and PB3)





**Figure 52. PORTB Schematic Diagram (Pin PB6)**



**Figure 53. PORTB Schematic Diagram (Pin PB7)**

## Port C

PORT C is an 8-bit bi-directional I/O port.

Three data 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 pullups. The PORT C output buffers can sink 20mA and thus drive LED displays directly. When pins PC0 to PC7 are used as inputs and are externally pulled low, they will source current ( $I_{IL}$ ) if the internal pullups 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. In this mode, PORT C uses internal pullups when emitting 1's.

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.

### THE PORT C DATA REGISTER - PORTC

|               |                                                                                                                                                  |        |        |        |        |        |        |     |        |        |        |        |        |        |        |        |       |
|---------------|--------------------------------------------------------------------------------------------------------------------------------------------------|--------|--------|--------|--------|--------|--------|-----|--------|--------|--------|--------|--------|--------|--------|--------|-------|
| Bit           | 7                                                                                                                                                | 6      | 5      | 4      | 3      | 2      | 1      | 0   |        |        |        |        |        |        |        |        |       |
| \$15 (\$35)   | <table><tr><td>PORTC7</td><td>PORTC6</td><td>PORTC5</td><td>PORTC4</td><td>PORTC3</td><td>PORTC2</td><td>PORTC1</td><td>PORTC0</td></tr></table> |        |        |        |        |        |        |     | PORTC7 | PORTC6 | PORTC5 | PORTC4 | PORTC3 | PORTC2 | PORTC1 | PORTC0 | PORTC |
| PORTC7        | PORTC6                                                                                                                                           | PORTC5 | PORTC4 | PORTC3 | PORTC2 | PORTC1 | PORTC0 |     |        |        |        |        |        |        |        |        |       |
| 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 PORT C DATA DIRECTION REGISTER - DDRC

|               |                                                                                                                                  |      |      |      |      |      |      |     |      |      |      |      |      |      |      |      |      |
|---------------|----------------------------------------------------------------------------------------------------------------------------------|------|------|------|------|------|------|-----|------|------|------|------|------|------|------|------|------|
| Bit           | 7                                                                                                                                | 6    | 5    | 4    | 3    | 2    | 1    | 0   |      |      |      |      |      |      |      |      |      |
| \$14 (\$34)   | <table><tr><td>DDC7</td><td>DDC6</td><td>DDC5</td><td>DDC4</td><td>DDC3</td><td>DDC2</td><td>DDC1</td><td>DDC0</td></tr></table> |      |      |      |      |      |      |     | DDC7 | DDC6 | DDC5 | DDC4 | DDC3 | DDC2 | DDC1 | DDC0 | DDRC |
| DDC7          | DDC6                                                                                                                             | DDC5 | DDC4 | DDC3 | DDC2 | DDC1 | DDC0 |     |      |      |      |      |      |      |      |      |      |
| 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 PORT C INPUT PINS ADDRESS - PINC

|               |                                                                                                                                          |       |       |       |       |       |       |      |       |       |       |       |       |       |       |       |      |
|---------------|------------------------------------------------------------------------------------------------------------------------------------------|-------|-------|-------|-------|-------|-------|------|-------|-------|-------|-------|-------|-------|-------|-------|------|
| Bit           | 7                                                                                                                                        | 6     | 5     | 4     | 3     | 2     | 1     | 0    |       |       |       |       |       |       |       |       |      |
| \$13 (\$33)   | <table><tr><td>PINC7</td><td>PINC6</td><td>PINC5</td><td>PINC4</td><td>PINC3</td><td>PINC2</td><td>PINC1</td><td>PINC0</td></tr></table> |       |       |       |       |       |       |      | PINC7 | PINC6 | PINC5 | PINC4 | PINC3 | PINC2 | PINC1 | PINC0 | PINC |
| PINC7         | PINC6                                                                                                                                    | PINC5 | PINC4 | PINC3 | PINC2 | PINC1 | PINC0 |      |       |       |       |       |       |       |       |       |      |
| Read/Write    | R                                                                                                                                        | R     | R     | R     | R     | R     | R     | R    |       |       |       |       |       |       |       |       |      |
| Initial value | Hi-Z                                                                                                                                     | Hi-Z  | Hi-Z  | Hi-Z  | Hi-Z  | Hi-Z  | Hi-Z  | Hi-Z |       |       |       |       |       |       |       |       |      |

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

### PORTC AS GENERAL DIGITAL I/O

All 8 bits in PORT C are equal 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 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.

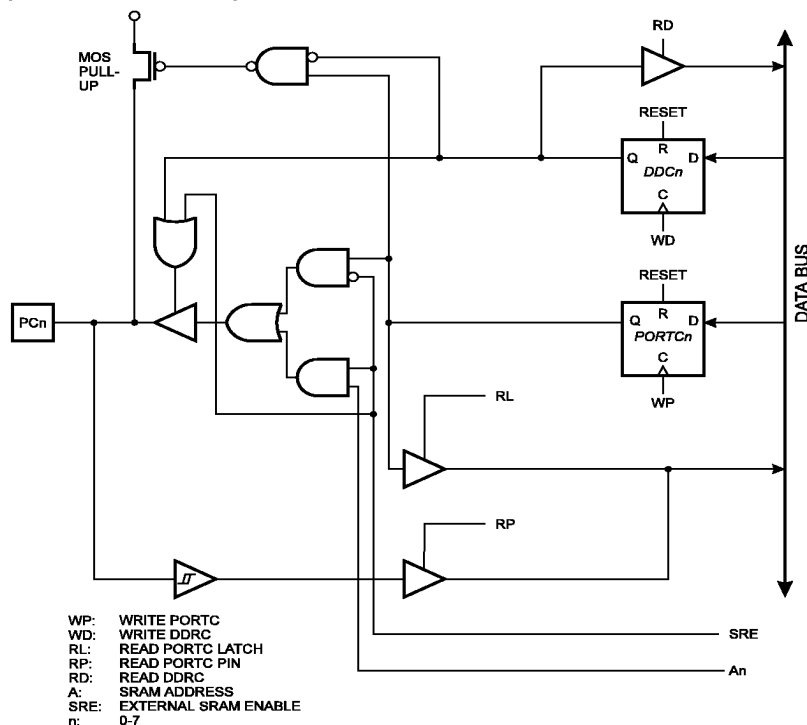
**Table 21.** DDcn Effects on PORT C Pins

| DDcn | PORTCn | I/O    | Pull up | Comment                                                  |
|------|--------|--------|---------|----------------------------------------------------------|
| 0    | 0      | Input  | No      | Tri-state (Hi-Z)                                         |
| 0    | 1      | Input  | Yes     | PCn will source current ( $I_{IL}$ ) if ext. pulled low. |
| 1    | 0      | Output | No      | Push-Pull Zero Output                                    |
| 1    | 1      | Output | No      | Push-Pull One Output                                     |

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 54.** PORTC Schematic Diagram (Pins PC0 - PC7)

## Port D

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

Three data 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 ( $I_{IL}$ ) if the pullups are activated.

Some Port D pins have alternate functions as shown in the following table:

**Table 22.** Port D Pins Alternate Functions

| Port Pin | Alternate Function                                 |
|----------|----------------------------------------------------|
| PD0      | RDX (UART Input line )                             |
| PD1      | TDX (UART Output line)                             |
| PD2      | INT0 (External interrupt 0 input)                  |
| PD3      | INT1 (External interrupt 1 input)                  |
| PD5      | OC1A (Timer/Counter1 Output compareA match output) |
| PD6      | WR (Write strobe to external memory)               |
| PD7      | RD (Read strobe to external memory)                |

When the pins are used for the alternate function the DDRD and PORTD register has to be set according to the alternate function description.



## THE 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      |       |

## THE 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    |      |

## THE 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 | Hi-Z  | Hi-Z  | Hi-Z  | Hi-Z  | Hi-Z  | Hi-Z  | Hi-Z  | Hi-Z  |      |

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

## PORTD 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.

**Table 23.** DDDn Bits on Port D Pins

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

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

## ALTERNATE FUNCTIONS OF PORTD

### RD - PORTD, Bit 7:

RD is the external data memory read control strobe.

### WR - PORTD, Bit 6:

WR is the external data memory write control strobe.

### OC1 - PORTD, Bit 5:

OC1, 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 OC1 pin is also the output pin for the PWM mode timer function.

**INT1 - PORTD, 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 - PORTD, 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 - PORTD, Bit 1:**

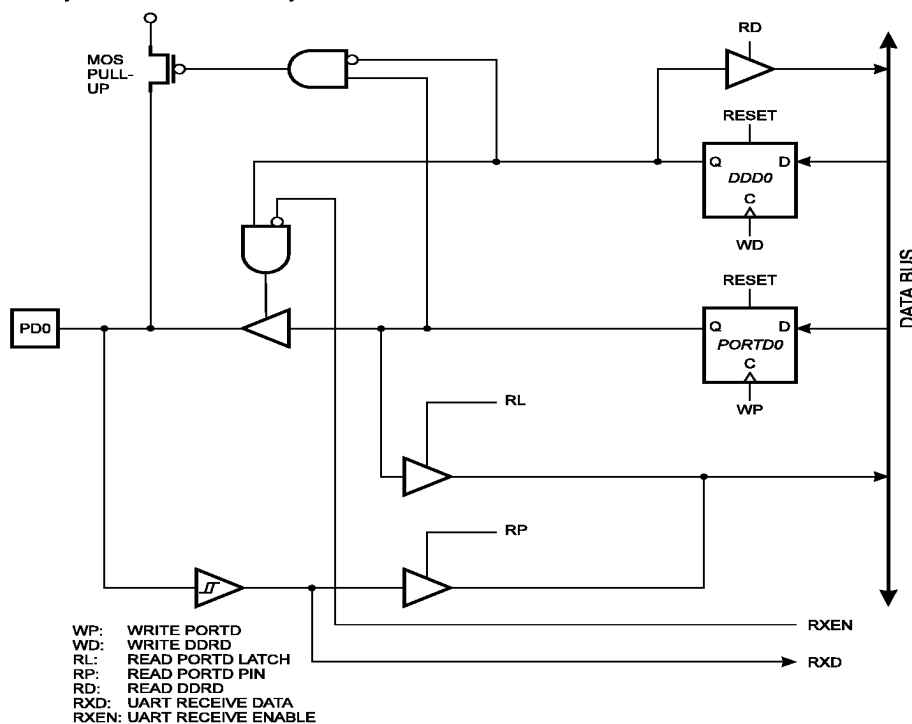
Transmit Data (Data output pin for the UART). When the UART transmitter is enabled, this pin is configured as an output regardless of the value of DDRD1.

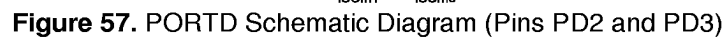
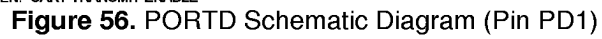
**RXD - PORTD, Bit 0:**

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

## PORTD SCHEMATICS

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





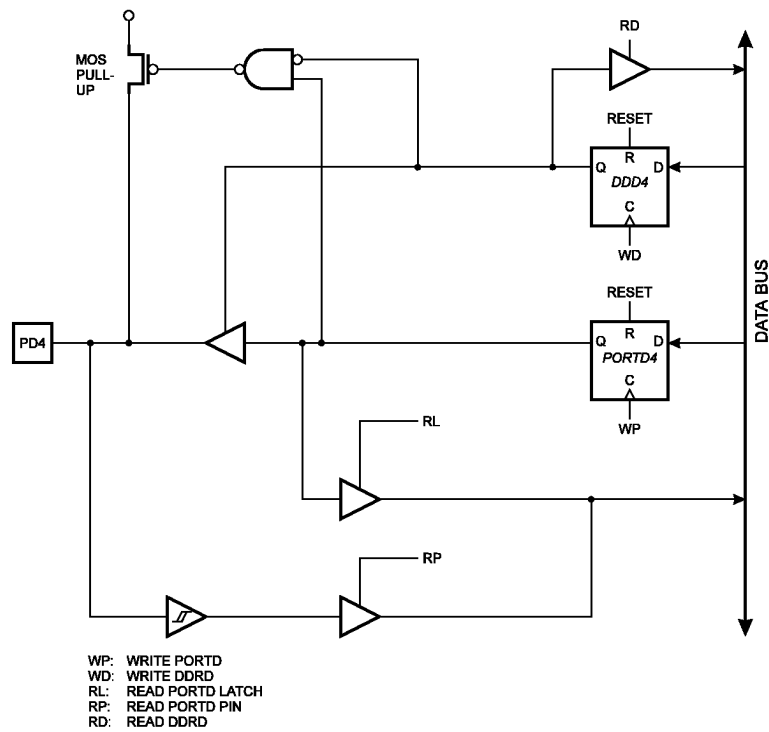


Figure 58. PORTD Schematic Diagram (Pin PD4)

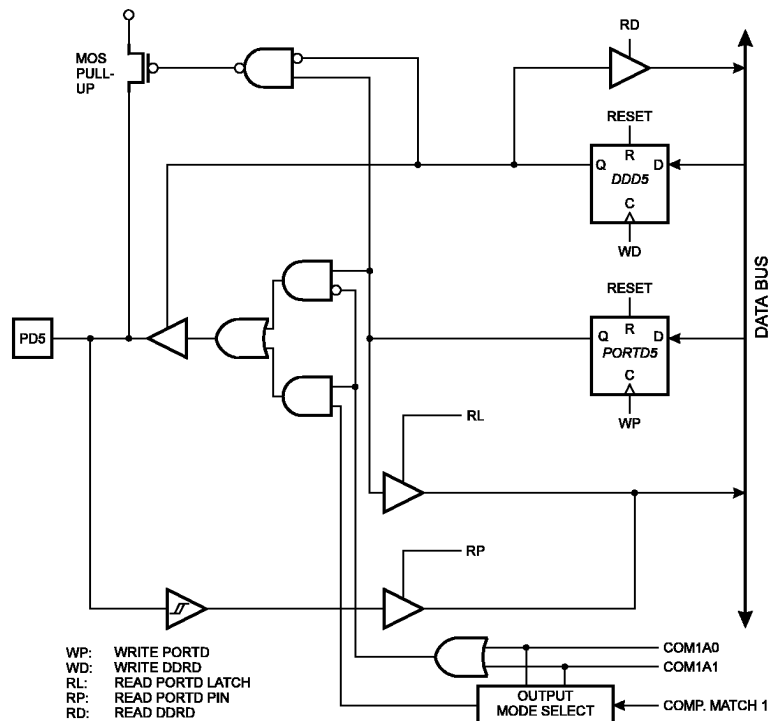
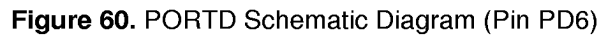


Figure 59. PORTD Schematic Diagram (Pin PD5)



## Memory Programming

### Program Memory Lock Bits

The AT90S8515 MCU provides two lock bits which can be left unprogrammed ('1') or can be programmed ('0') to obtain the additional features listed in Table 24.

**Table 24.** Lock Bit Protection Modes

| Program Lock Bits |     |     | Protection Type                              |
|-------------------|-----|-----|----------------------------------------------|
| Mode              | LB1 | LB2 |                                              |
| 1                 | 1   | 1   | No program lock features                     |
| 2                 | 0   | 1   | Further programming of the Flash is disabled |
| 3                 | 0   | 0   | Same as mode 2, but verify is also disabled. |

Note: The Lock Bits can only be erased with the Chip Erase operation.

### Fuse Bits

The AT90S8515 has two fuse bits, SPIEN and FSTRT.

- When SPIEN is programmed ('0'), Serial Program Downloading is enabled. Default value is programmed ('0').
- When FSTRT 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.

These bits are not accessible in Serial Programming Mode and are not affected by a chip erase.

### Signature Bytes

All Atmel microcontrollers have a three-byte signature code which identifies the device. This code can be read in both serial and parallel mode. The three bytes reside in a separate address space, and for the AT90S8515 they are:

1. \$000: \$1E (indicates manufactured by Atmel)
2. \$001: \$93 (indicates 8kB Flash memory)
3. \$002: \$01 (indicates 90S8515 device when \$001 is \$93)

### 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.

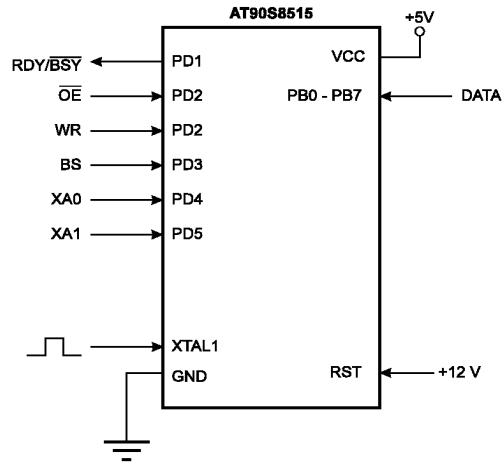
The AT90S8515 is normally shipped with the on-chip Flash Program and EEPROM Data memory arrays in the erased state (i.e. contents = \$FF) and ready to be programmed. This device supports a High-Voltage (12V) Parallel programming mode and a Low-Voltage Serial programming mode. The +12V is used for programming enable only, and no current of significance is drawn by this pin. The serial programming mode provides a convenient way to download the 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 modes. For the EEPROM, an auto-erase cycle is provided with the self-timed programming operation in the serial programming mode.



## Parallel Programming

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



**Figure 62.** Parallel Programming

### SIGNAL NAMES

In this section, some pins of the AT908515 are referenced by signal names describing their functionality during parallel programming rather than their pin names. Pins not described in the following table are referenced by pin names.

**Table 25.** Pin Name Mapping

| Signal Name in Programming Mode | Pin Name | I/O | Function                                                          |
|---------------------------------|----------|-----|-------------------------------------------------------------------|
| RDY / BSY                       | PD1      | O   | 0: Device is busy programming, 1: Device is ready for new command |
| OE                              | PD2      | I   | Output Enable (Active Low)                                        |
| WR                              | PD3      | I   | Write Pulse (Active Low)                                          |
| BS                              | PD4      | I   | Byte Select                                                       |
| XA0                             | PD5      | I   | XTAL Action Bit 0                                                 |
| XA1                             | PD6      | I   | XTAL Action Bit 1                                                 |

The XA1/XA0 bits determine the action taken when the XTAL1 pin is given a positive pulse. The bit settings are shown in the following table:

**Table 26.** XA1 and XA0 Coding

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

When pulsing  $\overline{WR}$  or  $\overline{OE}$ , the command loaded determines the action on input or output. The command is a byte where the different bits are assigned functions as shown in the following table:

**Table 27.** Command Byte Bit Coding

| Bit# | Meaning when Set                                                                                                                                                    |
|------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 7    | Chip Erase                                                                                                                                                          |
| 6    | Write Fuse Bits. Located in the data byte at the following bit positions:<br>D5: SPIEN Fuse, D0: FSTRT Fuse (Note: Write '0' to program, '1' to erase)              |
| 5    | Write Lock Bits. Located in the data byte at the following bit positions:<br>D1: LB1, D0: LB2 (Note: write '0' to program)                                          |
| 4    | Write Flash or EEPROM (determined by bit 0)                                                                                                                         |
| 3    | Read signature row                                                                                                                                                  |
| 2    | Read Lock and Fuse Bits. Located in the data byte at the following bits positions:<br>D7: LB1, D6: LB2, D5: SPIEN Fuse, D0: FSTRT Fuse (Note: '0' means programmed) |
| 1    | Read from Flash or EEPROM (determined by bit 0)                                                                                                                     |
| 0    | 0 : Flash Access, 1 : EEPROM Access                                                                                                                                 |

## ENTER PROGRAMMING MODE

The following algorithm puts the device in parallel programming mode:

1. Apply 4.5 - 5.5 V between VCC and GND.
2. Set RESET and BS pins to '0' and wait at least 100 ns.
3. Apply 12V to and wait at least 100 ns before changing BS.

## CHIP ERASE

The chip erase will erase the Flash and EEPROM memories plus Lock bits. The lock bits are not reset until the program memory has been completely erased. The Fuse bits are not changed. A chip erase must be performed before the chip is programmed.

*Load Command "Chip Erase"*

1. Set XA1, XA0 to '10'. This enables command loading.
2. Set BS to '0'.
3. Set PB(7:0) to '1000 0000'. This is the command for Chip erase.
4. Give XTAL1 a positive pulse. This loads the command, and starts the erase of the Flash and EEPROM arrays. After pulsing XTAL1, give  $\overline{WR}$  a negative pulse to enable lock bit erase at the end of the erase cycle, then wait for at least 10 ms. Chip erase does not generate any activity on the RDY/BSY pin.

## PROGRAMMING THE FLASH

*Load Command "Program Flash"*

1. Set XA1, XA0 to '10'. This enables command loading.
2. Set BS to '0'.
3. Set PB(7:0) to '0001 0000'. This is the command for Flash programming.
4. Give XTAL1 a positive pulse. This loads the command.

*Load Address Low byte*

1. Set XA1, XA0 to '00'. This enables address loading.
2. Set BS to '0'. This selects Low address.
3. Set PB(7:0) = Address Low byte (\$00 - \$FF)
4. Give XTAL1 a positive pulse. This loads the Address Low byte.



#### Load Address High byte

1. Set XA1, XA0 to '00'. This enables address loading.
2. Set BS to '1'. This selects High address.
3. Set PB(7:0) = Address High byte (\$00 - \$0F)
4. Give XTAL1 a positive pulse. This loads the Address High byte.

#### Load Data byte

1. Set XA1, XA0 to '01'. This enables data loading.
2. Set PB(7:0) = Data Low byte (\$00 - \$FF)
3. Give XTAL1 a positive pulse. This loads the Data byte.

#### Write Data Low byte

1. Set BS to '0'.
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.

#### Load Data byte

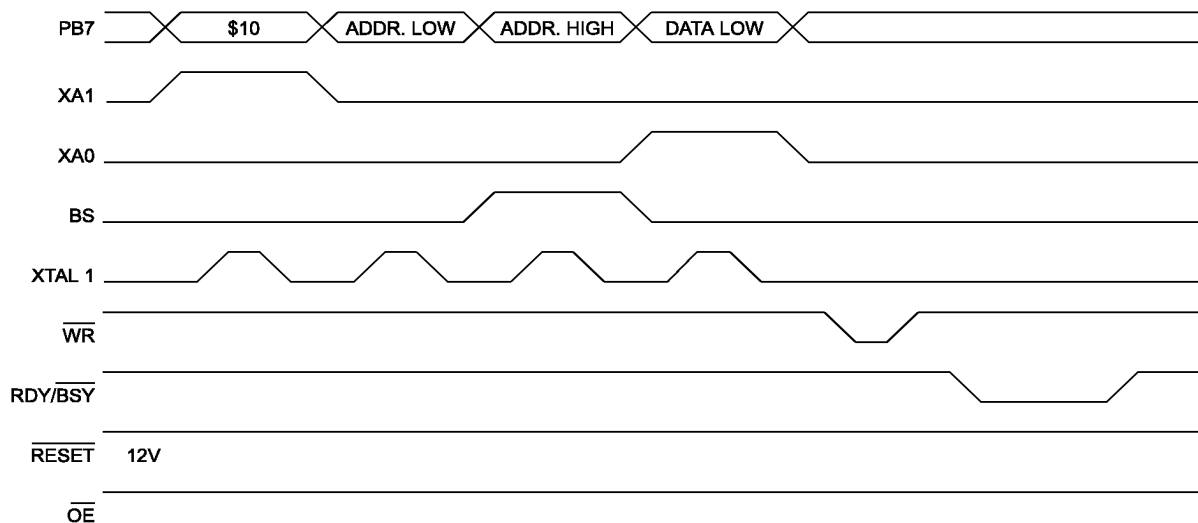
1. Set XA1, XA0 to '01'. This enables data loading.
2. Set PB(7:0) = Data High byte (\$00 - \$FF)
3. Give XTAL1 a positive pulse. This loads the Data byte.

#### Write Data High byte

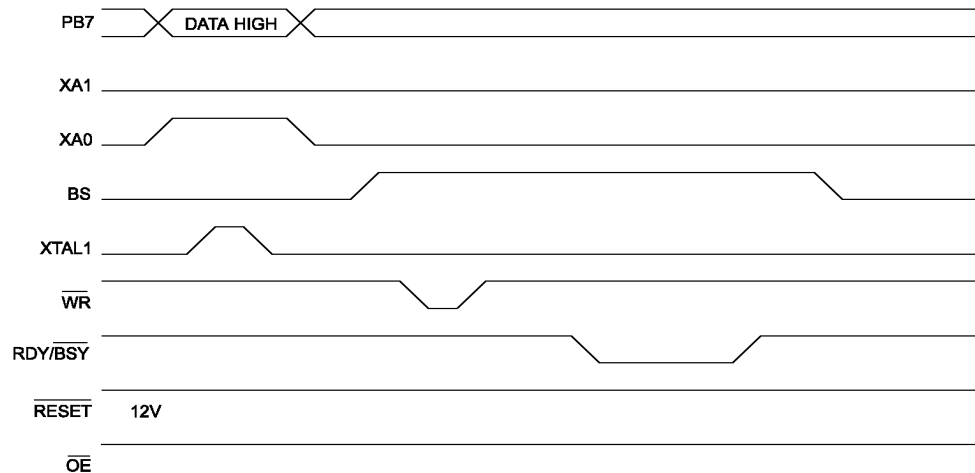
1. Set BS to '1'.
2. Give  $\overline{WR}$  a negative pulse. This starts programming of the data byte. RDY /  $\overline{BSY}$  goes low.
3. Wait until RDY /  $\overline{BSY}$  goes high to program the next byte.

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

- The command for Flash programming needs only be loaded before programming of the first byte.
- Address High byte needs only be loaded before programming a new 256 word page in the Flash.



**Figure 63.** Programming Flash Low Byte



**Figure 64.** Programming Flash High Byte

### PROGRAMMING THE EEPROM

The programming algorithm for the EEPROM data memory is as follows (refer to Flash Programming for details on Command, Address and Data loading):

1. Load Command '0001 0001'.
2. Load Low EEPROM Address (\$00 - \$FF)
3. Load High EEPROM Address (\$00 - \$01)
4. Load Low EEPROM Data (\$00 - \$FF)
5. Give  $\overline{WR}$  a negative pulse and wait for RDY/BSY to go high.

The Command needs only be loaded before programming the first byte.

### READING THE FLASH

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

1. Load Command '0000 0010'.
2. Load Low Address (\$00 - \$FF)
3. Load High Address (\$00 - \$0F)
4. Set  $\overline{OE}$  to '0', and BS to '0'. The Low Data byte can now be read at PB(7:0)
5. Set BS to '1'. The High Data byte can now be read from PB(7:0)
6. Set  $\overline{OE}$  to '1'.

The Command needs only be loaded before reading the first byte.

### READING THE EEPROM

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

1. Load Command '0000 0011'.
2. Load Low EEPROM Address (\$00 - \$FF)
3. Load High EEPROM Address (\$00 - \$FF)
4. Set  $\overline{OE}$  to '0', and BS to '0'. The EEPROM Data byte can now be read at PB(7:0)
5. Set  $\overline{OE}$  to '1'.

The Command needs only be loaded before reading the first byte.



### PROGRAMMING THE FUSE BITS

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

1. Load Command '0100 0000'.
2. Load Data.
  - Bit 5 = '0' programs the SPIEN Fuse bit. Bit 5 = '1' erases the SPIEN Fuse bit.
  - Bit 0 = '0' programs the FSTRT fuse bit. Bit 5 = '1' erases the FSTRT fuse bit.
3. Give  $\overline{WR}$  a negative pulse and wait for RDY/BSY to go high.

### PROGRAMMING THE LOCK BITS

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

1. Load Command '0010 0000'.
2. Load Data.
  - Bit 2 = '0' programs Lock Bit2
  - Bit 1 = '0' programs Lock Bit1
3. Give  $\overline{WR}$  a negative pulse and wait for RDY/BSY to go high.

The lock bits can only be cleared by executing a chip erase.

### READING THE FUSE AND LOCK BITS

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

1. Load Command '0000 0100'.
2. Set  $\overline{OE}$  to '0', and BS to '1'. The Status of Fuse and Lock bits can now be read at PB(7:0)
  - Bit 7: Lock Bit1 ('0' means programmed)
  - Bit 6: Lock Bit2 ('0' means programmed)
  - Bit 5: SPIEN Fuse ('0' means programmed, '1' means erased)
  - Bit 0: FSTRT Fuse ('0' means programmed, '1' means erased)
3. Set  $\overline{OE}$  to '1'.

Observe especially that BS needs to be set to '1'.

### READING THE SIGNATURE BYTES

The algorithm for reading the Signature Bytes bits is as follows (refer to Flash Programming for details on Command, Address and Data loading):

1. Load Command '0000 1000'.
2. Load Low address (\$00 - \$02)
3. Set  $\overline{OE}$  to '0', and BS to '0'. The Selected Signature byte can now be read at PB(7:0)
4. Set  $\overline{OE}$  to '1'.

The command needs only be programmed before reading the first byte.

### Serial Downloading

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

When programming the EEPROM, an auto-erase cycle is built into the self-timed programming operation (in the serial mode ONLY) and there is no need to first execute the Chip Erase instruction. The Chip Erase operation 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 for Program memory and \$0000 to \$001FF for EEPROM memory.

Either an external system 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: > 1 XTAL1 clock cycle

High: > 4 XTAL1 clock cycles

### SERIAL PROGRAMMING ALGORITHM

To program and verify the AT90S8515 in the serial programming mode, the following sequence is recommended (See four byte instruction formats in Table 28):

1. *Power-up sequence:*

Apply power between VCC and GND while **RESET** and SCK are set to '0'. (If the programmer can not guarantee that SCK is held low during power-up, **RESET** must be given a positive pulse after SCK has been set to '0'.) If a crystal is not connected across pins XTAL1 and XTAL2, apply a 0 to 20 MHz clock to the XTAL1 pin.

2. Wait for at least 20 ms and enable serial programming by sending the Programming Enable serial instruction to pin MOSI/PB5. Refer to the above section for minimum high and low periods for the serial clock input (SCK).

3. If a chip erase is performed (must be done to erase the Flash), wait 10ms, give **RESET** a positive pulse and start over again from Step 2.

4. 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. The next byte can be written after 4 ms.

5. Any memory location can be verified by using the Read instruction which returns the content at the selected address at serial output MISO/PB6.

6. At the end of the programming session, **RESET** can be set high to commence normal operation.

7. *Power-off sequence (if needed):*

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

Set **RESET** to '1'.

Turn V<sub>CC</sub> power off

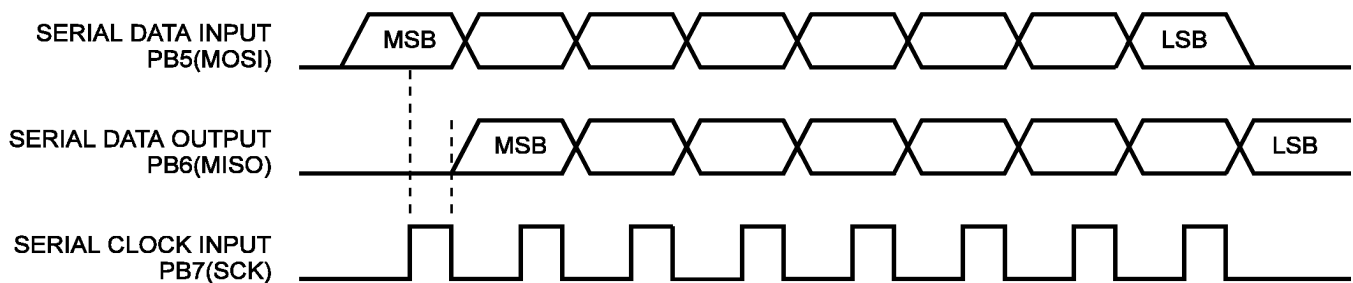


**Table 28. Serial Programming Instruction Set**

| Instruction          | Instruction Format |           |           |           | Operation                                                          |
|----------------------|--------------------|-----------|-----------|-----------|--------------------------------------------------------------------|
|                      | Byte 1             | Byte 2    | Byte 3    | Byte4     |                                                                    |
| Programming Enable   | 1010 1100          | 0101 0011 | xxxx xxxx | xxxx xxxx | Enable Serial Programming after RESET goes low.                    |
| Chip Erase           | 1010 1100          | 100x xxxx | xxxx xxxx | xxxx xxxx | Chip erase both 8K & 512byte memory arrays                         |
| Read Program Memory  | 0010 H000          | 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 H000          | 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 xxx0 | bbbb bbbb | oooo oooo | Read data o from EEPROM memory at address a:b                      |
| Write EEPROM Memory  | 1100 0000          | xxxx xxx0 | bbbb bbbb | iiii iiii | Write data i to EEPROM memory at address a:b                       |
| Write Lock Bits      | 1010 1100          | 111x x21x | xxxx xxxx | xxxx xxxx | Write lock bits. Set bits 1,2='0' to program lock bits.            |
| Read Device Code     | 0011 0000          | xxxx xxxx | xxxx xxbb | oooo oooo | Read Device Code o at address b                                    |

Notes: **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

## Programming Characteristics



**Figure 65. Serial Downloading Waveforms**

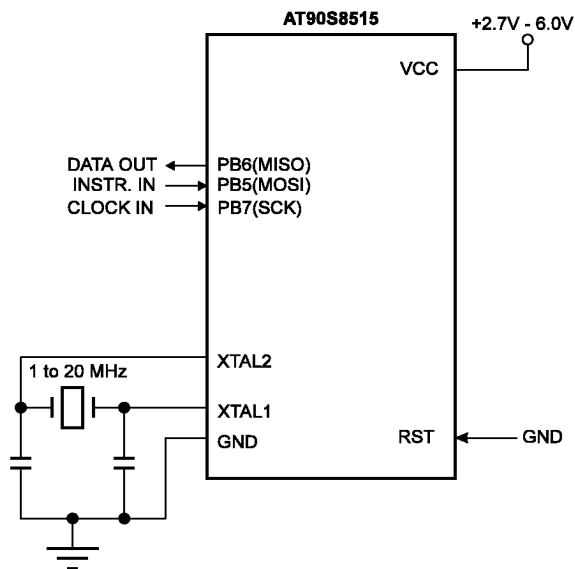


Figure 66. Serial Programming and Verify

When *writing* serial data to the AT90S8515, data is clocked on the *rising* edge of CLK.  
When *reading* data from the AT90S8515, data is clocked on the *falling* edge of CLK. See Figure 65 for an explanation.

Absolute Maximum Ratings\*

|                                                                |                 |
|----------------------------------------------------------------|-----------------|
| Operating Temperature .....                                    | -40°C to +105°C |
| Storage Temperature.....                                       | -65°C to +150°C |
| Voltage on any Pin except RESET<br>with respect to Ground..... | -1.0V to +7.0V  |
| Maximum Operating Voltage .....                                | 6.6V            |
| I/O Pin Maximum Current .....                                  | 40.0 mA         |
| Maximum Current VCC and GND.....                               | 140.0 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

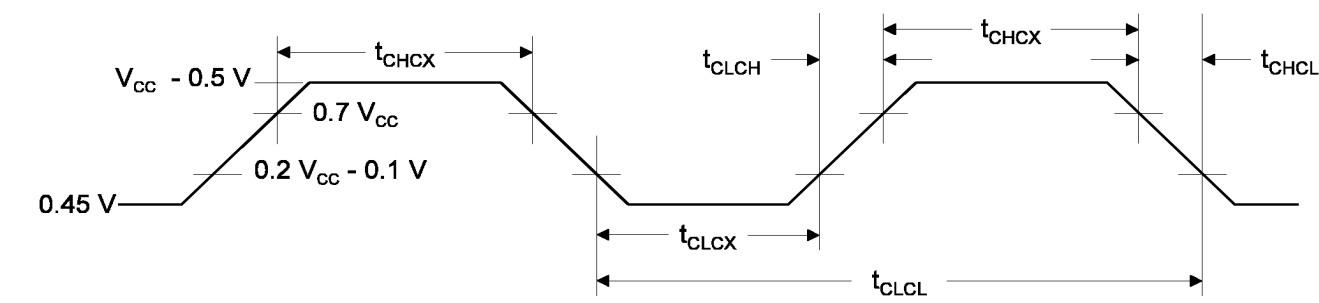
$T_A = -40^{\circ}\text{C}$  to  $85^{\circ}\text{C}$ ,  $V_{CC} = 2.7\text{V}$  to  $6.0\text{V}$  (unless otherwise noted)

| Symbol     | Parameter                                          | Condition                                                                                          | Min                | Typ  | Max                | Units         |
|------------|----------------------------------------------------|----------------------------------------------------------------------------------------------------|--------------------|------|--------------------|---------------|
| $V_{IL}$   | Input Low Voltage                                  |                                                                                                    | -0.5               |      | $0.2 V_{CC} - 0.1$ | V             |
| $V_{IH}$   | Input High Voltage                                 | (Except XTAL1, RESET)                                                                              | $0.2 V_{CC} + 0.9$ |      | $V_{CC} + 0.5$     | V             |
| $V_{IH1}$  | Input High Voltage                                 | (XTAL1, RESET)                                                                                     | $0.7 V_{CC}$       |      | $V_{CC} + 0.5$     | V             |
| $V_{OL}$   | Output Low Voltage <sup>(1)</sup><br>(Ports B,C,D) | $I_{OL} = 20\text{ mA}$ , $V_{CC} = 5\text{V}$<br>$I_{OL} = 10\text{ mA}$ , $V_{CC} = 2.7\text{V}$ |                    |      | 0.5                | V             |
| $V_{OH}$   | Output High Voltage<br>(Ports B,C,D)               | $I_{HI} = 10\text{ mA}$ , $V_{CC} = 5\text{V}$<br>$I_{HI} = 5\text{ mA}$ , $V_{CC} = 2.7\text{V}$  | 4.5                |      |                    | V             |
| $I_{OH}$   | Output Source Current<br>(Ports B,C,D)             | $V_{CC} = 5\text{V}$<br>$V_{CC} = 2.7\text{V}$                                                     |                    |      | 10<br>5            | mA            |
| $I_{OL}$   | Output Sink Current<br>(Port B,C,D)                | $V_{CC} = 5\text{V}$<br>$V_{CC} = 2.7\text{V}$                                                     |                    |      | 20<br>10           | mA            |
| RRST       | Reset Pulldown Resistor                            |                                                                                                    | 10                 |      | 50                 | k $\Omega$    |
| $R_{I/O}$  | I/O Pin Pull-Up Resistor                           |                                                                                                    | 35                 |      | 120                | k $\Omega$    |
| $I_{CC}$   | Power Supply Current                               | Active Mode, 3V, 4MHz                                                                              |                    | 3.5  |                    | mA            |
|            |                                                    | Idle Mode 3V, 4MHz                                                                                 |                    | 1000 |                    | $\mu\text{A}$ |
| $I_{CC}$   | Power Down Mode(2)                                 | WDT enabled, 3V                                                                                    |                    | 50   |                    | $\mu\text{A}$ |
|            |                                                    | WDT disabled, 3V                                                                                   |                    | <1   |                    | $\mu\text{A}$ |
| $V_{ACIO}$ | Analog Comparator Input Offset Voltage             | $V_{CC} = 5\text{V}$                                                                               |                    |      | 20                 | mV            |
| $I_{ACLK}$ | Analog Comparator Input Leakage Current            |                                                                                                    | 1                  | 5    | 10                 | nA            |
| $t_{ACPD}$ | Analog Comparator Propagation Delay                | $V_{CC} = 2.7\text{V}$                                                                             |                    | 750  |                    | ns            |
|            |                                                    | $V_{CC} = 4.0\text{V}$                                                                             |                    | 500  |                    |               |

### Notes:

- Under steady state (non-transient) conditions,  $I_{OL}$  must be externally limited as follows:  
Maximum  $I_{OL}$  per port pin: 10 mA  
Maximum total  $I_{OL}$  for all output pins: 80 mA  
Port A: 26 mA  
Ports A, B, D: 15 mA  
Maximum total  $I_{OL}$  for all output pins: 70 mA  
If  $I_{OL}$  exceeds the test condition,  $V_{OL}$  may exceed the related specification.  
Pins are not guaranteed to sink current greater than the listed test conditions.
- Minimum  $V_{CC}$  for Power Down is 2V.

External Clock Drive Waveforms



External Clock Drive

| Symbol                      | Parameter            | V <sub>CC</sub> = 2.7V to 6.0V |     | V <sub>CC</sub> = 4.0V to 6.0V |      | Units |
|-----------------------------|----------------------|--------------------------------|-----|--------------------------------|------|-------|
|                             |                      | Min                            | Max | Min                            | Max  |       |
| 1/ <i>t</i> <sub>CLCL</sub> | Oscillator Frequency | 0                              | 10  | 0                              | 20   | MHz   |
| <i>t</i> <sub>CLCL</sub>    | Clock Period         | 100                            |     | 41.7                           |      | ns    |
| <i>t</i> <sub>CHCX</sub>    | High Time            | 40                             |     | 16.7                           |      | ns    |
| <i>t</i> <sub>CLCX</sub>    | Low Time             | 40                             |     | 16.7                           |      | ns    |
| <i>t</i> <sub>CLCH</sub>    | Rise Time            |                                | 10  |                                | 4.15 | ns    |
| <i>t</i> <sub>CHCL</sub>    | Fall Time            |                                | 10  |                                | 4.15 | ns    |





## Ordering Information

| Ordering Code                | Package     | Operation Range               |
|------------------------------|-------------|-------------------------------|
| AT90S8515-JC<br>AT90S8515-PC | 44J<br>40P6 | Commercial<br>(0°C to 70°C)   |
| AT90S8515-JI<br>AT90S8515-PI | 44J<br>40P6 | Industrial<br>(-40°C to 85°C) |

| Package Type |                                                           |
|--------------|-----------------------------------------------------------|
| 44J          | 44 Lead, Plastic J-Leaded Chip Carrier (PLCC)             |
| 40P6         | 40 Lead, 0.600" Wide, Plastic Dual in Line Package (PDIP) |



## AT90S8515 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      | 5-23 |
| \$3E (\$5E) | SPH      | SP15                                                 | SP14   | SP13   | SP12   | SP11   | SP10   | SP9    | SP8    | 5-24 |
| \$3D (\$5D) | SPL      | SP7                                                  | SP6    | SP5    | SP4    | SP3    | SP2    | SP1    | SP0    | 5-24 |
| \$3C (\$5C) | Reserved |                                                      |        |        |        |        |        |        |        |      |
| \$3B (\$5B) | GIMSK    | INT1                                                 | INT0   | -      | -      | -      | -      | -      | -      | 5-29 |
| \$3A (\$5A) | GIFR     | INTF1                                                | INTF0  |        |        |        |        |        |        | 5-29 |
| \$39 (\$59) | TIMSK    | TOIE1                                                | OCIE1A | OCIE1B | -      | TICIE1 | -      | TOIE0  | -      | 5-29 |
| \$38 (\$58) | TIFR     | TOV1                                                 | OCF1A  | OCF1B  | -      | ICF1   | -      | TOV0   | -      | 5-30 |
| \$37 (\$57) | Reserved |                                                      |        |        |        |        |        |        |        |      |
| \$36 (\$56) | Reserved |                                                      |        |        |        |        |        |        |        |      |
| \$35 (\$55) | MCUCR    | SRE                                                  | SRW    | SE     | SM     | ISC11  | ISC10  | ISC01  | ISC00  | 5-31 |
| \$34 (\$54) | Reserved |                                                      |        |        |        |        |        |        |        |      |
| \$33 (\$53) | TCCR0    | -                                                    | -      | -      | -      | -      | CS02   | CS01   | CS00   | 5-34 |
| \$32 (\$52) | TCNT0    | Timer/Counter0 (8 Bit)                               |        |        |        |        |        |        |        | 5-35 |
| \$31 (\$51) | Reserved |                                                      |        |        |        |        |        |        |        |      |
| \$30 (\$50) | Reserved |                                                      |        |        |        |        |        |        |        |      |
| \$2F (\$4F) | TCCR1A   | COM1A1                                               | COM1A0 | COM1B1 | COM1B0 | -      | -      | PWM11  | PWM10  | 5-37 |
| \$2E (\$4E) | TCCR1B   | ICNC1                                                | ICES1  | -      | -      | CTC1   | CS12   | CS11   | CS10   | 5-38 |
| \$2D (\$4D) | TCNT1H   | Timer/Counter1 - Counter Register High Byte          |        |        |        |        |        |        |        | 5-39 |
| \$2C (\$4C) | TCNT1L   | Timer/Counter1 - Counter Register Low Byte           |        |        |        |        |        |        |        | 5-39 |
| \$2B (\$4B) | OCR1AH   | Timer/Counter1 - Output Compare Register A High Byte |        |        |        |        |        |        |        | 5-40 |
| \$2A (\$4A) | OCR1AL   | Timer/Counter1 - Output Compare Register A Low Byte  |        |        |        |        |        |        |        | 5-40 |
| \$29 (\$49) | OCR1BH   | Timer/Counter1 - Output Compare Register B High Byte |        |        |        |        |        |        |        | 5-40 |
| \$28 (\$48) | OCR1BL   | Timer/Counter1 - Output Compare Register B Low Byte  |        |        |        |        |        |        |        | 5-40 |
| \$27 (\$47) | Reserved |                                                      |        |        |        |        |        |        |        |      |
| \$26 (\$46) | Reserved |                                                      |        |        |        |        |        |        |        |      |
| \$25 (\$45) | ICR1H    | Timer/Counter1 - Input Capture Register High Byte    |        |        |        |        |        |        |        | 5-40 |
| \$24 (\$44) | ICR1L    | Timer/Counter1 - Input Capture Register Low Byte     |        |        |        |        |        |        |        | 5-40 |
| \$23 (\$43) | Reserved |                                                      |        |        |        |        |        |        |        |      |
| \$22 (\$42) | Reserved |                                                      |        |        |        |        |        |        |        |      |
| \$21 (\$41) | WDTCR    | -                                                    | -      | -      | WDTOE  | WDE    | WDP2   | WDP1   | WDP0   | 5-43 |
| \$20 (\$40) | Reserved |                                                      |        |        |        |        |        |        |        |      |
| \$1F (\$3F) | Reserved |                                                      |        |        |        |        |        |        |        |      |
| \$1E (\$3E) | EEARL    | EEPROM Address Register                              |        |        |        |        |        |        |        | 5-44 |
| \$1D (\$3D) | EEDR     | EEPROM Data Register                                 |        |        |        |        |        |        |        | 5-44 |
| \$1C (\$3C) | EECR     | -                                                    | -      | -      | -      | -      | EEMWE  | EERE   | EERE   | 5-45 |
| \$1B (\$3B) | PORTA    | PORTA7                                               | PORTA6 | PORTA5 | PORTA4 | PORTA3 | PORTA2 | PORTA1 | PORTA0 | 5-59 |
| \$1A (\$3A) | DDRA     | DDA7                                                 | DDA6   | DDA5   | DDA4   | DDA3   | DDA2   | DDA1   | DDA0   | 5-59 |
| \$19 (\$39) | PINA     | PINA7                                                | PINA6  | PINA5  | PINA4  | PINA3  | PINA2  | PINA1  | PINA0  | 5-59 |
| \$18 (\$38) | PORTB    | PORTB7                                               | PORTB6 | PORTB5 | PORTB4 | PORTB3 | PORTB2 | PORTB1 | PORTB0 | 5-61 |
| \$17 (\$37) | DDRB     | ddb7                                                 | ddb6   | ddb5   | ddb4   | ddb3   | ddb2   | ddb1   | ddb0   | 5-61 |
| \$16 (\$36) | PINB     | PINB7                                                | PINB6  | PINB5  | PINB4  | PINB3  | PINB2  | PINB1  | PINB0  | 5-61 |
| \$15 (\$35) | PORTC    | PORTC7                                               | PORTC6 | PORTC5 | PORTC4 | PORTC3 | PORTC2 | PORTC1 | PORTC0 | 5-66 |
| \$14 (\$34) | DDRC     | DDC7                                                 | DDC6   | DDC5   | DDC4   | DDC3   | DDC2   | DDC1   | DDC0   | 5-66 |
| \$13 (\$33) | PINC     | PINC7                                                | PINC6  | PINC5  | PINC4  | PINC3  | PINC2  | PINC1  | PINC0  | 5-66 |
| \$12 (\$32) | PORTD    | PORTD7                                               | PORTD6 | PORTD5 | PORTD4 | PORTD3 | PORTD2 | PORTD1 | PORTD0 | 5-68 |
| \$11 (\$31) | DDRD     | DDD7                                                 | DDD6   | DDD5   | DDD4   | DDD3   | DDD2   | DDD1   | DDD0   | 5-68 |
| \$10 (\$30) | PIND     | PIND7                                                | PIND6  | PIND5  | PIND4  | PIND3  | PIND2  | PIND1  | PIND0  | 5-68 |
| \$0F (\$2F) | SPDR     | SPI Data Register                                    |        |        |        |        |        |        |        | 5-50 |
| \$0E (\$2E) | SPSR     | SPIF                                                 | WCOL   | -      | -      | -      | -      | -      | -      | 5-49 |
| \$0D (\$2D) | SPCR     | SPIE                                                 | SPE    | DORD   | MSTR   | CPOL   | CPHA   | SPR1   | SPR0   | 5-49 |
| \$0C (\$2C) | UDR      | UART I/O Data Register                               |        |        |        |        |        |        |        | 5-53 |
| \$0B (\$2B) | USR      | RXC                                                  | TXC    | UDRE   | FE     | OR     | -      | -      | -      | 5-53 |
| \$0A (\$2A) | UCR      | RXCIE                                                | TXCIE  | UDRIE  | RXEN   | TXEN   | CHR9   | RXB8   | TXB8   | 5-54 |
| \$09 (\$29) | UBRR     | UART Baud Rate Register                              |        |        |        |        |        |        |        | 5-56 |
| \$08 (\$28) | ACSR     | ACD                                                  | -      | ACO    | ACI    | ACIE   | ACIC   | ACIS1  | ACIS0  | 5-57 |
| ...         | Reserved |                                                      |        |        |        |        |        |        |        |      |
| \$00 (\$20) | Reserved |                                                      |        |        |        |        |        |        |        |      |



## AT90S8515 Instruction Set Summary

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

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

