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	Active
Core Processor	AVR
Core Size	8-Bit
Speed	10MHz
Connectivity	I²C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	32
Program Memory Size	64KB (32K x 16)
Program Memory Type	FLASH
EEPROM Size	2K x 8
RAM Size	4K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 5.5V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
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/atmega644pv-10aq

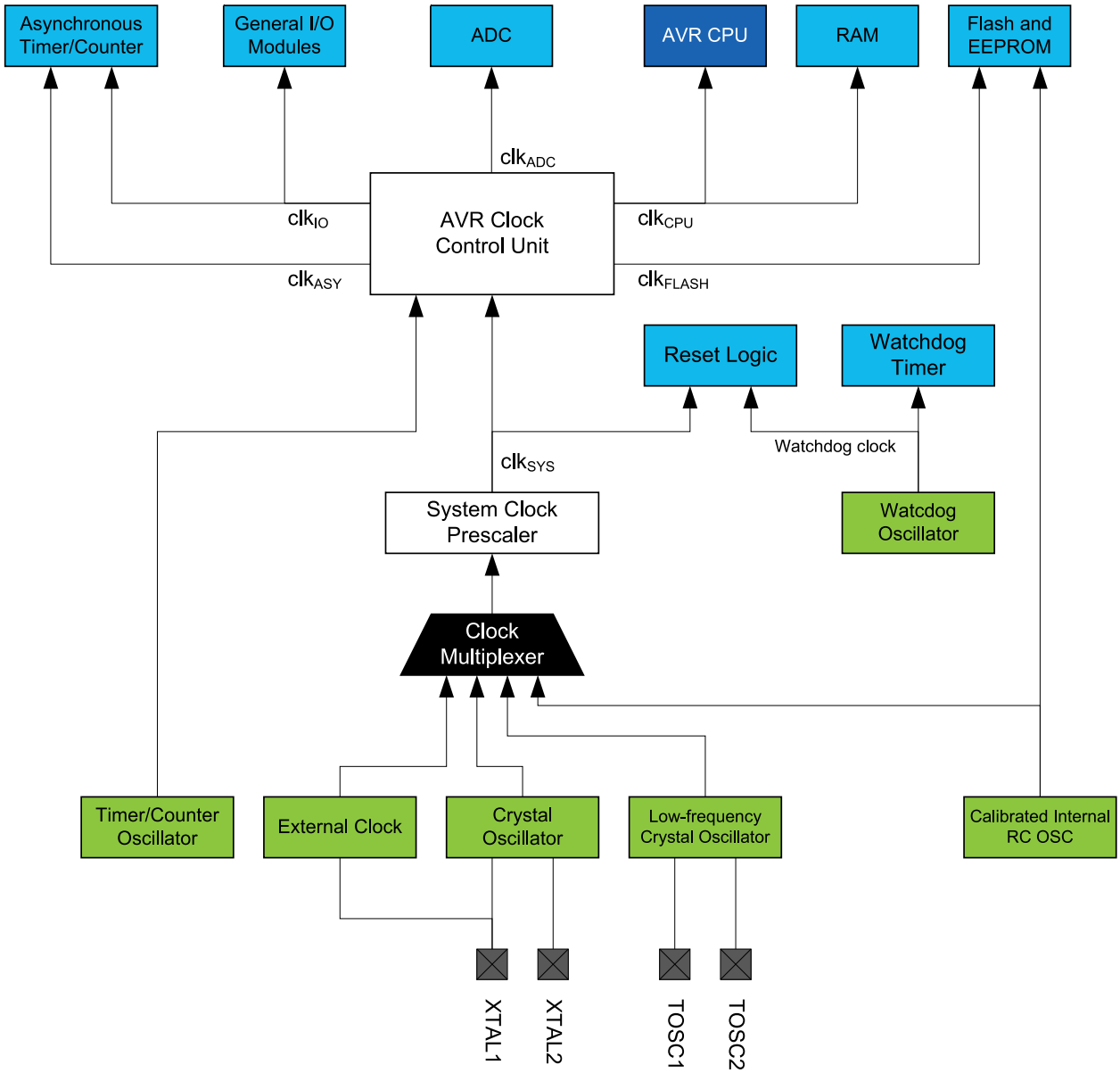
10. System Clock and Clock Options

10.1. Clock Systems and Their Distribution

The following figure illustrates the principal clock systems in the device and their distribution. All the clocks need not be active at a given time. In order to reduce power consumption, the clocks to modules not being used can be halted by using different sleep modes. The clock systems are described in the following sections.

The system clock frequency refers to the frequency generated from the System Clock Prescaler. All clock outputs from the AVR Clock Control Unit runs in the same frequency.

Figure 10-1. Clock Distribution



11.12.2. MCU Control Register

When addressing I/O Registers as data space using LD and ST instructions, the provided offset must be used. When using the I/O specific commands IN and OUT, the offset is reduced by 0x20, resulting in an I/O address offset within 0x00 - 0x3F.

The device is a complex microcontroller with more peripheral units than can be supported within the 64 locations reserved in Opcode for the IN and OUT instructions. For the Extended I/O space from 0x60 in SRAM, only the ST/STS/STD and LD/LDS/LDD instructions can be used.

Name: MCUCR
Offset: 0x55
Reset: 0x00
Property: When addressing as I/O Register: address offset is 0x35

Bit	7	6	5	4	3	2	1	0
	JTD	BODS	BODSE	PUD			IVSEL	IVCE
Access	R/W	R/W	R/W	R/W			R/W	R/W
Reset	0	0	0	0			0	0

Bit 7 – JTD
When this bit is zero, the JTAG interface is enabled if the JTAGEN Fuse is programmed. If this bit is one, the JTAG interface is disabled. In order to avoid unintentional disabling or enabling of the JTAG interface, a timed sequence must be followed when changing this bit: The application software must write this bit to the desired value twice within four cycles to change its value. Note that this bit must not be altered when using the On-chip Debug system.

Bit 6 – BODS: BOD Sleep
The BODS bit must be written to '1' in order to turn off BOD during sleep. Writing to the BODS bit is controlled by a timed sequence and the enable bit BODSE. To disable BOD in relevant sleep modes, both BODS and BODSE must first be written to '1'. Then, BODS must be written to '1' and BODSE must be written to zero within four clock cycles.

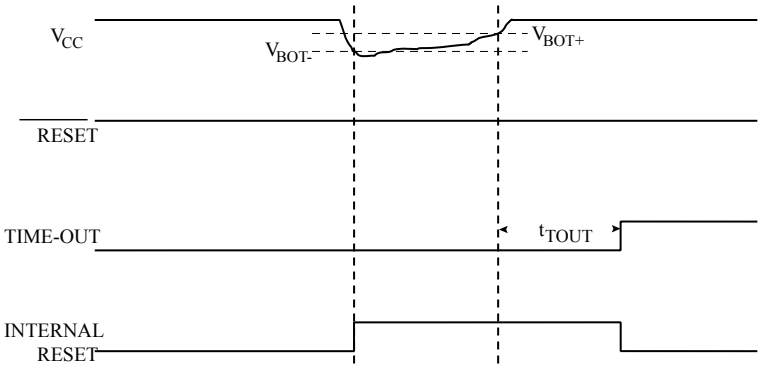
The BODS bit is active three clock cycles after it is set. A sleep instruction must be executed while BODS is active in order to turn off the BOD for the actual sleep mode. The BODS bit is automatically cleared after three clock cycles.

Bit 5 – BODSE: BOD Sleep Enable
BODSE enables setting of BODS control bit, as explained in BODS bit description. BOD disable is controlled by a timed sequence.

Bit 4 – PUD: Pull-up Disable
When this bit is written to one, the pull-ups in the I/O ports are disabled even if the DDxn and PORTxn Registers are configured to enable the pull-ups ({DDxn, PORTxn} = 0b01).

Bit 1 – IVSEL: Interrupt Vector Select
When the IVSEL bit is cleared (zero), the Interrupt Vectors are placed at the start of the Flash memory. When this bit is set (one), the Interrupt Vectors are moved to the beginning of the Boot Loader section of the Flash. The actual address of the start of the Boot Flash Section is determined by the BOOTSZ Fuses. To avoid unintentional changes of Interrupt Vector tables, a special write procedure must be followed to change the IVSEL bit:

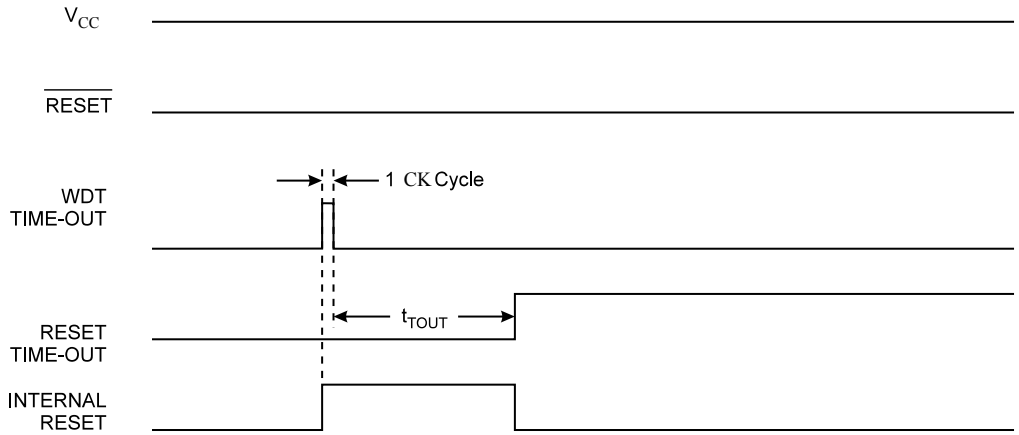
Figure 12-5. Brown-out Reset During Operation



12.6. Watchdog System Reset

When the Watchdog times out, it will generate a short reset pulse of one CK cycle duration. On the falling edge of this pulse, the delay timer starts counting the Time-out period t_{TOUT}.

Figure 12-6. Watchdog System Reset During Operation



12.7. Internal Voltage Reference

The device features an internal bandgap reference. This reference is used for Brown-out Detection, and it can be used as an input to the Analog Comparator or the ADC.

12.7.1. Voltage Reference Enable Signals and Start-up Time

The voltage reference has a start-up time that may influence the way it should be used. To save power, the reference is not always turned on. The reference is on during the following situations:

1. When the BOD is enabled (by programming the BODLEVEL [2:0] Fuses).
2. When the bandgap reference is connected to the Analog Comparator (by setting the ACBG bit in ACSR (ACSR.ACBG)).
3. When the ADC is enabled.

Thus, when the BOD is not enabled, after setting ACSR.ACBG or enabling the ADC, the user must always allow the reference to start up before the output from the Analog Comparator or ADC is used. To reduce power consumption in Power-Down mode, the user can avoid the three conditions above to ensure that the reference is turned off before entering Power-Down mode.

Port Pin	Alternate Functions
PB1	T1 (Timer/Counter 1 External Counter Input) CLKO (Divided System Clock Output) PCINT9 (Pin Change Interrupt 9)
PB0	T0 (Timer/Counter 0 External Counter Input) XCK0 (USART0 External Clock Input/Output) PCINT8 (Pin Change Interrupt 8)

The alternate pin configuration is as follows:

- SCK/PCINT15 – Port B, Bit 7
  - SCK: Master Clock output, Slave Clock input pin for SPI0 channel. When the SPI0 is enabled as a slave, this pin is configured as an input regardless of the setting of DDB7. When the SPI0 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.
  - PCINT15: Pin Change Interrupt source 15. The PB7 pin can serve as an external interrupt source.
- MISO/PCINT14 – Port B, Bit 6
  - MISO: Master Data input, Slave Data output pin for SPI channel. When the SPI0 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.
  - PCINT14: Pin Change Interrupt source 14. The PB6 pin can serve as an external interrupt source.
- MOSI/PCINT13 – Port B, Bit 5
  - MOSI: SPI Master Data output, Slave Data input for SPI channel. When the SPI0 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.
  - PCINT13: Pin Change Interrupt source 13. The PB5 pin can serve as an external interrupt source.
- $\overline{SS}$/OC0B/PCINT12 – Port B, Bit 4
  - $\overline{SS}$: Slave Port Select input. When the SPI is enabled as a slave, this pin is configured as an input regardless of the setting of DDB4. As a slave, the SPI0 is activated when this pin is driven low. When the SPI is enabled as a master, the data direction of this pin is controlled by DDB4. When the pin is forced to be an input, the pull-up can still be controlled by the PORTB4 bit.
  - OC0B: Output Compare Match B output. The PB4 pin can serve as an external output for the Timer/Counter0 Output Compare. The pin has to be configured as an output (DDB4 set “1”) to serve this function. The OC0B pin is also the output pin for the PWM mode timer function.
  - PCINT12: Pin Change Interrupt source 12. The PB4 pin can serve as an external interrupt source.
- AIN1/OC0A/PCINT11– Port B, Bit 3
  - AIN1: Analog Comparator Negative input. This pin is directly connected to the negative input of the Analog Comparator.

15.4.10. Port C Input Pins Address

When addressing I/O Registers as data space using LD and ST instructions, the provided offset must be used. When using the I/O specific commands IN and OUT, the offset is reduced by 0x20, resulting in an I/O address offset within 0x00 - 0x3F.

The device is a complex microcontroller with more peripheral units than can be supported within the 64 locations reserved in Opcode for the IN and OUT instructions. For the Extended I/O space from 0x60 in SRAM, only the ST/STS/STD and LD/LDS/LDD instructions can be used.

Name: PINC
Offset: 0x26
Reset: N/A
Property: When addressing as I/O Register: address offset is 0x06

Bit	7	6	5	4	3	2	1	0
	PINC7	PINC6	PINC5	PINC4	PINC3	PINC2	PINC1	PINC0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	x	x	x	x	x	x	x

Bits 7:0 – PINCn: Port C Input Pins Address [n = 7:0]
Writing to the pin register provides toggle functionality for IO. Refer to Toggling the Pin.

15.4.11. Port D Data Register

When addressing I/O Registers as data space using LD and ST instructions, the provided offset must be used. When using the I/O specific commands IN and OUT, the offset is reduced by 0x20, resulting in an I/O address offset within 0x00 - 0x3F.

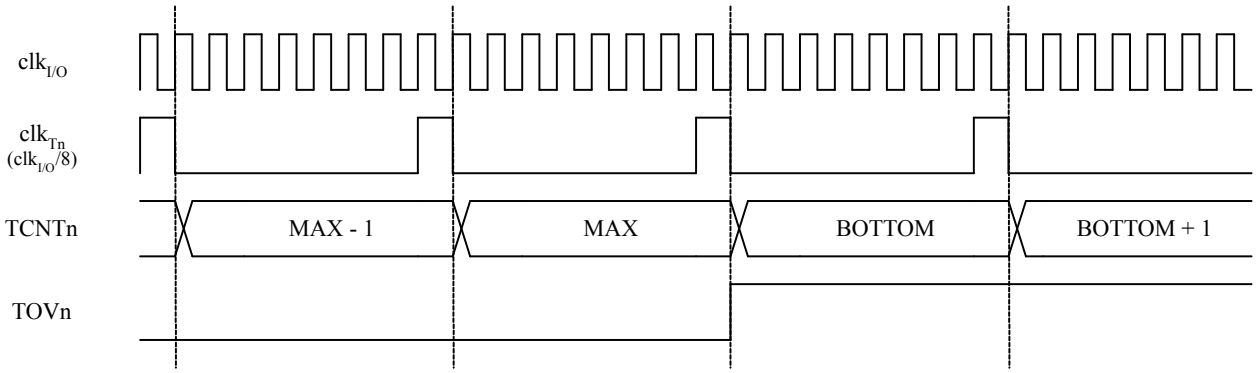
The device is a complex microcontroller with more peripheral units than can be supported within the 64 locations reserved in Opcode for the IN and OUT instructions. For the Extended I/O space from 0x60 in SRAM, only the ST/STS/STD and LD/LDS/LDD instructions can be used.

Name: PORTD
Offset: 0x2B
Reset: 0x00
Property: When addressing as I/O Register: address offset is 0x0B

Bit	7	6	5	4	3	2	1	0
	PORTD7	PORTD6	PORTD5	PORTD4	PORTD3	PORTD2	PORTD1	PORTD0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – PORTDn: Port D Data [n = 7:0]

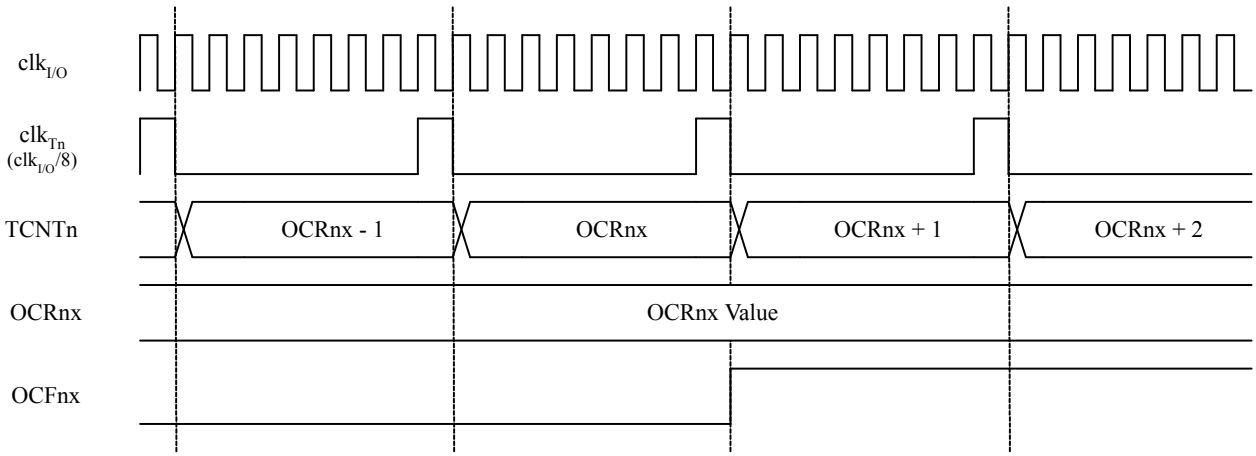
Figure 16-9. Timer/Counter Timing Diagram, with Prescaler ($f_{clk\_I/O}/8$)



Note: The “n” in the register and bit names indicates the device number (n = 0 for Timer/Counter 0), and the “x” indicates Output Compare unit (A/B).

The next figure shows the setting of OCF0B in all modes and OCF0A in all modes (except CTC mode and PWM mode where OCR0A is TOP).

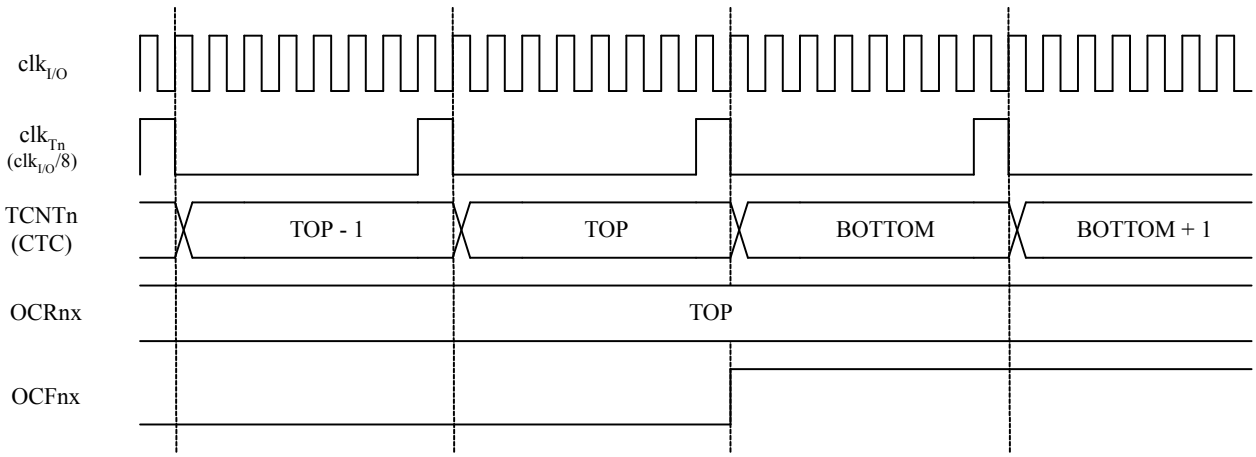
Figure 16-10. Timer/Counter Timing Diagram, Setting of OCF0x, with Prescaler ($f_{clk\_I/O}/8$)



Note: The “n” in the register and bit names indicates the device number (n = 0 for Timer/Counter 0), and the “x” indicates Output Compare unit (A/B).

The next figure shows the setting of OCF0A and the clearing of TCNT0 in CTC mode and fast PWM mode where OCR0A is TOP.

Figure 16-11. Timer/Counter Timing Diagram, Clear Timer on Compare Match mode, with Prescaler ($f_{clk\_I/O}/8$)



19.11.8. Asynchronous Status Register

Name: ASSR
Offset: 0xB6
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
		EXCLK	AS2	TCN2UB	OCR2AUB	OCR2BUB	TCR2AUB	TCR2BUB
Access		R	R	R	R	R	R	R
Reset		0	0	0	0	0	0	0

Bit 6 – EXCLK: Enable External Clock Input
When EXCLK is written to one, and asynchronous clock is selected, the external clock input buffer is enabled and an external clock can be input on Timer Oscillator 1 (TOSC1) pin instead of a 32kHz crystal. Writing to EXCLK should be done before asynchronous operation is selected. Note that the crystal Oscillator will only run when this bit is zero.

Bit 5 – AS2: Asynchronous Timer/Counter2
When AS2 is written to zero, Timer/Counter2 is clocked from the I/O clock, clkI/O. When AS2 is written to one, Timer/Counter2 is clocked from a crystal Oscillator connected to the Timer Oscillator 1 (TOSC1) pin. When the value of AS2 is changed, the contents of TCNT2, OCR2A, OCR2B, TCCR2A and TCCR2B might be corrupted.

Bit 4 – TCN2UB: Timer/Counter2 Update Busy
When Timer/Counter2 operates asynchronously and TCNT2 is written, this bit becomes set. When TCNT2 has been updated from the temporary storage register, this bit is cleared by hardware. A logical zero in this bit indicates that TCNT2 is ready to be updated with a new value.

Bit 3 – OCR2AUB: Enable External Clock Input
When Timer/Counter2 operates asynchronously and OCR2A is written, this bit becomes set. When OCR2A has been updated from the temporary storage register, this bit is cleared by hardware. A logical zero in this bit indicates that OCR2A is ready to be updated with a new value.

Bit 2 – OCR2BUB: Output Compare Register2 Update Busy
When Timer/Counter2 operates asynchronously and OCR2B is written, this bit becomes set. When OCR2B has been updated from the temporary storage register, this bit is cleared by hardware. A logical zero in this bit indicates that OCR2B is ready to be updated with a new value.

Bit 1 – TCR2AUB: Timer/Counter Control Register2 Update Busy
When Timer/Counter2 operates asynchronously and TCCR2A is written, this bit becomes set. When TCCR2A has been updated from the temporary storage register, this bit is cleared by hardware. A logical zero in this bit indicates that TCCR2A is ready to be updated with a new value.

Bit 0 – TCR2BUB: Timer/Counter Control Register2 Update Busy
When Timer/Counter2 operates asynchronously and TCCR2B is written, this bit becomes set. When TCCR2B has been updated from the temporary storage register, this bit is cleared by hardware. A logical zero in this bit indicates that TCCR2B is ready to be updated with a new value.

If a write is performed to any of the five Timer/Counter2 Registers while its update busy flag is set, the updated value might get corrupted and cause an unintentional interrupt to occur.

20.5.2. SPI Status Register 0

When addressing I/O Registers as data space using LD and ST instructions, the provided offset must be used. When using the I/O specific commands IN and OUT, the offset is reduced by 0x20, resulting in an I/O address offset within 0x00 - 0x3F.

The device is a complex microcontroller with more peripheral units than can be supported within the 64 locations reserved in Opcode for the IN and OUT instructions. For the Extended I/O space from 0x60 in SRAM, only the ST/STS/STD and LD/LDS/LDD instructions can be used.

Name: SPSR0
Offset: 0x4D
Reset: 0x00
Property: When addressing as I/O Register: address offset is 0x2D

Bit	7	6	5	4	3	2	1	0
	SPIF0	WCOL0						SPI2X0
Access	R	R						R/W
Reset	0	0						0

Bit 7 – SPIF0: SPI Interrupt Flag
When a serial transfer is complete, the SPIF Flag is set. An interrupt is generated if SPIE in SPCR is set 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, then accessing the SPI Data Register (SPDR).

Bit 6 – WCOL0: Write Collision Flag
The WCOL bit is set if the SPI Data Register (SPDR) is written during a data transfer. The WCOL bit (and the SPIF bit) are cleared by first reading the SPI Status Register with WCOL set, and then accessing the SPI Data Register.

Bit 0 – SPI2X0: Double SPI Speed Bit
When this bit is written logic one the SPI speed (SCK Frequency) will be doubled when the SPI is in Master mode (refer to Table 20-5). This means that the minimum SCK period will be two CPU clock periods. When the SPI is configured as Slave, the SPI is only guaranteed to work at fosc/4 or lower.

The SPI interface is also used for program memory and EEPROM downloading or uploading. See *Serial Downloading* for serial programming and verification.

21.12.1. USART I/O Data Register n

The USART Transmit Data Buffer Register and USART Receive Data Buffer Registers share the same I/O address referred to as USART Data Register or UDRn. The Transmit Data Buffer Register (TXB) will be the destination for data written to the UDR1 Register location. Reading the UDRn Register location will return the contents of the Receive Data Buffer Register (RXB).

For 5-, 6-, or 7-bit characters the upper unused bits will be ignored by the Transmitter and set to zero by the Receiver.

The transmit buffer can only be written when the UDRE Flag in the UCSRnA Register is set. Data written to UDRn when the UCSRnA.UDRE Flag is not set, will be ignored by the USART Transmitter n. When data is written to the transmit buffer, and the Transmitter is enabled, the Transmitter will load the data into the Transmit Shift Register when the Shift Register is empty. Then the data will be serially transmitted on the TxDn pin.

The receive buffer consists of a two level FIFO. The FIFO will change its state whenever the receive buffer is accessed. Due to this behavior of the receive buffer, do not use Read-Modify-Write instructions (SBI and CBI) on this location. Be careful when using bit test instructions (SBIC and SBIS), since these also will change the state of the FIFO.

Name: UDRn
Offset: 0xC6 + n*0x08 [n=0..1]
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
	TXB / RXB[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – TXB / RXB[7:0]: USART Transmit / Receive Data Buffer

BAUD	Baud rate (in bits per second, bps)
f_{OSC}	System Oscillator clock frequency
UBRRn	Contents of the UBRRnH and UBRRnL Registers, (0-4095)

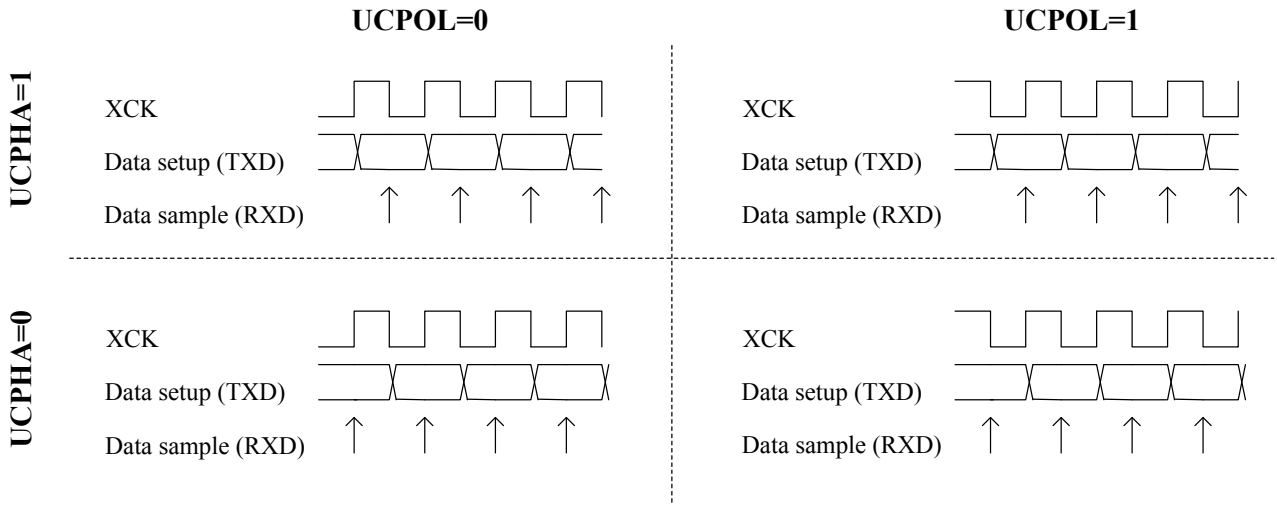
22.4. SPI Data Modes and Timing

There are four combinations of XCKn (SCK) phase and polarity with respect to serial data, which are determined by control bits UCPHAn and UCPOLn. The data transfer timing diagrams are shown in the following figure. Data bits are shifted out and latched in on opposite edges of the XCKn signal, ensuring sufficient time for data signals to stabilize. The UCPOLn and UCPHAn functionality is summarized in the following table. Note that changing the setting of any of these bits will corrupt all ongoing communication for both the Receiver and Transmitter.

Table 22-2. UCPOLn and UCPHAn Functionality

UCPOLn	UCPHAn	SPI Mode	Leading Edge	Trailing Edge
0	0	0	Sample (Rising)	Setup (Falling)
0	1	1	Setup (Rising)	Sample (Falling)
1	0	2	Sample (Falling)	Setup (Rising)
1	1	3	Setup (Falling)	Sample (Rising)

Figure 22-1. UCPHAn and UCPOLn data transfer timing diagrams.



22.5. Frame Formats

A serial frame for the MSPIM is defined to be one character of eight data bits. The USART in MSPIM mode has two valid frame formats:

- 8-bit data with MSB first
- 8-bit data with LSB first

A frame starts with the least or most significant data bit. Then the next data bits, up to a total of eight, are succeeding, ending with the most or least significant bit accordingly. When a complete frame is transmitted, a new frame can directly follow it, or the communication line can be set to an idle (high) state.

23. TWI - 2-wire Serial Interface

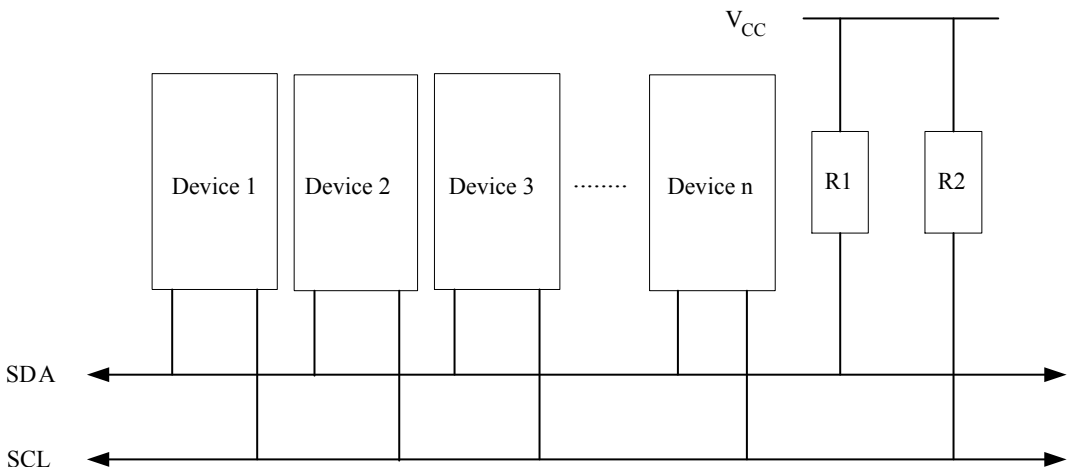
23.1. Features

- Simple, yet Powerful and Flexible Communication Interface, only two Bus Lines Needed
- Both Master and Slave Operation Supported
- Device can Operate as Transmitter or Receiver
- 7-bit Address Space Allows up to 128 Different Slave Addresses
- Multi-master Arbitration Support
- Up to 400kHz Data Transfer Speed
- Slew-rate Limited Output Drivers
- Noise Suppression Circuitry Rejects Spikes on Bus Lines
- Fully Programmable Slave Address with General Call Support
- Address Recognition Causes Wake-up When AVR is in Sleep Mode
- Compatible with Philips' I^2C protocol

23.2. Two-Wire Serial Interface Bus Definition

The Two-Wire Serial Interface (TWI) is ideally suited for typical microcontroller applications. The TWI protocol allows the systems designer to interconnect up to 128 different devices using only two bi-directional bus lines: one for clock (SCL) and one for data (SDA). The only external hardware needed to implement the bus is a single pull-up resistor for each of the TWI bus lines. All devices connected to the bus have individual addresses, and mechanisms for resolving bus contention are inherent in the TWI protocol.

Figure 23-1. TWI Bus Interconnection



23.2.1. TWI Terminology

The following definitions are frequently encountered in this section.

23.9.2. TWI Status Register

Name: TWSR
Offset: 0xB9
Reset: 0xF8
Property: -

Bit	7	6	5	4	3	2	1	0
	TWS7	TWS6	TWS5	TWS4	TWS3		TWPS[1:0]	
Access	R	R	R	R	R	R	R/W	R/W
Reset	1	1	1	1	1	0	0	0

Bits 1:0 – TWPS[1:0]: TWI Prescaler
These bits can be read and written, and control the bit rate prescaler.

Table 23-8. TWI Bit Rate Prescaler

TWS[1:0]	Prescaler Value
00	1
01	4
10	16
11	64

To calculate bit rates, refer to Bit Rate Generator Unit. The value of TWPS1...0 is used in the equation.

Bits 3, 4, 5, 6, 7 – TWS3, TWS4, TWS5, TWS6, TWS7: TWI Status Bit
The TWS[7:3] reflect the status of the TWI logic and the 2-wire Serial Bus. The different status codes are described later in this section. Note that the value read from TWSR contains both the 5-bit status value and the 2-bit prescaler value. The application designer should mask the prescaler bits to zero when checking the Status bits. This makes status checking independent of prescaler setting. This approach is used in this datasheet, unless otherwise noted.

Bit Number	Signal Name	Module
39	PD0.Data	Port D
38	PD0.Control	
37	PD1.Data	
36	PD1.Control	
35	PD2.Data	
34	PD2.Control	
33	PD3.Data	
32	PD3.Control	
31	PD4.Data	
30	PD4.Control	
29	PD5.Data	
28	PD5.Control	
27	PD6.Data	
26	PD6.Control	
25	PD7.Data	
24	PD7.Control	
23	PC0.Data	Port C
22	PC0.Control	
21	PC1.Data	
20	PC1.Control	
19	PC6.Data	
18	PC6.Control	
17	PC7.Data	
16	PC7.Control	

```
    lpm r0, Z+
    ld r1, Y+
    cpse r0, r1
    jmp Error
    sbiw loophi:looplo, 1 ;use subi for PAGESIZEB<=256
    brne Rdloop

    ; return to RWW section
    ; verify that RWW section is safe to read
Return:
    in temp1, SPMCSR
    sbrs temp1, RWWSB ; If RWWSB is set, the RWW section is not ready yet
    ret
    ; re-enable the RWW section
    ldi spmcrval, (1<<RWWSRE) | (1<<SPMEN)
    call Do_spm
    rjmp Return

Do_spm:
    ; check for previous SPM complete
Wait_spm:
    in temp1, SPMCSR
    sbrc temp1, SPMEN
    rjmp Wait_spm

    ; input: spmcrval determines SPM action
    ; disable interrupts if enabled, store status
    in temp2, SREG
    cli
    ; check that no EEPROM write access is present
Wait_ee:
    sbic EECR, EEPE
    rjmp Wait_ee
    ; SPM timed sequence
    out SPMCSR, spmcrval
    spm
    ; restore SREG (to enable interrupts if originally enabled)
    out SREG, temp2
```

```
    ret
```

27.8.14. ATmega644P Boot Loader Parameters

In the following tables, the parameters used in the description of the self programming are given.

Table 27-7. Boot Size Configuration, ATmega644P

BOOTSZ1	BOOTSZ0	Boot Size	Pages	Application Flash Section	Boot Loader Flash Section	End Application Section	Boot Reset Address (Start Boot Loader Section)
1	1	512 words	4	0x0000 - 0x7DFF	0x7E00 - 0x7FFF	0x7DFF	0x7E00
1	0	1024 words	8	0x0000 - 0x7BFF	0x7C00 - 0x7FFF	0x7BFF	0x7C00
0	1	2048 words	16	0x0000 - 0x77FF	0x7800 - 0x7FFF	0x77FF	0x7800
0	0	4096 words	32	0x0000 - 0x7FFF	0x7000 - 0x3FFF	0x6FFF	0x7000

Note: The different BOOTSZ Fuse configurations are shown in Figure 27-2

Table 27-8. Read-While-Write Limit, ATmega644P

Section	Pages	Address
Read-While-Write section (RWW)	224	0x0000 - 0x6FFF
No Read-While-Write section (NRWW)	32	0x7000 - 0x7FFF

Note: For details about these two section, see NRWW – No Read-While-Write Section and RWW – Read-While-Write Section.

Table 27-9. Explanation of Different Variables used in Figure and the Mapping to the Z-pointer, ATmega644P

Variable		Corresponding Variable (1)	Description
PCMSB	14		Most significant bit in the Program Counter. (The Program Counter is 15 bits PC[14:0])
PAGEMSB	6		Most significant bit which is used to address the words within one page (128 words in a page requires 7 bits PC [6:0]).
ZPCMSB		Z15	Bit in Z-register that is mapped to PCMSB. Because Z0 is not used, the ZPCMSB equals PCMSB + 1.
ZPAGEMSB		Z7	Bit in Z-register that is mapped to PAGEMSB. Because Z0 is not used, the ZPAGEMSB equals PAGEMSB + 1.

27.9.1. SPMCSR – Store Program Memory Control and Status Register

The Store Program Memory Control and Status Register contains the control bits needed to control the Boot Loader operations.

When addressing I/O Registers as data space using LD and ST instructions, the provided offset must be used. When using the I/O specific commands IN and OUT, the offset is reduced by 0x20, resulting in an I/O address offset within 0x00 - 0x3F.

The device is a complex microcontroller with more peripheral units than can be supported within the 64 locations reserved in Opcode for the IN and OUT instructions. For the Extended I/O space from 0x60 in SRAM, only the ST/STS/STD and LD/LDS/LDD instructions can be used.

Name: SPMCSR
Offset: 0x57
Reset: 0x00
Property: When addressing as I/O Register: address offset is 0x37

Bit	7	6	5	4	3	2	1	0
	SPMIE	RWWSB	SIGRD	RWWSRE	BLBSET	PGWRT	PGERS	SPMEN
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0		0	0	0	0

Bit 7 – SPMIE: SPM Interrupt Enable
When the SPMIE bit is written to one, and the I-bit in the Status Register is set (one), the SPM ready interrupt will be enabled. The SPM ready Interrupt will be executed as long as the SPMEN bit in the SPMCSR Register is cleared.

Bit 6 – RWWSB: Read-While-Write Section Busy
When a Self-Programming (Page Erase or Page Write) operation to the RWW section is initiated, the RWWSB will be set (one) by hardware. When the RWWSB bit is set, the RWW section cannot be accessed. The RWWSB bit will be cleared if the RWWSRE bit is written to one after a Self-Programming operation is completed. Alternatively the RWWSB bit will automatically be cleared if a page load operation is initiated.

Bit 5 – SIGRD: Signature Row Read
If this bit is written to one at the same time as SPMEN, the next LPM instruction within three clock cycles will read a byte from the signature row into the destination register. Please refer to *Reading the Fuse and Lock Bits from Software* in this chapter. An SPM instruction within four cycles after SIGRD and SPMEN are set will have no effect. This operation is reserved for future use and should not be used.

Bit 4 – RWWSRE: Read-While-Write Section Read Enable
When programming (Page Erase or Page Write) to the RWW section, the RWW section is blocked for reading (the RWWSB will be set by hardware). To re-enable the RWW section, the user software must wait until the programming is completed (SPMEN will be cleared). Then, if the RWWSRE bit is written to one at the same time as SPMEN, the next SPM instruction within four clock cycles re-enables the RWW section. The RWW section cannot be re-enabled while the Flash is busy with a Page Erase or a Page Write (SPMEN is set). If the RWWSRE bit is written while the Flash is being loaded, the Flash load operation will abort and the data loaded will be lost.

28. MEMPROG- Memory Programming

28.1. Program And Data Memory Lock Bits

The devices provides Lock bits. These can be left unprogrammed ('1') or can be programmed ('0') to obtain the additional features listed in Table. Lock Bit Protection Modes in this section. The Lock bits can only be erased to “1” with the Chip Erase command.

Table 28-1. Lock Bit Byte(1)

Lock Bit Byte	Bit No.	Description	Default Value
	7	–	1 (unprogrammed)
	6	–	1 (unprogrammed)
BLB12	5	Boot Lock bit	1 (unprogrammed)
BLB11	4	Boot Lock bit	1 (unprogrammed)
BLB02	3	Boot Lock bit	1 (unprogrammed)
BLB01	2	Boot Lock bit	1 (unprogrammed)
LB2	1	Lock bit	1 (unprogrammed)
LB1	0	Lock bit	1 (unprogrammed)

Note:

1. '1' means unprogrammed, '0' means programmed.

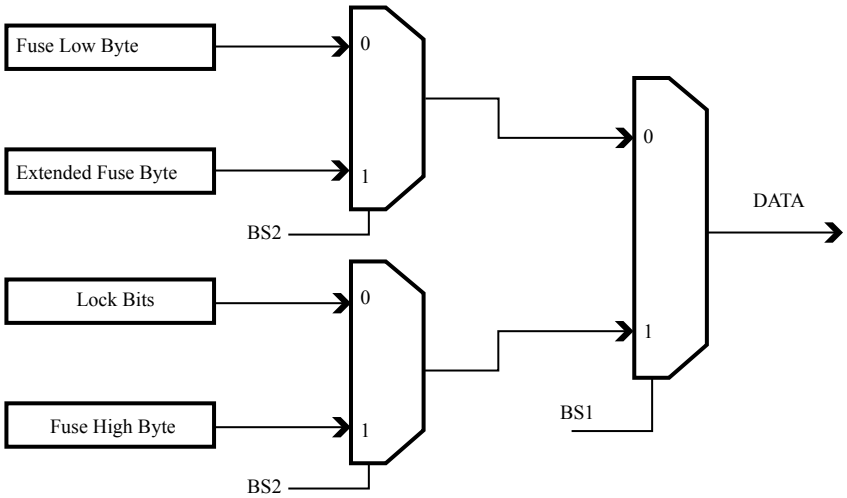
Table 28-2. Lock Bit Protection Modes(1)(2)

Memory Lock Bits			Protection Type
LB Mode	LB2	LB1	
1	1	1	No memory lock features enabled.
2	1	0	Further programming of the Flash and EEPROM is disabled in Parallel and Serial Programming mode. The Fuse bits are locked in both Serial and Parallel Programming mode.(1)
3	0	0	Further programming and verification of the Flash and EEPROM is disabled in Parallel and Serial Programming mode. The Boot Lock bits and Fuse bits are locked in both Serial and Parallel Programming mode.(1)

Note:

1. Program the Fuse bits and Boot Lock bits before programming the LB1 and LB2.
2. '1' means unprogrammed, '0' means programmed.

Figure 28-5. Mapping Between BS1, BS2 and the Fuse and Lock Bits During Read



28.8.13. Reading the Signature Bytes

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

1. Step A: Load Command “0000 1000”.
2. Step B: Load Address Low Byte (0x00 - 0x02).
3. Set $\overline{OE}$ to “0”, and BS1 to “0”. The selected Signature byte can now be read at DATA.
4. Set $\overline{OE}$ to “1”.

28.8.14. Reading the Calibration Byte

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

1. Step A: Load Command “0000 1000”.
2. Step B: Load Address Low Byte, 0x00.
3. Set $\overline{OE}$ to “0”, and BS1 to “1”. The Calibration byte can now be read at DATA.
4. Set $\overline{OE}$ to “1”.

28.8.15. Parallel Programming Characteristics

For characteristics of the Parallel Programming, please refer to *Parallel Programming Characteristics*.

28.9. Serial Downloading

Both the Flash and EEPROM 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.

BIT AND BIT-TEST INSTRUCTIONS					
Mnemonics	**Operands**	**Description**	**Operation**	**Flags**	**#Clocks**
LSL	Rd	Logical Shift Left	Rd(n+1) ← Rd(n), Rd(0) ← 0	Z,C,N,V	1
LSR	Rd	Logical Shift Right	Rd(n) ← Rd(n+1), Rd(7) ← 0	Z,C,N,V	1
ROL	Rd	Rotate Left Through Carry	Rd(0)←C,Rd(n+1)← Rd(n),C¬Rd(7)	Z,C,N,V	1
ROR	Rd	Rotate Right Through Carry	Rd(7)←C,Rd(n)← Rd(n+1),C←Rd(0)	Z,C,N,V	1
ASR	Rd	Arithmetic Shift Right	Rd(n) ← Rd(n+1), n=0...6	Z,C,N,V	1
SWAP	Rd	Swap Nibbles	Rd(3...0)←Rd(7...4),Rd(7...4)¬Rd(3...0)	None	1
BSET	s	Flag Set	SREG(s) ← 1	SREG(s)	1
BCLR	s	Flag Clear	SREG(s) ← 0	SREG(s)	1
BST	Rr, b	Bit Store from Register to T	T ← Rr(b)	T	1
BLD	Rd, b	Bit load from T to Register	Rd(b) ← T	None	1
SEC		Set Carry	C ← 1	C	1
CLC		Clear Carry	C ← 0	C	1
SEN		Set Negative Flag	N ← 1	N	1
CLN		Clear Negative Flag	N ← 0	N	1
SEZ		Set Zero Flag	Z ← 1	Z	1
CLZ		Clear Zero Flag	Z ← 0	Z	1
SEI		Global Interrupt Enable	I ← 1	I	1
CLI		Global Interrupt Disable	I ← 0	I	1
SES		Set Signed Test Flag	S ← 1	S	1
CLS		Clear Signed Test Flag	S ← 0	S	1
SEV		Set Two's Complement Overflow.	V ← 1	V	1
CLV		Clear Two's Complement Overflow	V ← 0	V	1
SET		Set T in SREG	T ← 1	T	1
CLT		Clear T in SREG	T ← 0	T	1
SEH		Set Half Carry Flag in SREG	H ← 1	H	1
CLH		Clear Half Carry Flag in SREG	H ← 0	H	1

DATA TRANSFER INSTRUCTIONS					
Mnemonics	**Operands**	**Description**	**Operation**	**Flags**	**#Clocks**
MOV	Rd, Rr	Move Between Registers	Rd ← Rr	None	1
MOVW	Rd, Rr	Copy Register Word	Rd+1:Rd ← Rr+1: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-Increment	Rd ← (X), X ← X + 1	None	2
LD	Rd, - X	Load Indirect and Pre-Decrement	X ← X - 1, Rd ← (X)	None	2
LD	Rd, Y	Load Indirect	Rd ← (Y)	None	2
LD	Rd, Y+	Load Indirect and Post-Increment	Rd ← (Y), Y ← Y + 1	None	2