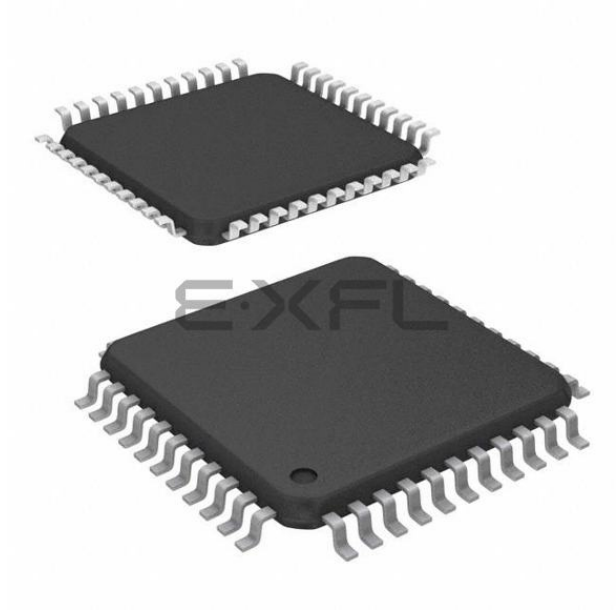


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Obsolete
Core Processor	AVR
Core Size	8-Bit
Speed	8MHz
Connectivity	SPI, UART/USART
Peripherals	PWM, WDT
Number of I/O	32
Program Memory Size	4KB (2K x 16)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	256 x 8
Voltage - Supply (Vcc/Vdd)	4V ~ 6V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	0°C ~ 70°C
Mounting Type	Surface Mount
Package / Case	44-TQFP
Supplier Device Package	44-TQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/at90s4414-8ac

The AVR uses a Harvard architecture concept - with separate memories and buses for program and data. The program memory is executed with a two stage pipeline. While one instruction is being executed, the next instruction is pre-fetched from the program memory. This concept enables instructions to be executed in every clock cycle. The program memory is in-system programmable Flash memory.

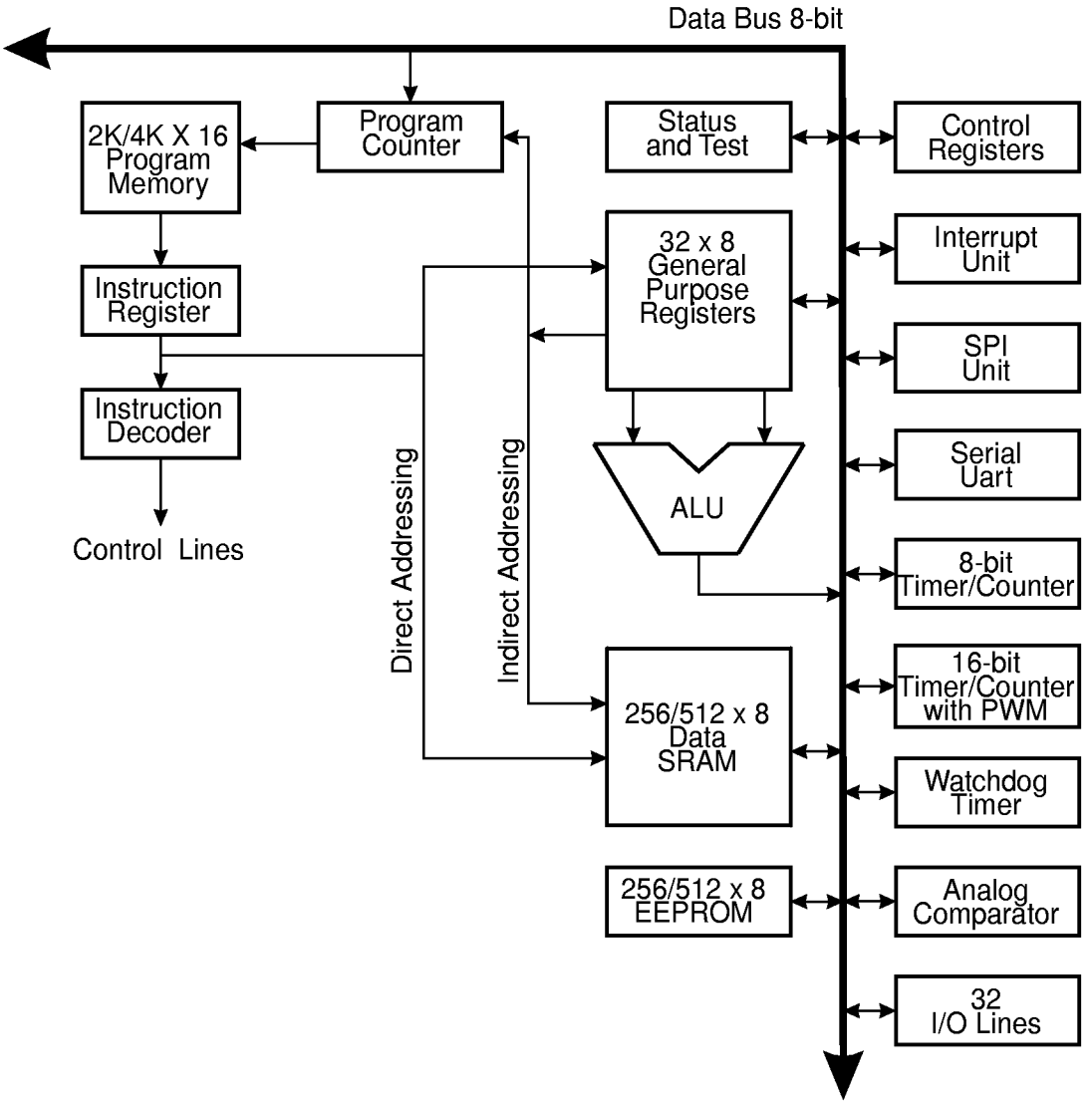
With the relative jump and call instructions, the whole 2K/4K address space is directly accessed. Most AVR instructions have a single 16-bit word format. Every program memory address contains a 16- or 32-bit instruction.

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 256/512 bytes data SRAM can be easily accessed through the five different addressing modes supported in the AVR architecture.

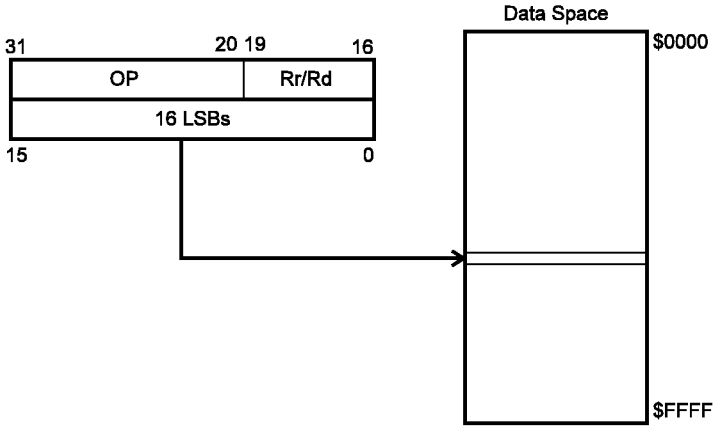
The memory spaces in the AVR architecture are all linear and regular memory maps.

Figure 4. The AT90S4414/8515 AVR RISC Architecture



Data Direct

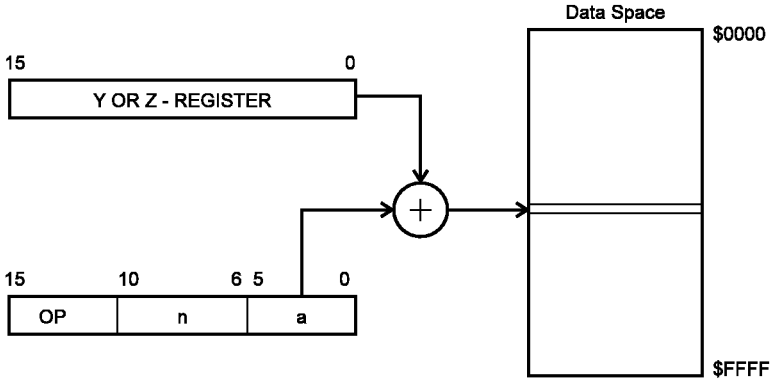
Figure 12. Direct Data Addressing



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

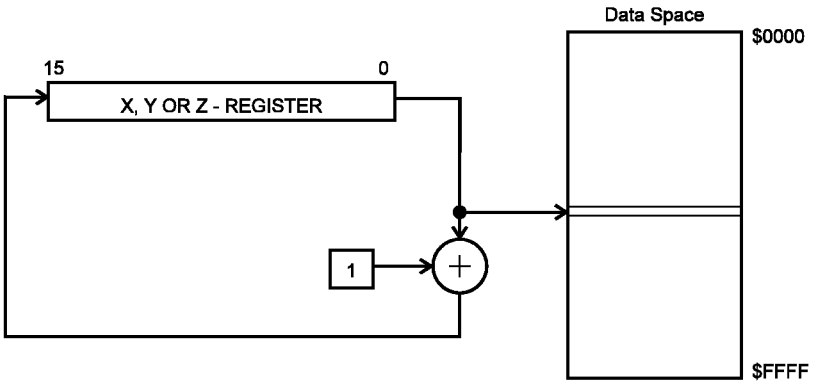
Figure 13. Data Indirect with Displacement



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 with Post-increment

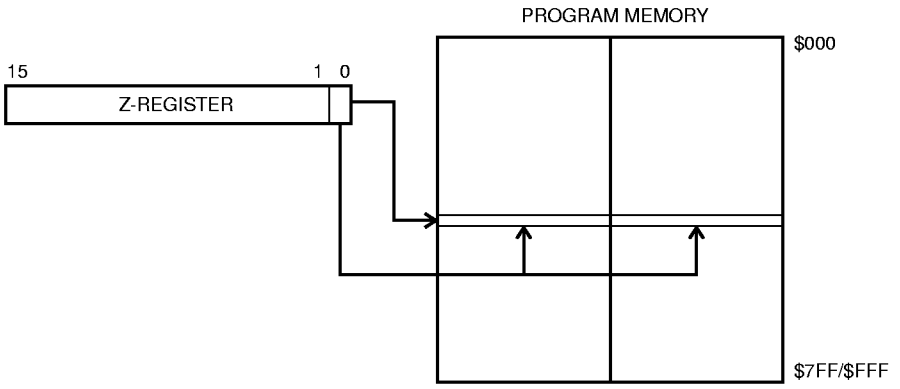
Figure 16. Data Indirect Addressing with Post-increment



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 - 2K/4K), the LSB selects low byte if cleared (LSB = 0) or high byte if set (LSB = 1).

Table 3. Reset and Interrupt Vectors

Vector No.	Program Address	Source	Interrupt Definition
1	$000	RESET	External Reset, Power-on Reset and Watchdog Reset
2	$001	INT0	External Interrupt Request 0
3	$002	INT1	External Interrupt Request 1
4	$003	TIMER1 CAPT	Timer/Counter1 Capture Event
5	$004	TIMER1 COMPA	Timer/Counter1 Compare Match A
6	$005	TIMER1 COMPB	Timer/Counter1 Compare Match B
7	$006	TIMER1 OVF	Timer/Counter1 Overflow
8	$007	TIMER0, OVF	Timer/Counter0 Overflow
9	$008	SPI, STC	Serial Transfer Complete
10	$009	UART, RX	UART, Rx Complete
11	$00A	UART, UDRE	UART Data Register Empty
12	$00B	UART, TX	UART, Tx Complete
13	$00C	ANA_COMP	Analog Comparator

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

```
Address          Labels      Code              Comments
$000                         rjmp    RESET     ; Reset Handler
$001                         rjmp    EXT_INT0  ; IRQ0 Handler
$002                         rjmp    EXT_INT1  ; IRQ1 Handler
$003                         rjmp    TIM1_CAPT ; Timer1 Capture Handler
$004                         rjmp    TIM1_COMPA ; Timer1 CompareA Handler
$005                         rjmp    TIM1_COMPB ; Timer1 CompareB Handler
$006                         rjmp    TIM1_OVF  ; Timer1 Overflow Handler
$007                         rjmp    TIM0_OVF  ; Timer0 Overflow Handler
$008                         rjmp    SPI_STC   ; SPI Transfer Complete Handler
$009                         rjmp    UART_RXC  ; UART RX Complete Handler
$00a                         rjmp    UART_DRE  ; UDR Empty Handler
$00b                         rjmp    UART_TXC  ; UART TX Complete Handler
$00c                         rjmp    ANA_COMP  ; Analog Comparator Handler
;
$00d             MAIN:       ldi r16,high(RAMEND); Main program start
$00e                         out SPH,r16
$00f                         ldi r16,low(RAMEND)
$010                         out SPL,r16
$011                         <instr>  xxx


  …                 …           …       …
```

Interrupt Handling

The AT90S4414/8515 has two 8-bit Interrupt Mask control registers; GIMSK - General Interrupt Mask register and TIMSK - Timer/Counter Interrupt Mask register.

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

For Interrupts triggered by events that can remain static (e.g. the Output Compare register1 matching the value of Timer/Counter1) the interrupt flag is set when the event occurs. If the interrupt flag is cleared and the interrupt condition persists, the flag will not be set until the event occurs the next time.

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

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

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

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

General Interrupt Mask Register - GIMSK

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

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

When the INT1 bit is set (one) and the I-bit in the Status Register (SREG) is set (one), the external pin interrupt is enabled. The Interrupt Sense Control1 bits 1/0 (ISC11 and ISC10) in the MCU general Control Register (MCUCR) 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 enabled. 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 AT90S4414/8515 and always read as zero.

General Interrupt Flag Register - GIFR

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

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

When an 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 - 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 A 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 B match Interrupt is executed.

- **Bit 4 - Res: Reserved bit**

This bit is a reserved bit in the AT90S4414/8515 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. When the SREG I-bit, and TICIE1 (Timer/Counter1 Input Capture Interrupt Enable), and ICF1 are set (one), the Timer/Counter1 Capture Interrupt is executed.

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

This bit is a reserved bit in the AT90S4414/8515 and always reads zero.

- **Bit 1 - TOV: Timer/Counter0 Overflow Flag**

The bit TOV0 is set (one) when an overflow occurs in Timer/Counter0. TOV0 is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, TOV0 is cleared by writing a 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 AT90S4414/8515 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

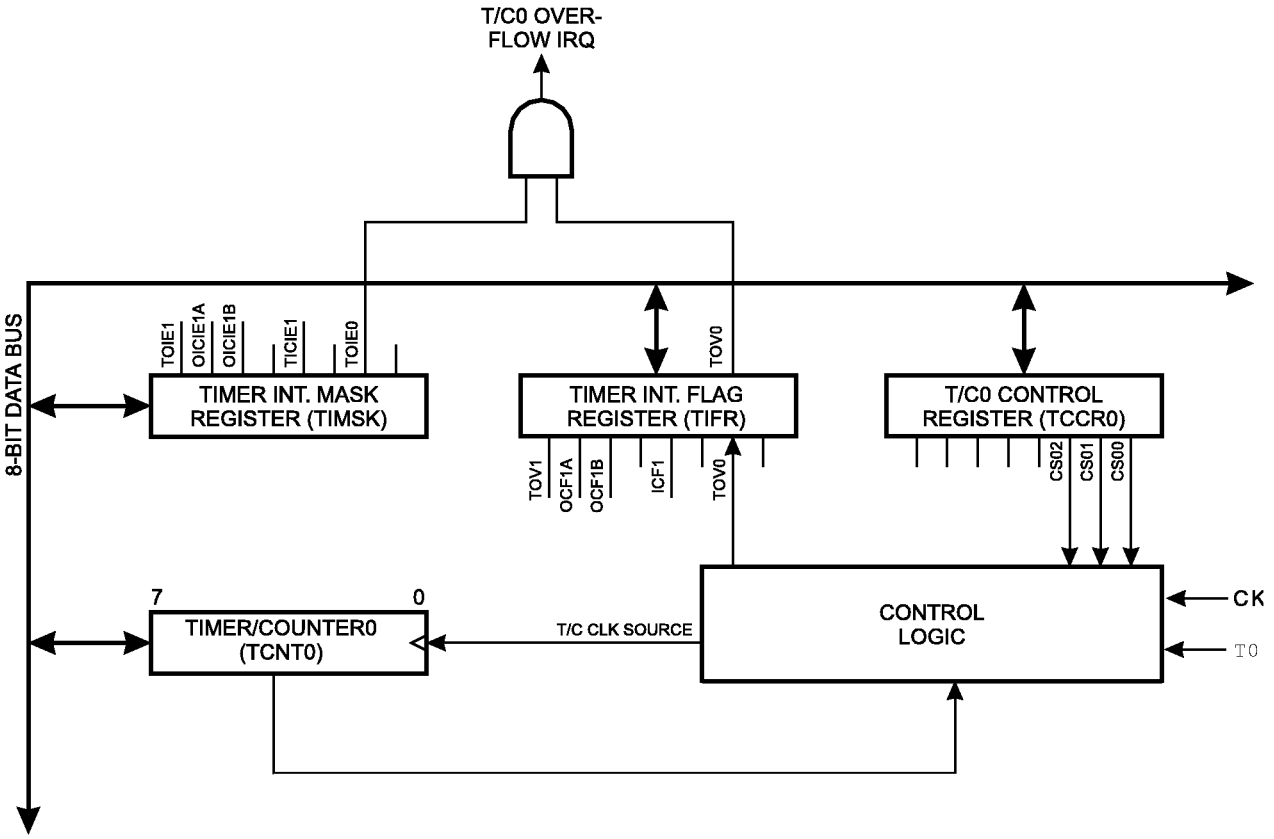
The interrupt execution response for all the enabled AVR interrupts is 4 clock cycles minimum. 4 clock cycles after the interrupt flag has been set, the program vector address for the actual interrupt handling routine is executed. During this 4 clock cycle period, the Program Counter (2 bytes) is pushed onto the Stack, and the Stack Pointer is decremented by 2. The vector is normally a relative jump to the interrupt routine, and this jump takes 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, the Stack Pointer is incremented by 2, and the I flag in SREG is set. When the AVR exits from an interrupt, it will always return to the main program and execute one more instruction before any pending interrupt is served.

Note that the Status Register - SREG - is not handled by the AVR hardware, 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. Note that an external level interrupt will only be remembered for as long as the interrupt condition is active.

Figure 29. Timer/Counter0 Block Diagram



Timer/Counter0 Control Register - TCCR0

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

- **Bits 7..3 - Res: Reserved bits**

These bits are reserved bits in the AT90S4414/8515 and always read as 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 Timer/Counter0.

Table 8. 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

EEPROM Data Register - EEDR

Bit	7	6	5	4	3	2	1	0	
$1D ($3D)	MSB							LSB	EEDR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

- **Bits 7..0 - EEDR7..0: EEPROM Data**

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

EEPROM Control Register - EECR

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

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

These bits are reserved bits in the AT90S4414/8515 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.

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

UART

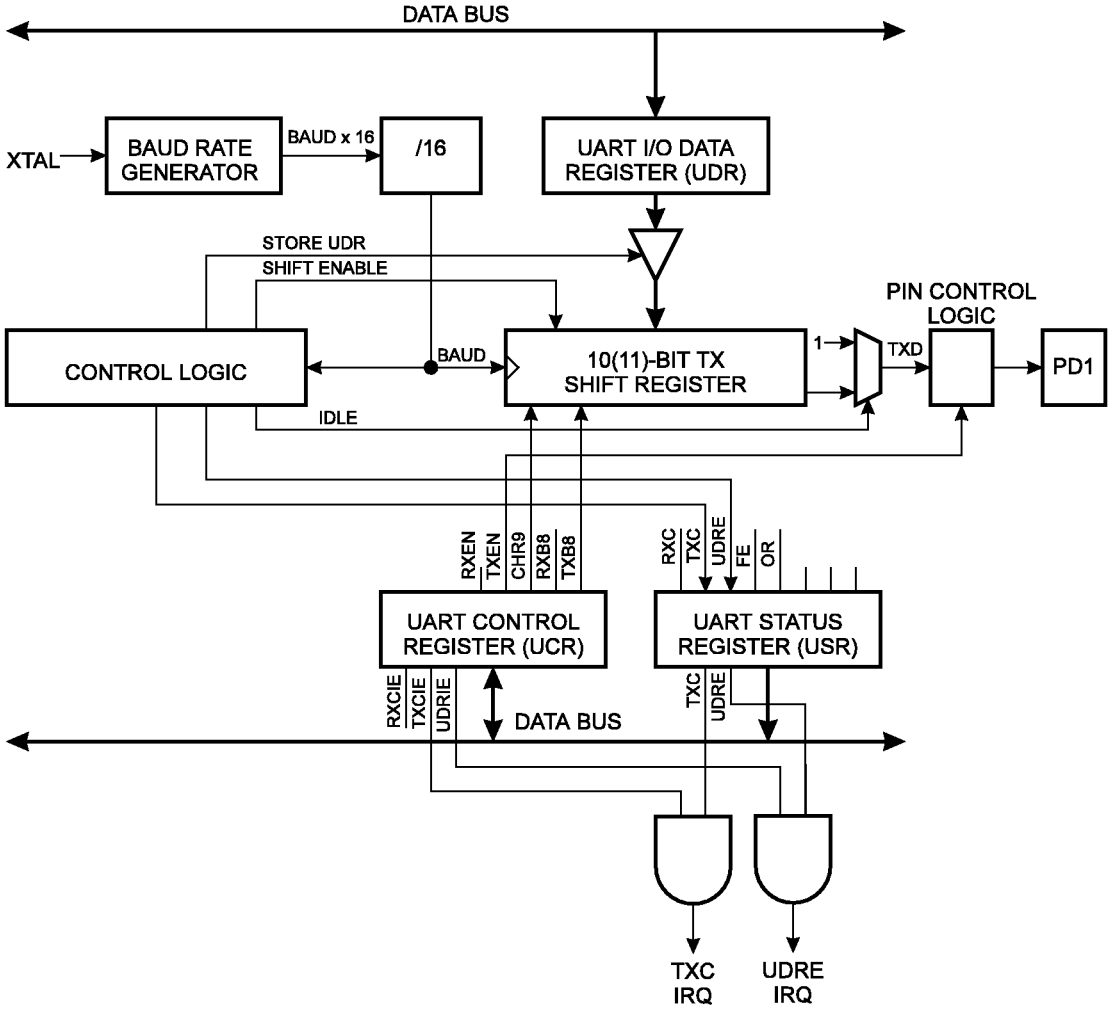
The AT90S4414/8515 features a full duplex (separate receive and transmit registers) Universal Asynchronous Receiver and Transmitter (UART). The main features are:

- Baud rate generator that can generate a large number of baud rates (bps)
- High baud rates at low XTAL frequencies
- 8 or 9 bits data
- Noise filtering
- Overrun detection
- Framing Error detection
- False Start Bit detection
- Three separate interrupts on TX Complete, TX Data Register Empty and RX Complete

Data Transmission

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

Figure 38. UART Transmitter



- **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 19.

Table 19. ACIS1/ACIS0 Settings

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

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

Interface to External SRAM

The interface to the SRAM consists of:

- Port A: Multiplexed low-order address bus and data bus
- Port C: High-order address bus
- The ALE-pin: Address latch enable
- The $\overline{RD}$ and $\overline{WR}$-pin: Read and write strobes.

The external data SRAM is enabled by setting the SRE - External SRAM enable bit of the MCUCR - MCU control register, and will override the setting of the data direction register DDRA. When the SRE bit is cleared (zero), the external data SRAM is disabled, and the normal pin and data direction settings are used. When SRE is cleared (zero), the address space above the internal SRAM boundary is not mapped into the internal SRAM, as in AVR parts not having interface to the external SRAM.

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

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

Figure 42 sketches how to connect an external SRAM to the AVR using 8 latches which are transparent when G is high.

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

For details in the timing for the SRAM interface, please refer to Figure 68, Table 38, Table 39, Table 40, and Table 41 in section “Absolute Maximum Ratings*” on page 81.

Figure 42. External SRAM Connected to the AVR

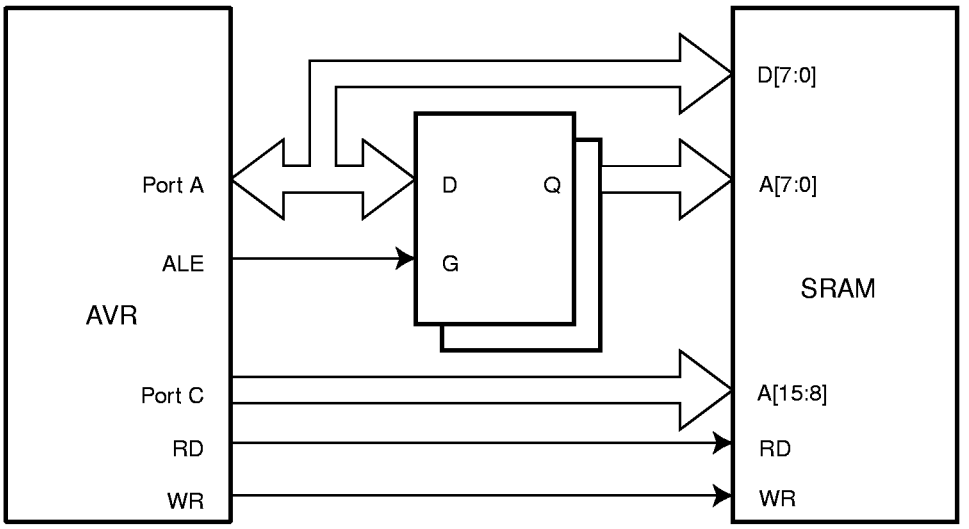


Figure 43. External Data SRAM Memory Cycles without Wait State

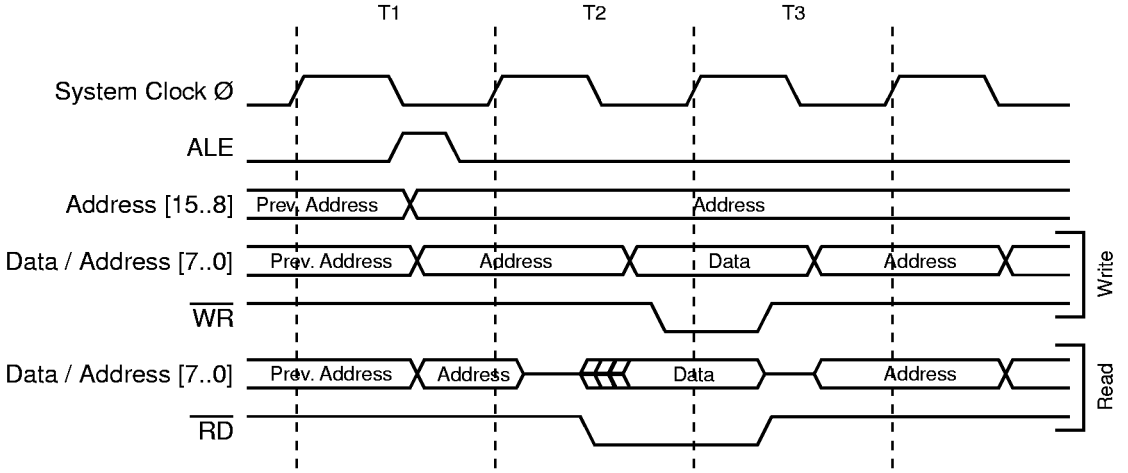
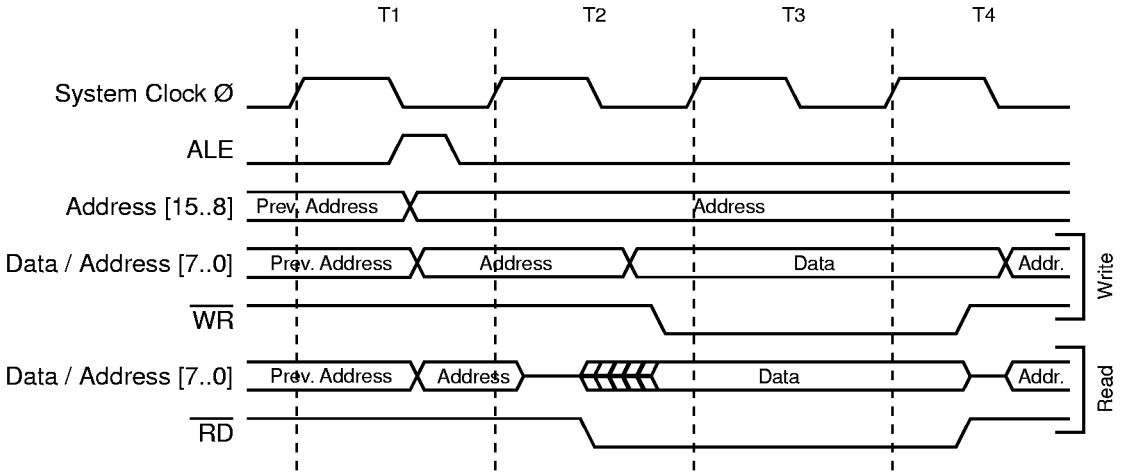


Figure 44. External Data SRAM Memory Cycles with Wait State



I/O-Ports

All AVR ports have true Read-Modify-Write functionality when used as general digital I/O ports. This means that the direction of one port pin can be changed without unintentionally changing the direction of any other pin with the SBI and CBI instructions. The same applies for changing drive value (if configured as output) or enabling/disabling of pull-up resistors (if configured as input).

Port A

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

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

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

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

When Port A is set to the alternate function by the SRE - External SRAM Enable - bit in the MCUCR - MCU Control Register, the alternate settings override the data direction register.

Port A Data Register - PORTA

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

Port A Data Direction Register - DDRA

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

Port A Input Pins Address - PINA

Bit	7	6	5	4	3	2	1	0	
$19 ($39)	**PINA7**	**PINA6**	**PINA5**	**PINA4**	**PINA3**	**PINA2**	**PINA1**	**PINA0**	**PINA**
Read/Write	R	R	R	R	R	R	R	R	
Initial value	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 Port A Data Latch is read, and when reading PINA, the logical values present on the pins are read.

Port A as General Digital I/O

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

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

Figure 56. Port D Schematic Diagram (Pin PD4)

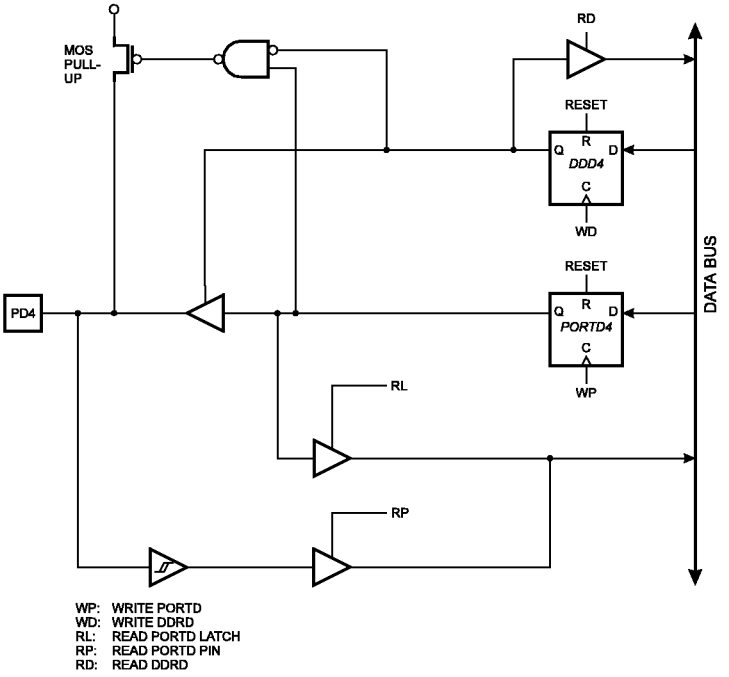
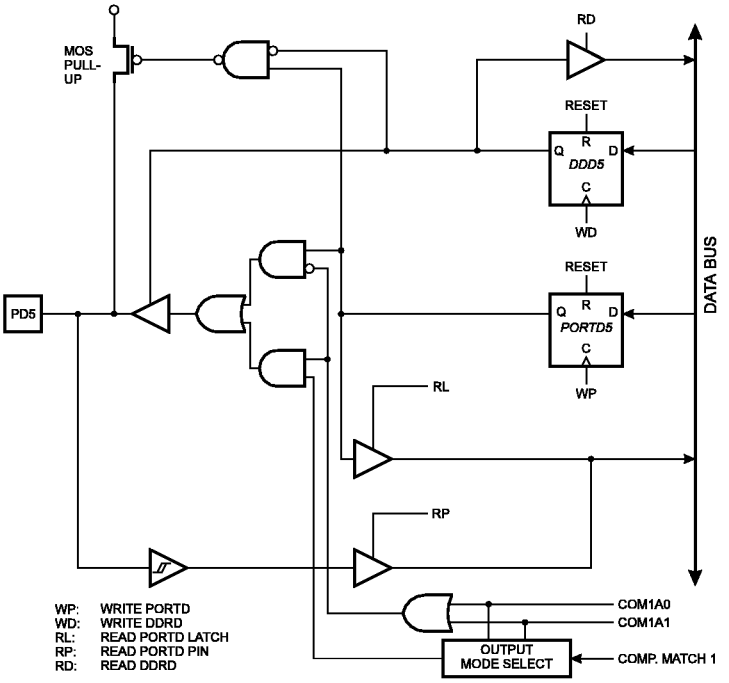


Figure 57. Port D Schematic Diagram (Pin PD5)



3. Wait until RDY/$\overline{BSY}$ goes high to program the next byte.

(See Figure 62 for signal waveforms.)

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

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

These considerations also applies to EEPROM programming, and Flash, EEPROM and Signature bytes reading.

Figure 61. Programming the Flash Waveforms

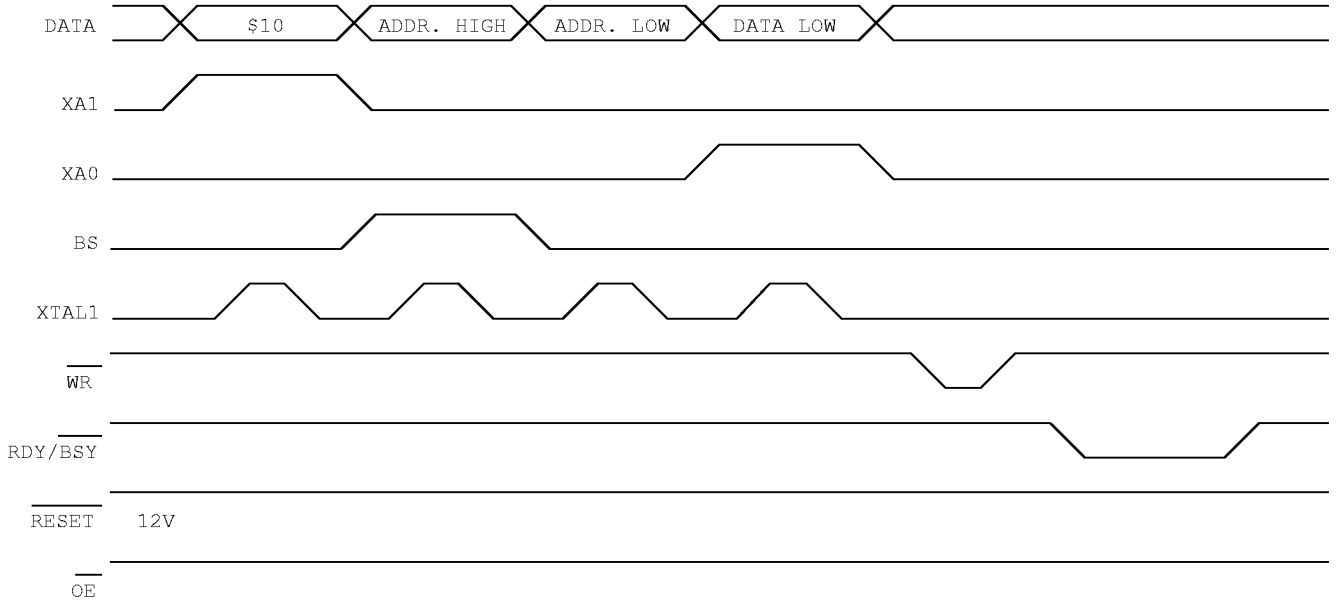
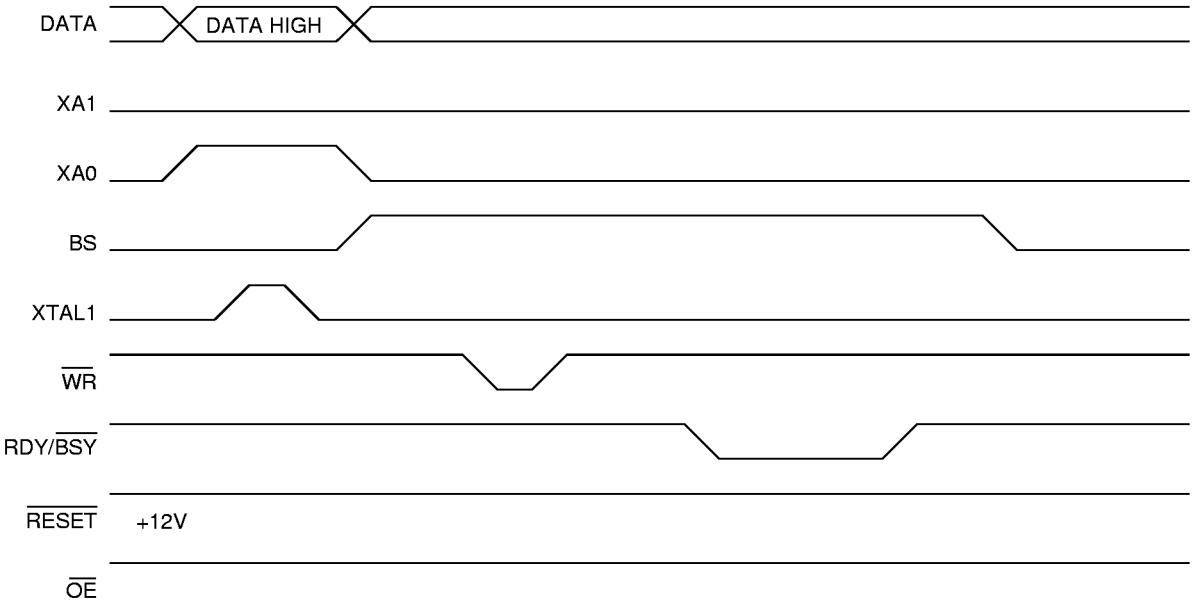


Figure 62. Programming the Flash Waveforms (continued)



Programming the Lock Bits

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

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

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

Reading the Fuse and Lock Bits

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

1. A: Load Command “0000 0100”.
2. Set $\overline{OE}$ to “0”, and BS to “1”. The status of the Fuse and Lock bits can now be read at DATA (“0” means programmed).
   Bit 7 = Lock Bit1
   Bit 6 = Lock Bit2
   Bit 5 = SPIEN Fuse bit
   Bit 0 = FSTRT Fuse bit
3. Set $\overline{OE}$ to “1”.

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

Reading the Signature Bytes

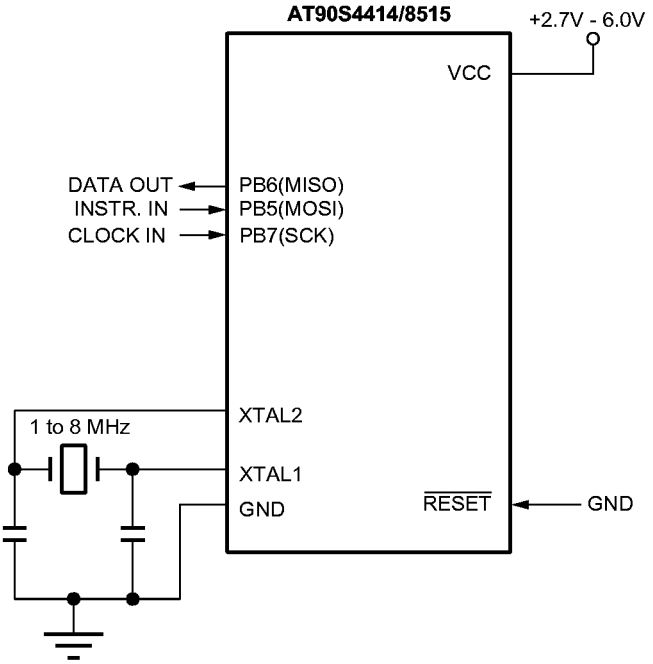
The algorithm for reading the Signature bytes is as follows (refer to Programming the Flash for details on Command and Address loading):

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

Serial Downloading

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

Figure 64. Serial Programming and Verify



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

The Program and EEPROM memory arrays have separate address spaces:

$0000 to $07FF/$0FFF (AT90S4414/8515) for Program memory and $0000 to $00FF/$01FF (AT90S4414/8515) for EEPROM memory.

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

Low: > 2 XTAL1 clock cycles

High: > 2 XTAL1 clock cycles

DC Characteristics

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

Symbol	Parameter	Condition	Min	Typ	Max	Units
V_{IL}	Input Low Voltage	(Except XTAL1)	-0.5		0.3 V_{CC}(1)	V
V_{IL1}	Input Low Voltage	(XTAL1)	-0.5		0.2 V_{CC}(1)	V
V_{IH}	Input High Voltage	(Except XTAL1, $\overline{RESET}$)	0.6 V_{CC}(2)		V_{CC} + 0.5	V
V_{IH1}	Input High Voltage	(XTAL1)	0.8 V_{CC}(2)		V_{CC} + 0.5	V
V_{IH2}	Input High Voltage	($\overline{RESET}$)	0.9 V_{CC}(2)		V_{CC} + 0.5	V
V_{OL}	Output Low Voltage(3) (Ports A,B,C,D)	I_{OL} = 20 mA, V_{CC} = 5V I_{OL} = 10 mA, V_{CC} = 3V			0.6 0.5	V V
V_{OH}	Output High Voltage(4) (Ports A,B,C,D)	I_{OH} = -3 mA, V_{CC} = 5V I_{OH} = -1.5 mA, V_{CC} = 3V	4.2 2.3			V V
I_{IL}	Input Leakage Current I/O pin	Vcc = 6V, pin low (absolute value)			8.0	µA
I_{IH}	Input Leakage Current I/O pin	Vcc = 6V, pin high (absolute value)			980	nA
RRST	Reset Pull-Up Resistor		100		500	kΩ
$R_{I/O}$	I/O Pin Pull-Up Resistor		35		120	kΩ
I_{CC}	Power Supply Current	Active Mode, V_{CC} = 3V, 4MHz			3.0	mA
		Idle Mode V_{CC} = 3V, 4MHz			1.2	mA
	Power Down Mode(5)	WDT enabled, V_{CC} = 3V		9	15.0	µA
		WDT disabled, V_{CC} = 3V		<1	2.0	µA
V_{ACIO}	Analog Comparator Input Offset Voltage	V_{CC} = 5V			40	mV
I_{ACLK}	Analog Comparator Input Leakage Current	V_{CC} = 5V V_{in} = V_{CC}/2	-50		50	nA
t_{ACPD}	Analog Comparator Propagation Delay	V_{CC} = 2.7V V_{CC} = 4.0V		750 500		ns

Notes:
1. "Max" means the highest value where the pin is guaranteed to be read as low.
2. "Min" means the lowest value where the pin is guaranteed to be read as high.
3. Although each I/O port can sink more than the test conditions (20mA at Vcc = 5V, 10mA at Vcc = 3V) under steady state conditions (non-transient), the following must be observed:
   1) The sum of all IOL, for all ports, should not exceed 200 mA.
   2) The sum of all IOL, for ports B0-B7, D0-D7 and XTAL2, should not exceed 100 mA.
   3) The sum of all IOL, for ports A0-A7, ALE, OC1B and C0-C7 should not exceed 100 mA.
   If IOL exceeds the test condition, VOL may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test condition.
4. Although each I/O port can source more than the test conditions (3mA at Vcc = 5V, 1.5mA at Vcc = 3V) under steady state conditions (non-transient), the following must be observed:
   1) The sum of all IOH, for all ports, should not exceed 200 mA.
   2) The sum of all IOH, for ports B0-B7, D0-D7 and XTAL2, should not exceed 100 mA.
   3) The sum of all IOH, for ports A0-A7, ALE, OC1B and C0-C7 should not exceed 100 mA.
   If IOH exceeds the test condition, VOH may exceed the related specification. Pins are not guaranteed to source current greater than the listed test condition.
5. Minimum V_{CC} for Power Down is 2V.

Table 40. External Data Memory Characteristics, 2.7 - 4.0 Volts, No Wait State

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

Table 41. External Data Memory Characteristics, 2.7 - 4.0 Volts, 1 Cycle Wait State

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

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

Figure 84. I/O Pin Source Curent vs. Output Voltage

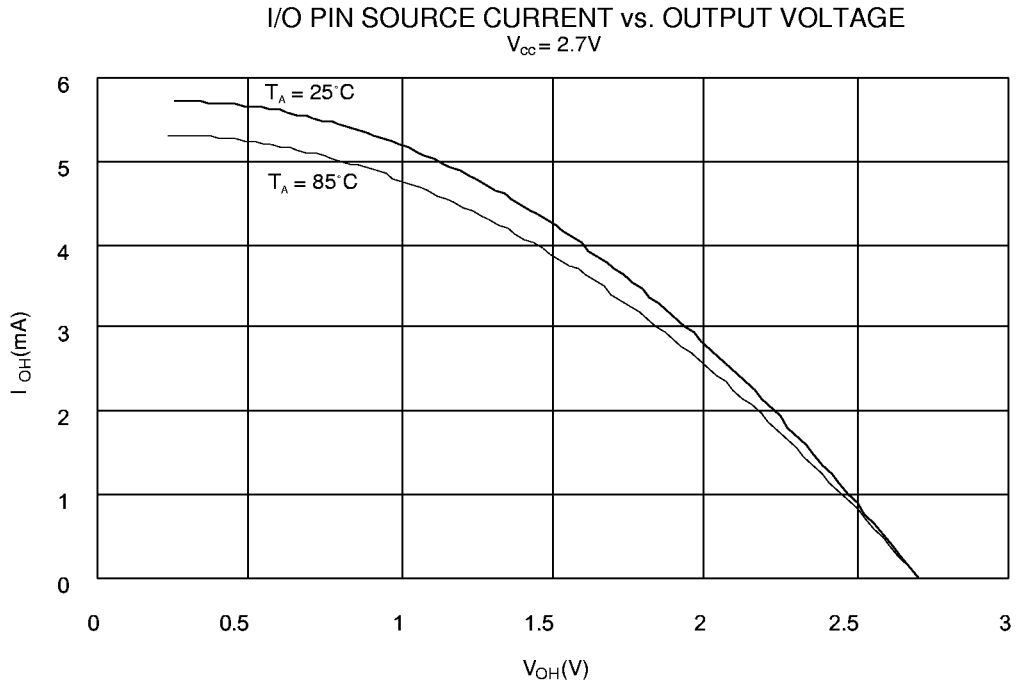
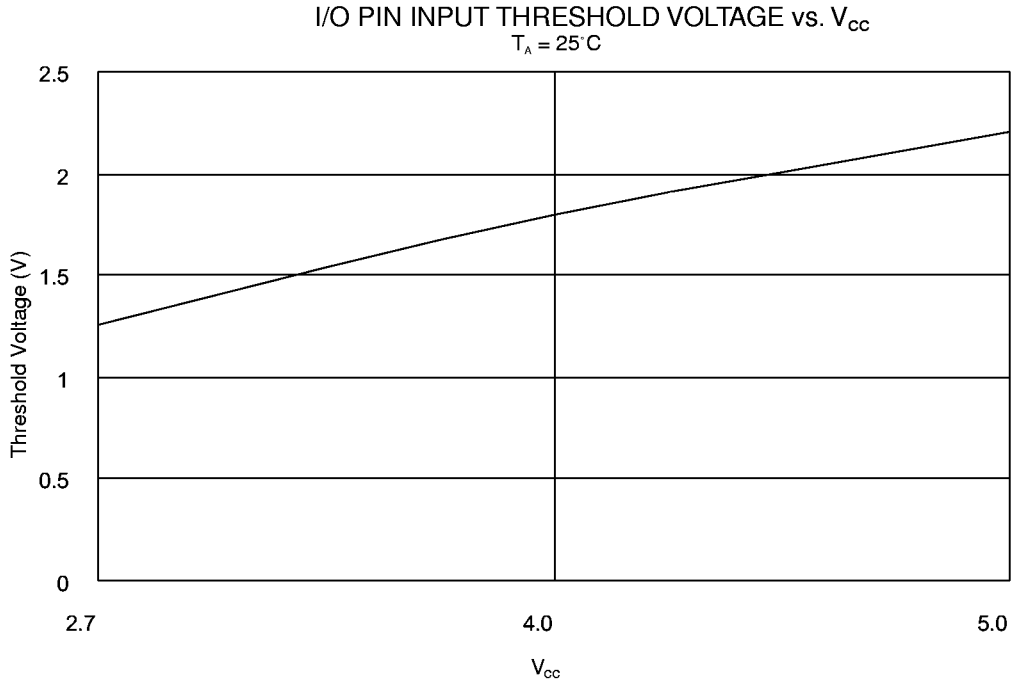


Figure 85. I/O Pin Input Threshold Voltage vs. V_{CC}



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	18
$3E ($5E)	SPH	SP15	SP14	SP13	SP12	SP11	SP10	SP9	SP8	19
$3D ($5D)	SPL	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0	19
$3C ($5C)	Reserved									
$3B ($5B)	GIMSK	INT1	INT0	-	-	-	-	-	-	24
$3A ($5A)	GIFR	INTF1	INTF0							24
$39 ($59)	TIMSK	TOIE1	OCIE1A	OCIE1B	-	TICIE1	-	TOIE0	-	25
$38 ($58)	TIFR	TOV1	OCF1A	OCF1B	-	ICF1	-	TOV0	-	25
$37 ($57)	Reserved									
$36 ($56)	Reserved									
$35 ($55)	MCUCR	SRE	SRW	SE	SM	ISC11	ISC10	ISC01	ISC00	27
$34 ($54)	Reserved									
$33 ($53)	TCCR0	-	-	-	-	-	CS02	CS01	CS00	30
$32 ($52)	TCNT0	Timer/Counter0 (8 Bit)								31
...	Reserved									
$2F ($4F)	TCCR1A	COM1A1	COM1A0	COM1B1	COM1B0	-	-	PWM11	PWM10	32
$2E ($4E)	TCCR1B	ICNC1	ICES1	-	-	CTC1	CS12	CS11	CS10	33
$2D ($4D)	TCNT1H	Timer/Counter1 - Counter Register High Byte								34
$2C ($4C)	TCNT1L	Timer/Counter1 - Counter Register Low Byte								34
$2B ($4B)	OCR1AH	Timer/Counter1 - Output Compare Register A High Byte								35
$2A ($4A)	OCR1AL	Timer/Counter1 - Output Compare Register A Low Byte								35
$29 ($49)	OCR1BH	Timer/Counter1 - Output Compare Register B High Byte								35
$28 ($48)	OCR1BL	Timer/Counter1 - Output Compare Register B Low Byte								35
...	Reserved									
$25 ($45)	ICR1H	Timer/Counter1 - Input Capture Register High Byte								36
$24 ($44)	ICR1L	Timer/Counter1 - Input Capture Register Low Byte								36
...	Reserved									
$21 ($41)	WDTCR	-	-	-	WDTOE	WDE	WDP2	WDP1	WDP0	38
$20 ($40)	Reserved									
$1F ($3F)	EEARH[1]	-	-	-	-	-	-	-	EEAR8	39
$1E ($3E)	EEARL	EEPROM Address Register Low Byte								39
$1D ($3D)	EEDR	EEPROM Data Register								40
$1C ($3C)	EECR	-	-	-	-	-	EEMWE	EEWE	EERE	40
$1B ($3B)	PORTA	PORTA7	PORTA6	PORTA5	PORTA4	PORTA3	PORTA2	PORTA1	PORTA0	55
$1A ($3A)	DDRA	DDA7	DDA6	DDA5	DDA4	DDA3	DDA2	DDA1	DDA0	55
$19 ($39)	PINA	PINA7	PINA6	PINA5	PINA4	PINA3	PINA2	PINA1	PINA0	55
$18 ($38)	PORTB	PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0	57
$17 ($37)	DDRB	DDB7	DDB6	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0	57
$16 ($36)	PINB	PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0	57
$15 ($35)	PORTC	PORTC7	PORTC6	PORTC5	PORTC4	PORTC3	PORTC2	PORTC1	PORTC0	62
$14 ($34)	DDRC	DDC7	DDC6	DDC5	DDC4	DDC3	DDC2	DDC1	DDC0	62
$13 ($33)	PINC	PINC7	PINC6	PINC5	PINC4	PINC3	PINC2	PINC1	PINC0	62
$12 ($32)	PORTD	PORTD7	PORTD6	PORTD5	PORTD4	PORTD3	PORTD2	PORTD1	PORTD0	64
$11 ($31)	DDRD	DDD7	DDD6	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0	64
$10 ($30)	PIND	PIND7	PIND6	PIND5	PIND4	PIND3	PIND2	PIND1	PIND0	64
$0F ($2F)	SPDR	SPI Data Register								45
$0E ($2E)	SPSR	SPIF	WCOL	-	-	-	-	-	-	45
$0D ($2D)	SPCR	SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR0	44
$0C ($2C)	UDR	UART I/O Data Register								48
$0B ($2B)	USR	RXC	TXC	UDRE	FE	OR	-	-	-	49
$0A ($2A)	UCR	RXCIE	TXCIE	UDRIE	RXEN	TXEN	CHR9	RXB8	TXB8	49
$09 ($29)	UBRR	UART Baud Rate Register								51
$08 ($28)	ACSR	ACD	-	ACO	ACI	ACIE	ACIC	ACIS1	ACIS0	52
...	Reserved									
$00 ($20)	Reserved									

Notes: 1. EEARH only present for AT90S8515

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

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