Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

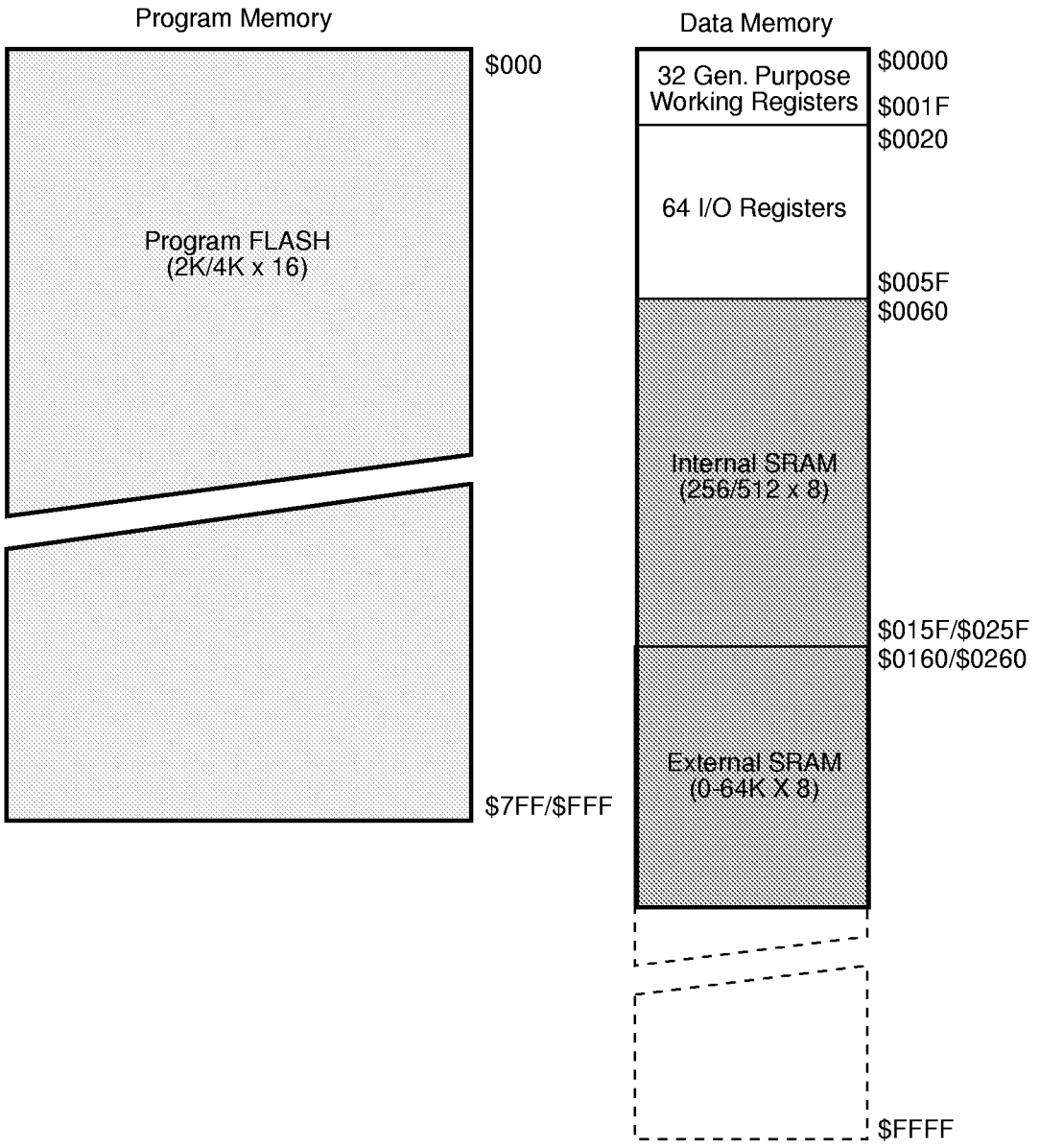
"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" 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	-40°C ~ 85°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/at90s4414-8ai

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

Figure 5. Memory Maps



ALU - Arithmetic Logic Unit

The high-performance AVR ALU operates in direct connection with all the 32 general purpose working registers. Within a single clock cycle, ALU operations between registers in the register file are executed. The ALU operations are divided into three main categories - arithmetic, logical and bit-functions.

In-System Programmable Flash Program Memory

The AT90S4414/8515 contains 4K/8K bytes on-chip In-System Programmable Flash memory for program storage. Since all instructions are 16-or 32-bit words, the Flash is organized as 2K x 16/4K x 16. The Flash memory has an endurance of at least 1000 write/erase cycles. The AT90S4414/8515 Program Counter (PC) is 11/12 bits wide, thus addressing the 2048/4096 program memory addresses.

See page 77 for a detailed description on Flash data downloading.

See page 10 for the different program memory addressing modes.

SRAM Data Memory - Internal and External

The following figure shows how the AT90S4414/8515 SRAM Memory is organized:

Figure 8. SRAM Organization

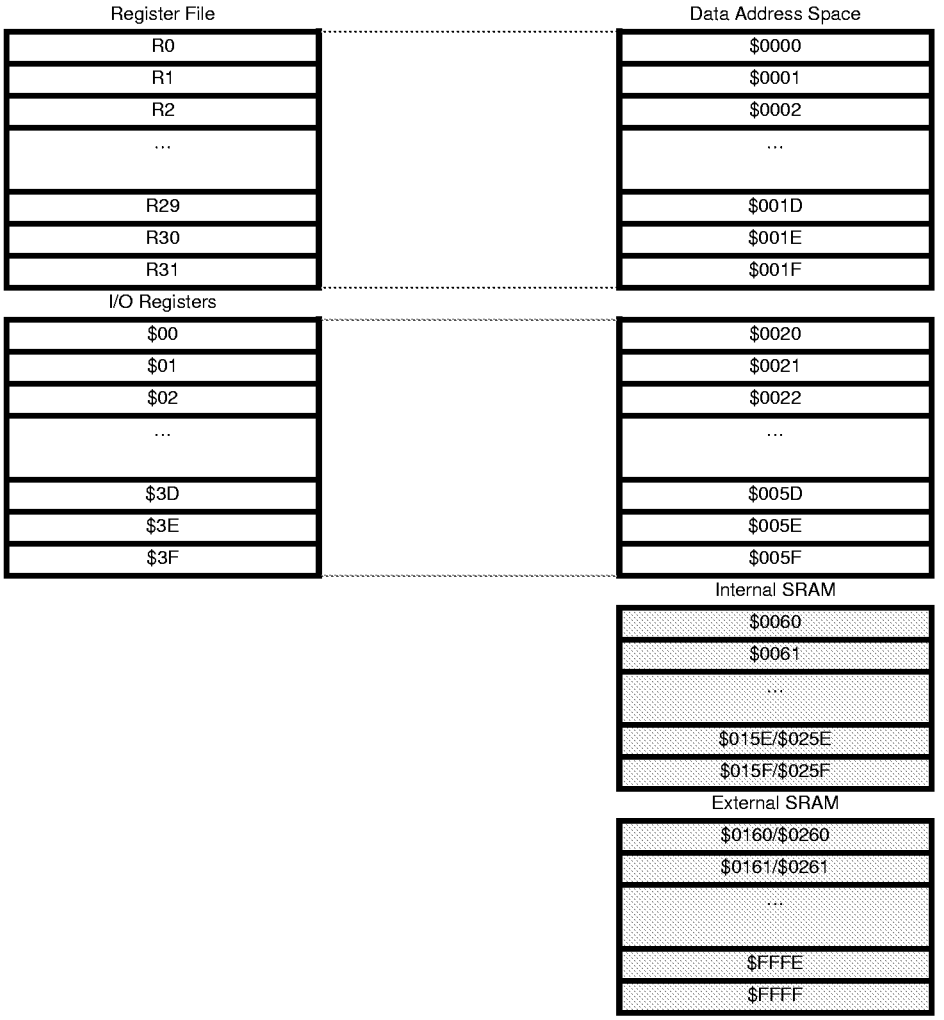


Figure 20. The Parallel Instruction Fetches and Instruction Executions

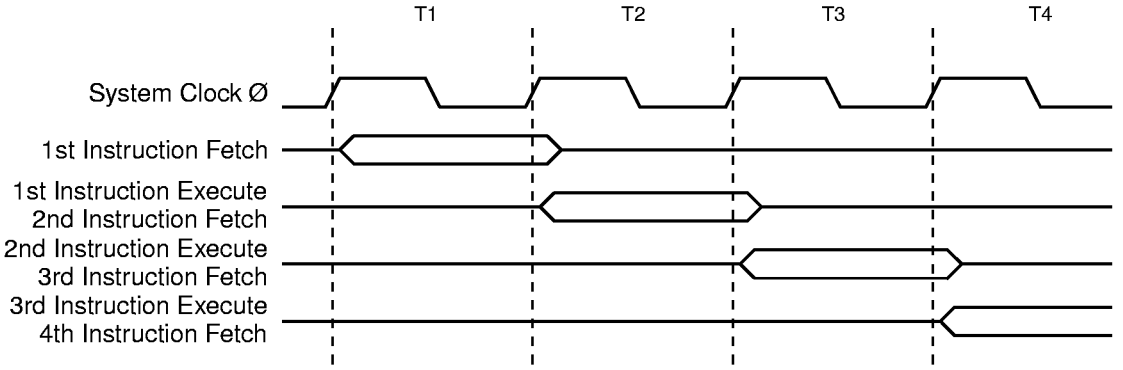
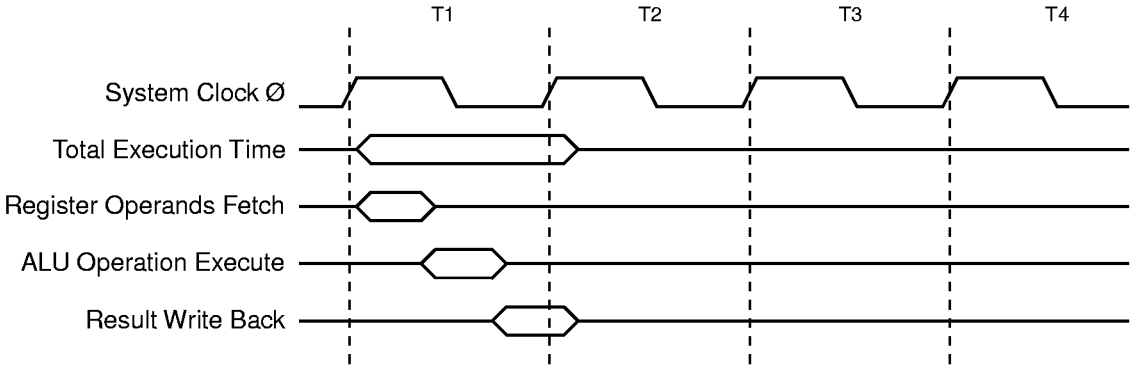


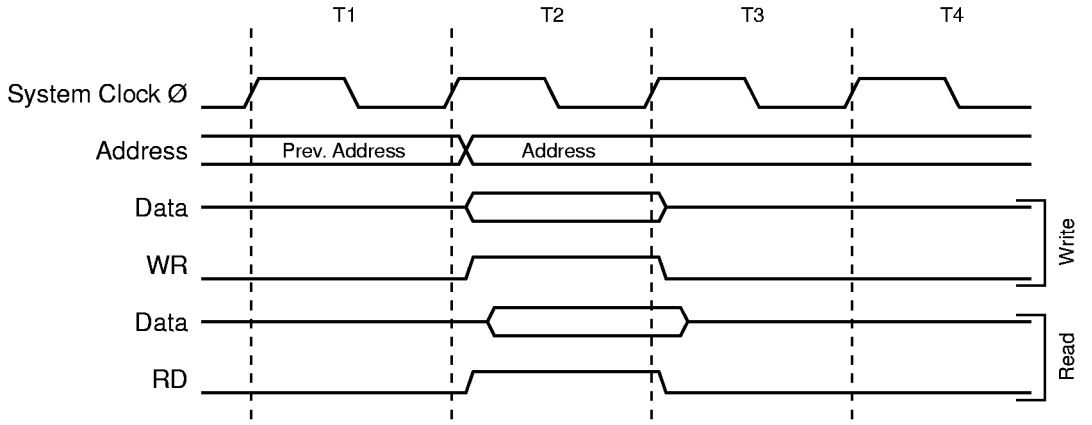
Figure 21 shows the internal timing concept for the register file. In a single clock cycle an ALU operation using two register operands is executed, and the result is stored back to the destination register.

Figure 21. Single Cycle ALU Operation



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

Figure 22. On-chip Data SRAM Access Cycles

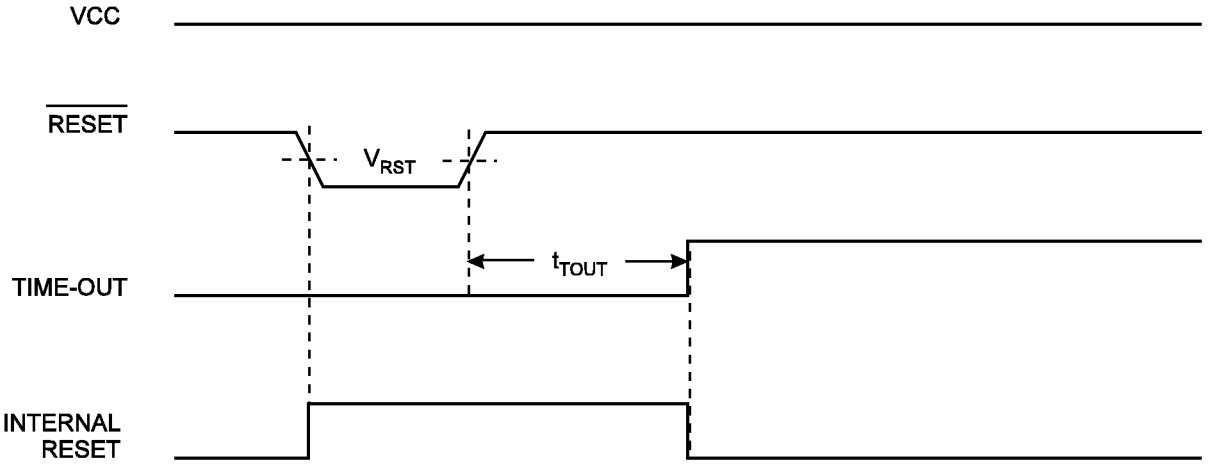


See “Interface to External SRAM” on page 53 for a description of the access to the external SRAM.

External Reset

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

Figure 26. External Reset During Operation



Watchdog Reset

When the Watchdog times out, it will generate a short reset pulse of 1 XTAL cycle duration. On the falling edge of this pulse, the delay timer starts counting the Time-out period t_{TOUT}. Refer to page 38 for details on operation of the Watchdog.

Figure 27. Watchdog Reset During Operation

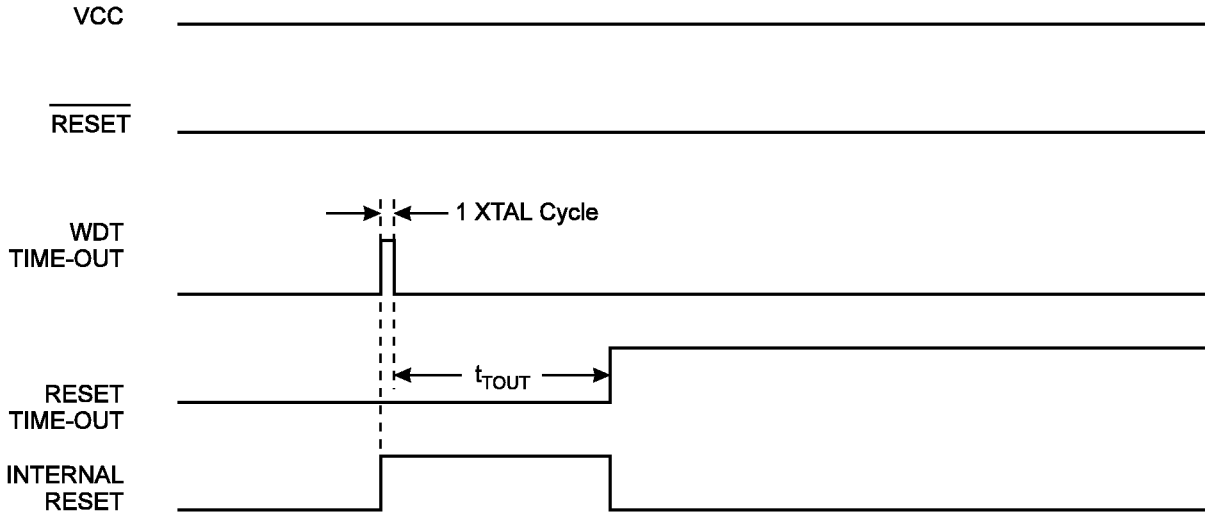


Table 7. Interrupt 0 Sense Control

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 ISC01/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.

The value on the INTn pin is sampled before detecting edges. If edge interrupt is selected, pulses with a duration longer than one CPU clock period will generate an interrupt. Shorter pulses are not guaranteed to generate an interrupt. If low level interrupt is selected, the low level must be held until the completion of the currently executing instruction to generate an interrupt. If enabled, a level triggered interrupt will generate an interrupt request as long as the pin is held low.

Sleep Modes

To enter the sleep modes, the SE bit in MCUCR must be set (one) and a SLEEP instruction must be executed. If an enabled interrupt occurs while the MCU is in a sleep mode, the MCU awakes, executes the interrupt routine, and resumes execution from the instruction following SLEEP. The contents of the register file, SRAM and I/O memory are unaltered. If a reset occurs during sleep mode, the MCU wakes up and executes from the Reset vector.

Idle Mode

When the SM bit is cleared (zero), the SLEEP instruction forces the MCU into the Idle Mode stopping the CPU but allowing Timer/Counters, Watchdog and the interrupt system to continue operating. This enables the MCU to wake up from external triggered interrupts as well as internal ones like Timer Overflow interrupt and watchdog reset. If wakeup from the Analog Comparator interrupt is not required, the analog comparator can be powered down by setting the ACD-bit in the Analog Comparator Control and Status register - ACSR. This will reduce power consumption in Idle Mode. When the MCU wakes up from Idle mode, the CPU starts program execution immediately.

Power Down Mode

When the SM bit is set (one), the SLEEP instruction forces the MCU into the Power Down Mode. In this mode, the external oscillator is stopped, while the external interrupts and the Watchdog (if enabled) continue operating. Only an external reset, a watchdog reset (if enabled), an external level interrupt on INT0 or INT1, can wake up the MCU.

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

Timer/Counters

The AT90S4414/8515 provides two general purpose Timer/Counters - one 8-bit T/C and one 16-bit T/C. The Timer/Counters have individual prescaling selection from the same 10-bit prescaling timer. Both Timer/Counters can either be used as a timer with an internal clock timebase or as a counter with an external pin connection which triggers the counting.

Timer/Counter1 Input Capture Register - ICR1H AND ICR1L

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

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

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

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

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

Timer/Counter1 In PWM Mode

When the PWM mode is selected, Timer/Counter1 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 12), where it turns and counts down again to zero before the cycle is repeated. When the counter value matches the contents of the 10 least significant bits of OCR1A or OCR1B, the PD5(OC1A)/OC1B pins are set or cleared according to the settings of the COM1A1/COM1A0 or COM1B1/COM1B0 bits in the Timer/Counter1 Control Register TCCR1A. Refer to Table 13 for details.

Table 12. Timer TOP Values and PWM Frequency

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

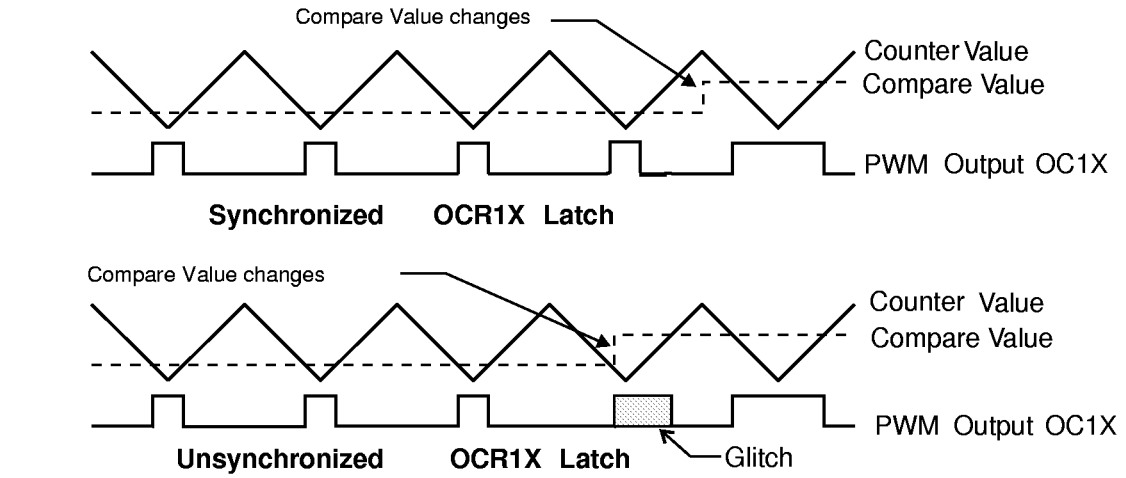
Table 13. Compare1 Mode Select in PWM Mode

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

Note: X = A or B

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

Figure 32. Effects on Unsynchronized OCR1 Latching



Note: X = A or B

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

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

Table 14. PWM Outputs OCR1X = $0000 or TOP

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

Note: X = A or B

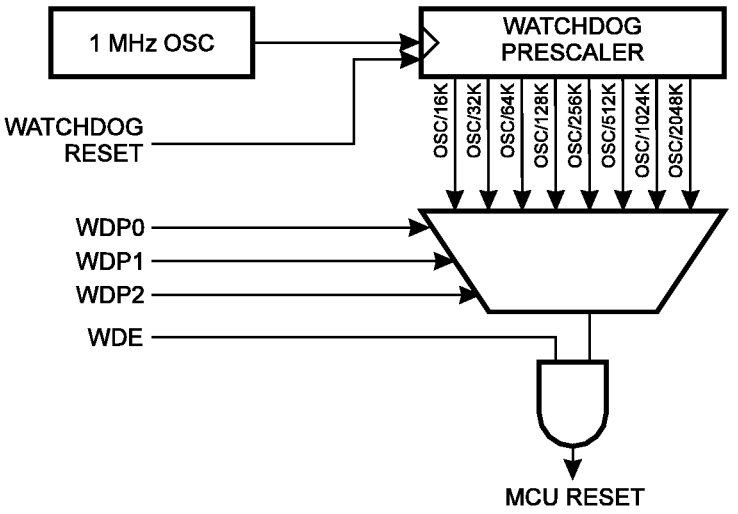
In PWM mode, the Timer Overflow Flag1, TOV1, is set when the counter advances from $0000. Timer Overflow Interrupt1 operates exactly as in normal Timer/Counter mode, i.e. it is executed when TOV1 is set provided that Timer Overflow Interrupt1 and global interrupts are enabled. This does also apply to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

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, see Table 15 for a detailed description. The WDR - Watchdog Reset - instruction resets the Watchdog Timer. Eight different clock cycle periods can be selected to determine the reset period. If the reset period expires without another Watchdog reset, the AT90S4414/8515 resets and executes from the reset vector. For timing details on the Watchdog reset, refer to page 23.

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

Figure 33. Watchdog Timer



Watchdog Timer Control Register - WDTCR

Bit	7	6	5	4	3	2	1	0	
$21 ($41)	-	-	-	WDTOE	WDE	WDP2	WDP1	WDP0	WDTCR
Read/Write	R	R	R	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

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

These bits are reserved bits in the AT90S4414/8515 and will always read as zero.

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

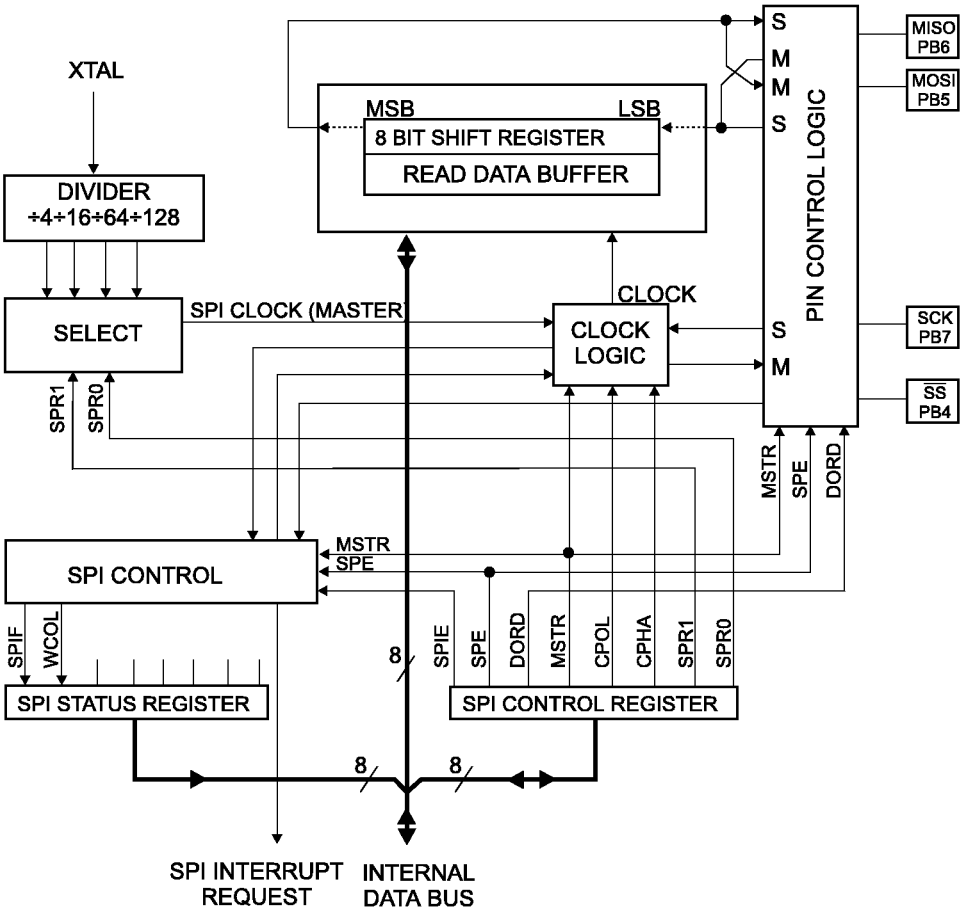
This bit must be set (one) when the WDE bit is cleared. Otherwise, the watchdog will not be disabled. Once set, hardware will clear this bit to zero after four clock cycles. Refer to the description of the WDE bit for a watchdog disable procedure.

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

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

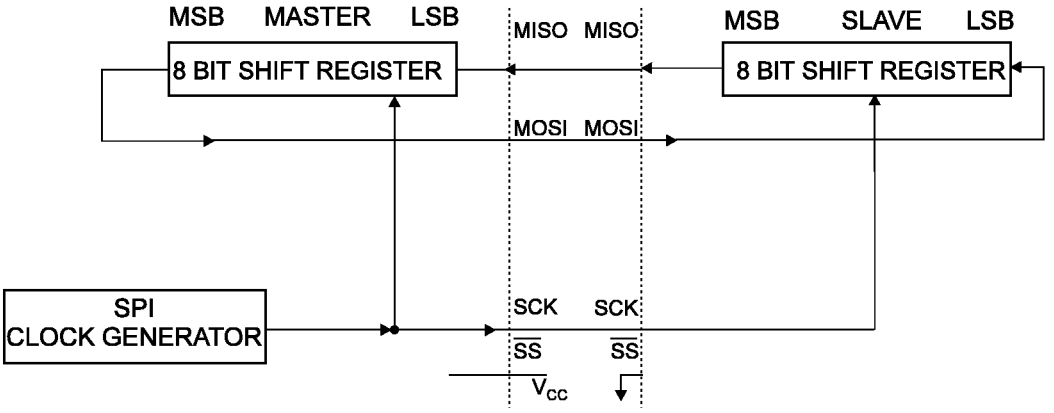
1. In the same operation, write a logical one to WDTOE and WDE. A logical one must be written to WDE even though it is set to one before the disable operation starts.
2. Within the next four clock cycles, write a logical 0 to WDE. This disables the watchdog.

Figure 34. SPI Block Diagram



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

Figure 35. SPI Master-slave Interconnection



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

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

Table 16. SPI Pin Overrides

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

Note: See "Alternate Functions of PortB" on page 58 for a detailed description of how to define the direction of the user defined SPI pins.

$\overline{SS}$ Pin Functionality

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

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

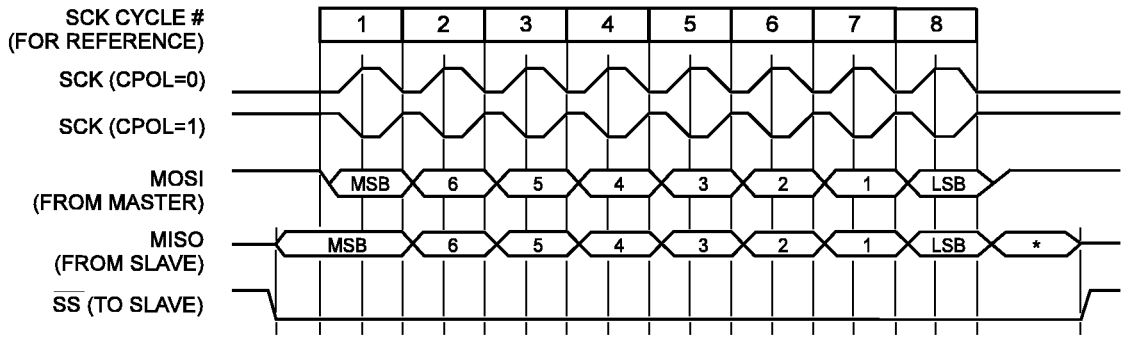
Thus, when interrupt-driven SPI transmittal is used in master mode, and there exists a possibility that $\overline{SS}$ is driven low, the interrupt should always check that the MSTR bit is still set. Once the MSTR bit has been cleared by a slave select, it must be set by the user to re-enable SPI master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

When the SPI is configured as a slave, the $\overline{SS}$ pin is always input. When $\overline{SS}$ is held low, the SPI is activated and MISO becomes an output if configured so by the user. All other pins are inputs. When $\overline{SS}$ is driven high, all pins are inputs, and the SPI is passive, which means that it will not receive incoming data. Note that the SPI logic will be reset once the $\overline{SS}$ pin is brought high. If the $\overline{SS}$ pin is brought high during a transmission, the SPI will stop sending and receiving immediately and both data received and data sent must be considered as lost.

Data Modes

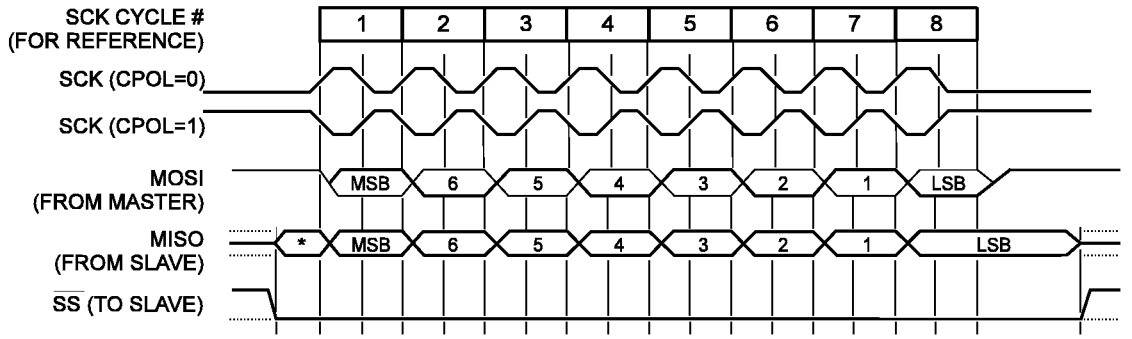
There are four combinations of SCK phase and polarity with respect to serial data, which are determined by control bits CPHA and CPOL. The SPI data transfer formats are shown in Figure 36 and Figure 37.

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



* Not defined but normally MSB of character just received

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



* Not defined but normally LSB of previously transmitted character

SPI Control Register - SPCR

Bit	7	6	5	4	3	2	1	0	
$0D ($2D)	SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR0	SPCR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

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

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

- **Bit 6 - SPE: SPI Enable**

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

- **Bit 5 - DORD: Data Order**

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

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

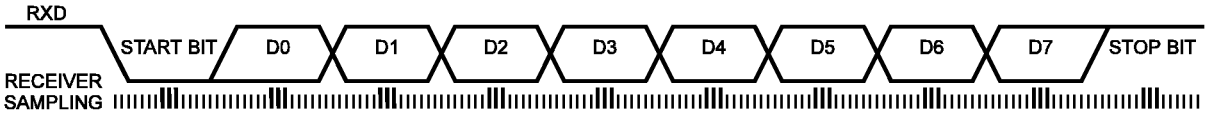
- **Bit 4 - MSTR: Master/Slave Select**

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

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 40.

Figure 40. 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 if the baud rate is high or CPU load is high.

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

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

UART Control

UART I/O Data Register - UDR

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

The UDR register is actually two physically separate registers sharing the same I/O address. When writing to the register, the UART Transmit Data register is written. When reading from UDR, the UART Receive Data register is read.

Port B Schematics

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

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

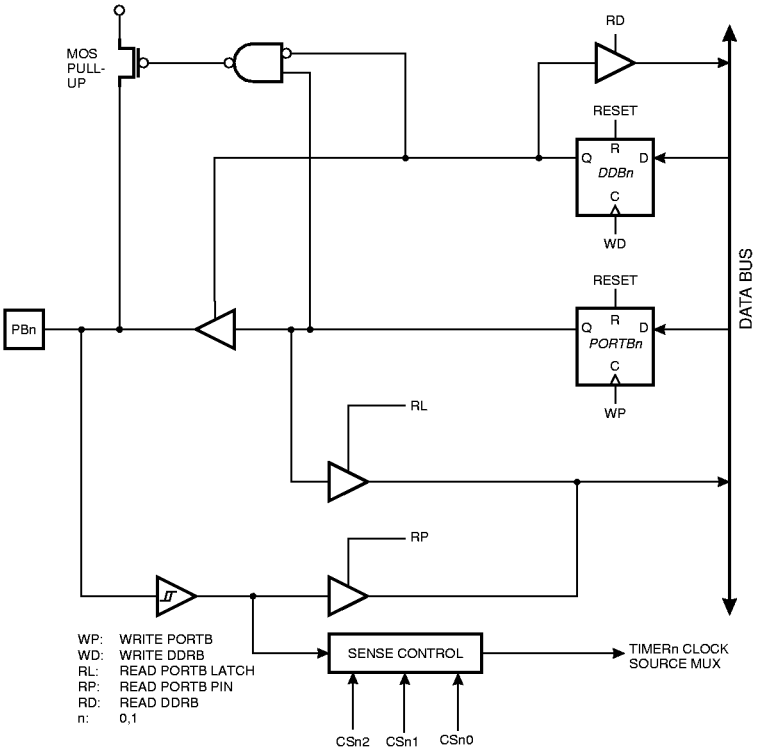
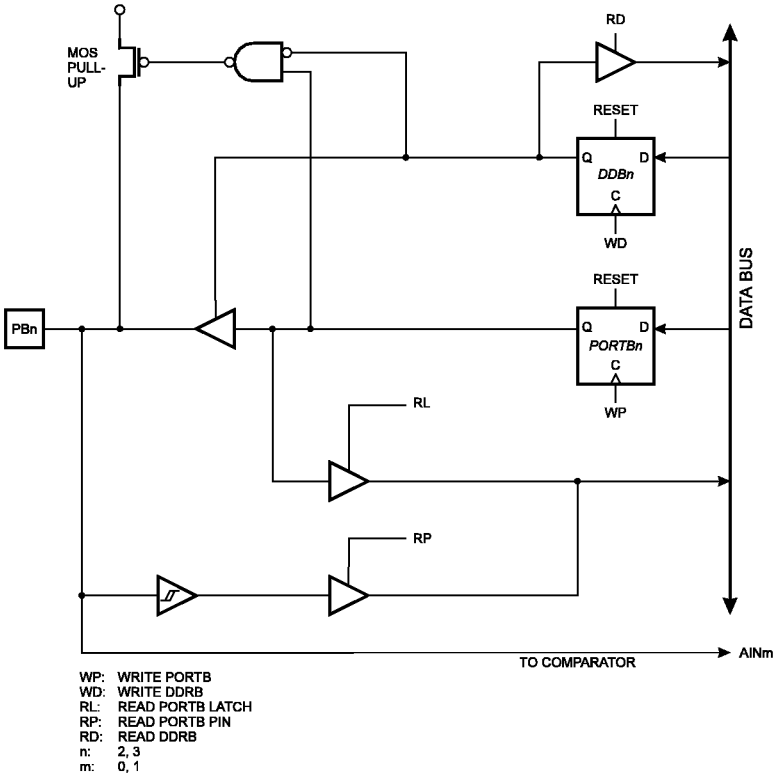


Figure 47. Port B Schematic Diagram (Pins PB2 and PB3)



Port C

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

All port pins have individually selectable pull-up resistors. The Port C output buffers can sink 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 if the internal pull-up resistors are activated.

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

Port C Data Register - PORTC

Bit	7	6	5	4	3	2	1	0	
$15 ($35)	PORTC7	PORTC6	PORTC5	PORTC4	PORTC3	PORTC2	PORTC1	PORTC0	PORTC
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

Port C Data Direction Register - DDRC

Bit	7	6	5	4	3	2	1	0	
$14 ($34)	DDC7	DDC6	DDC5	DDC4	DDC3	DDC2	DDC1	DDC0	DDRC
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

Port C Input Pins Address - PINC

Bit	7	6	5	4	3	2	1	0	
$13 ($33)	PINC7	PINC6	PINC5	PINC4	PINC3	PINC2	PINC1	PINC0	PINC
Read/Write	R	R	R	R	R	R	R	R	
Initial value	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 Port C 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 pins in Port C have equal functionality when used as digital I/O pins.

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

Table 23. DDCn Effects on Port C Pins

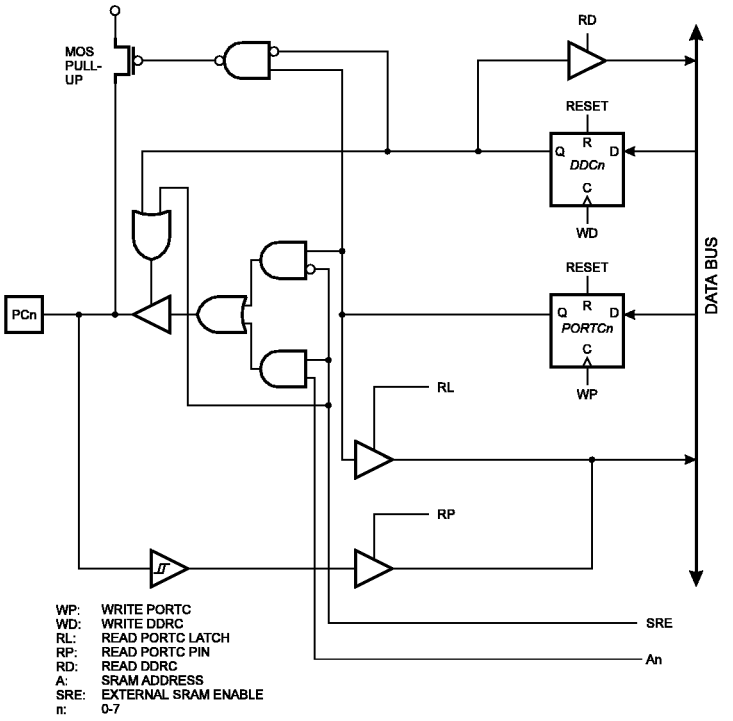
DDCn	PORTCn	I/O	Pull up	Comment
0	0	Input	No	Tri-state (Hi-Z)
0	1	Input	Yes	PCn will source current if ext. pulled low.
1	0	Output	No	Push-Pull Zero Output
1	1	Output	No	Push-Pull One Output

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



Port D

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

Three I/O memory address locations are allocated for the Port D, one each for the Data Register - PORTD, $12($32), Data Direction Register - DDRD, $11($31) and the Port D Input Pins - PIND, $10($30). The Port D Input Pins address is read only, while the Data Register and the Data Direction Register are read/write.

The Port D output buffers can sink 20 mA. As inputs, Port D pins that are externally pulled low will source current if the pull-up resistors are activated.

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

Table 24. Port D Pins Alternate Functions

Port Pin	Alternate Function
PD0	RXD (UART Input line)
PD1	TXD (UART Output line)
PD2	INT0 (External interrupt 0 input)
PD3	INT1 (External interrupt 1 input)
PD5	OC1A (Timer/Counter1 Output compareA match output)
PD6	$\overline{WR}$ (Write strobe to external memory)
PD7	$\overline{RD}$ (Read strobe to external memory)

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

The Program and Data memory arrays on the AT90S4414/8515 are programmed byte-by-byte in either programming modes. For the EEPROM, an auto-erase cycle is provided within the self-timed write operation in the serial programming mode. During programming, the supply voltage must be in accordance with Table 27.

Table 27. Supply Voltage During Programming

Part	Serial Programming	Parallel Programming
AT90S4414	2.7 - 6.0V	4.5 - 5.5V
AT90S8515	2.7 - 6.0V	4.5 - 5.5V

Parallel Programming

This section describes how to parallel program and verify Flash Program memory, EEPROM Data memory, Lock bits and Fuse bits in the AT90S4414/8515.

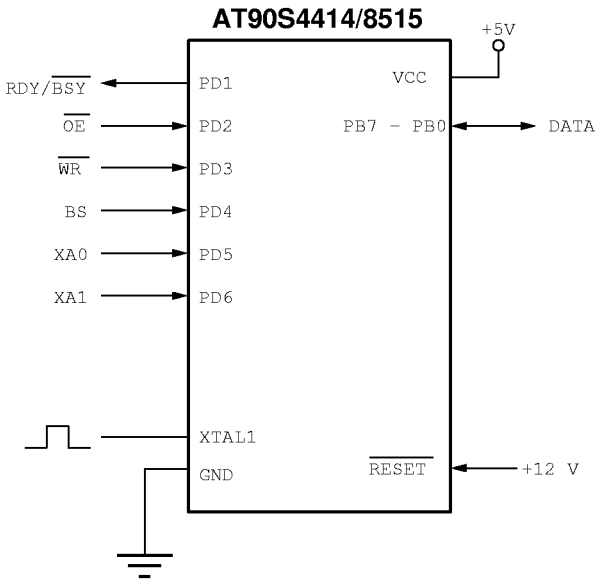
Signal Names

In this section, some pins of the AT90S4414/AT908515 are referenced by signal names describing their function during parallel programming. See Figure 60 and Table 28. Pins not described in Table 28 are referenced by pin names.

The XA1/XA0 pins determine the action executed when the XTAL1 pin is given a positive pulse. The bit coding are shown in Table 29.

When pulsing $\overline{WR}$ or $\overline{OE}$, the command loaded determines the action executed. The Command is a byte where the different bits are assigned functions as shown in Table 30.

Figure 60. Parallel Programming



Chip Erase

The Chip Erase command will erase the Flash and EEPROM memories, and the Lock bits. The Lock bits are not reset until the Flash and EEPROM have been completely erased. The Fuse bits are not changed. Chip Erase must be performed before the Flash or EEPROM is reprogrammed.

Load Command "Chip Erase"

1. Set XA1, XA0 to "10". This enables command loading.
2. Set BS to "0".
3. Set DATA to '1000 0000'. This is the command for Chip erase.
4. Give XTAL1 a positive pulse. This loads the command.
5. Give $\overline{WR}$ a t_{WLWH_CE} wide negative pulse to execute Chip Erase. See Table 31 for t_{WLWH_CE} value. Chip Erase does not generate any activity on the RDY/$\overline{BSY}$ pin.

Programming the Flash

A: Load Command "Write Flash"

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

B: Load Address High Byte

1. Set XA1, XA0 to "00". This enables address loading.
2. Set BS to "1". This selects high byte.
3. Set DATA = Address high byte ($00 - $07/$0F)
4. Give XTAL1 a positive pulse. This loads the address high byte.

C: Load Address Low Byte

1. Set XA1, XA0 to "00". This enables address loading.
2. Set BS to "0". This selects low byte.
3. Set DATA = Address low byte ($00 - $FF)
4. Give XTAL1 a positive pulse. This loads the address low byte.

D: Load Data Low Byte

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

E: Write Data Low Byte

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

(See Figure 61 for signal waveforms.)

F: Load Data High Byte

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

G: Write Data High Byte

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

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 72. Idle Supply current vs. V_{CC}

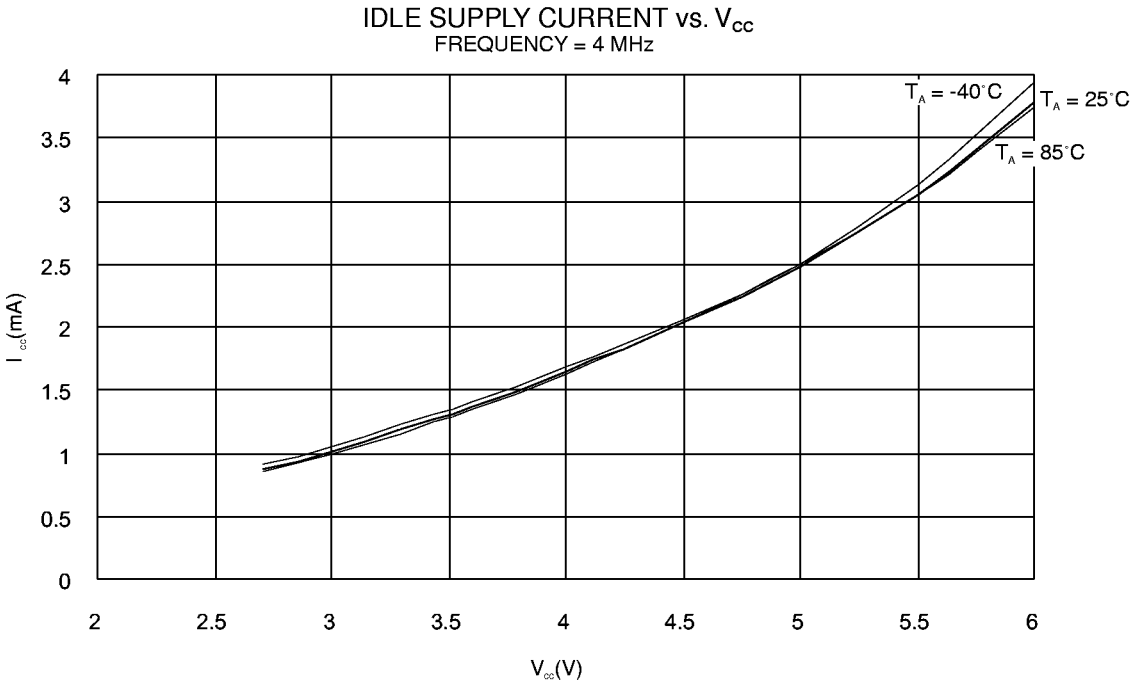
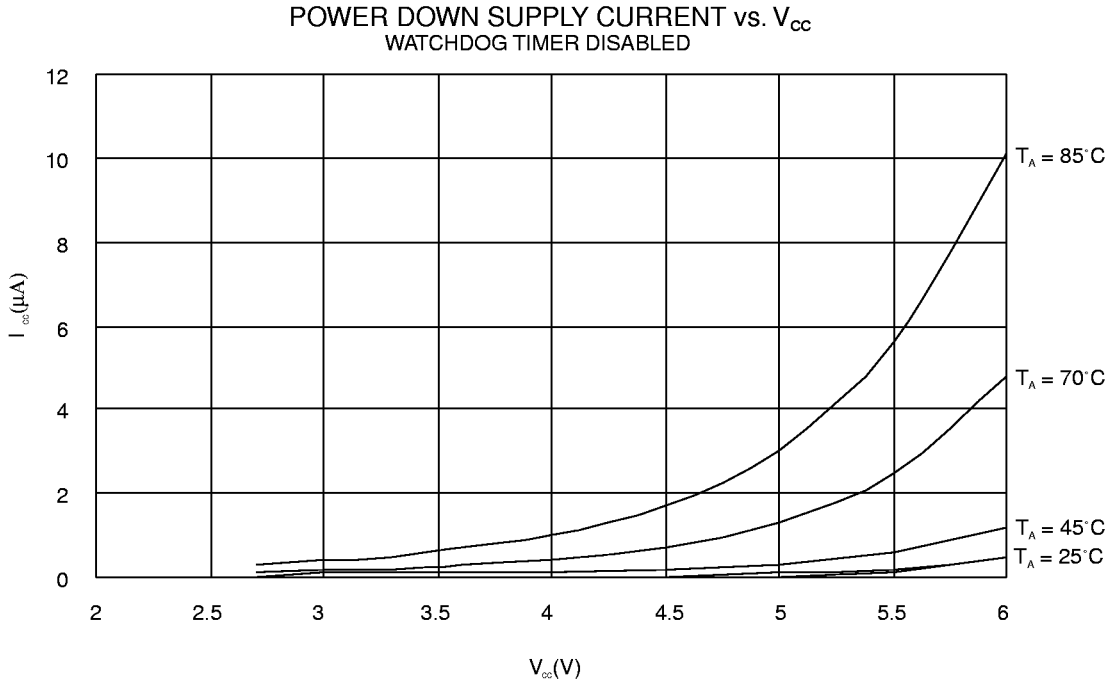


Figure 73. Power Down Supply Current vs. V_{CC}



Instruction Set Summary (Continued)

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