Welcome to E-XFL.COM

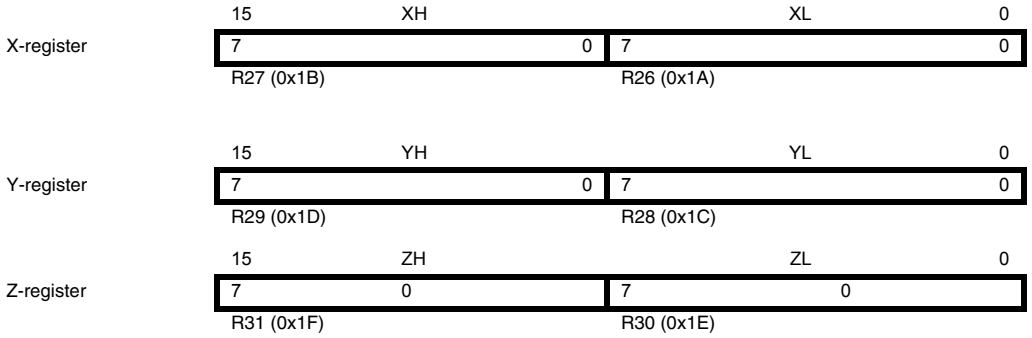
What is "Embedded - Microcontrollers"?

"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	16MHz
Connectivity	SPI, UART/USART, USI
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	53
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)	2.7V ~ 5.5V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	64-TQFP
Supplier Device Package	64-TQFP (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atmega645-16aur

Figure 7-3. The X-, Y-, and Z-registers



In the different addressing modes these address registers have functions as fixed displacement, automatic increment, and automatic decrement (see the instruction set reference for details).

7.6 Stack Pointer

The Stack is mainly used for storing temporary data, for storing local variables and for storing return addresses after interrupts and subroutine calls. The Stack Pointer Register always points to the top of the Stack. Note that the Stack is implemented as growing from higher memory locations to lower memory locations. This implies that a Stack PUSH command decreases the Stack Pointer.

The Stack Pointer points to the data SRAM Stack area where the Subroutine and Interrupt Stacks are located. This Stack space in the data SRAM must be defined by the program before any subroutine calls are executed or interrupts are enabled. The Stack Pointer must be set to point above 0x60. The Stack Pointer is decremented by one when data is pushed onto the Stack with the PUSH instruction, and it is decremented by two when the return address is pushed onto the Stack with subroutine call or interrupt. The Stack Pointer is incremented by one when data is popped from the Stack with the POP instruction, and it is incremented by two when data is popped from the Stack with return from subroutine RET or return from interrupt RETI.

The AVR Stack Pointer is implemented as two 8-bit registers in the I/O space. The number of bits actually used is implementation dependent. Note that the data space in some implementations of the AVR architecture is so small that only SPL is needed. In this case, the SPH Register will not be present.

Bit	15	14	13	12	11	10	9	8	
0x3E (0x5E)	**SP15**	**SP14**	**SP13**	**SP12**	**SP11**	**SP10**	**SP9**	**SP8**	**SPH**
0x3D (0x5D)	**SP7**	**SP6**	**SP5**	**SP4**	**SP3**	**SP2**	**SP1**	**SP0**	**SPL**
	7	6	5	4	3	2	1	0	
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

7.7 Instruction Execution Timing

This section describes the general access timing concepts for instruction execution. The AVR CPU is driven by the CPU clock clk_{CPU}, directly generated from the selected clock source for the chip. No internal clock division is used.

Figure 7-4 on page 15 shows the parallel instruction fetches and instruction executions enabled by the Harvard architecture and the fast-access Register File concept. This is the basic pipelin-

Table 9-4. Start-up Times for the Crystal Oscillator Clock Selection (Continued)

CKSEL0	SUT1:0	Start-up Time from Power-down and Power-save	Additional Delay from Reset (V_{CC} = 5.0V)	Recommended Usage
0	11	1K CK(2)	14CK + 4.1ms	Ceramic resonator, fast rising power
1	00	1K CK(2)	14CK + 65ms	Ceramic resonator, slowly rising power
1	01	16K CK	14CK	Crystal Oscillator, BOD enabled
1	10	16K CK	14CK + 4.1ms	Crystal Oscillator, fast rising power
1	11	16K CK	14CK + 65ms	Crystal Oscillator, slowly rising power

Note: 1. These options should only be used when not operating close to the maximum frequency of the device, and only if frequency stability at start-up is not important for the application. These options are not suitable for crystals.

2. These options are intended for use with ceramic resonators and will ensure frequency stability at start-up. They can also be used with crystals when not operating close to the maximum frequency of the device, and if frequency stability at start-up is not important for the application.

9.4 Low-frequency Crystal Oscillator

To use a 32.768 kHz watch crystal as the clock source for the device, the low-frequency crystal Oscillator must be selected by setting the CKSEL Fuses to “0110” or “0111”. The crystal should be connected as shown in Figure 9-2 on page 28. When this Oscillator is selected, start-up times are determined by the SUT Fuses as shown in Table 9-5 on page 29 and CKSEL1..0 as shown in Table 9-6 on page 29.

Table 9-5. Start-up Times for the Low-frequency Crystal Oscillator Clock Selection

SUT1..0	Additional Delay from Reset (V_{CC} = 5.0V)	Recommended Usage
00	14CK	Fast rising power or BOD enabled
01	14CK + 4.1ms	Slowly rising power
10	14CK + 65ms	Stable frequency at start-up
11	Reserved	

Table 9-6. Start-up Times for the Low-frequency Crystal Oscillator Clock Selection

CKSEL3..0	Start-up Time from Power-down and Power-save	Recommended Usage
0110(Note:)	1K CK	
0111	32K CK	Stable frequency at start-up

Note: This option should only be used if frequency stability at start-up is not important for the application

9.5 Calibrated Internal RC Oscillator

The calibrated Internal RC Oscillator by default provides a 8.0MHz clock. The frequency is nominal value at 3V and 25°C. The device is shipped with the CKDIV8 Fuse programmed. See “System Clock Prescaler” on page 32 for more details.

Table 14-4 and Table 14-5 relate the alternate functions of Port B to the overriding signals shown in Figure 14-5 on page 66. SPI MSTR INPUT and SPI SLAVE OUTPUT constitute the MISO signal, while MOSI is divided into SPI MSTR OUTPUT and SPI SLAVE INPUT.

Table 14-4. Overriding Signals for Alternate Functions in PB7:PB4

Signal Name	PB7/OC2A/ PCINT15	PB6/OC1B/ PCINT14	PB5/OC1A/ PCINT13	PB4/OC0A/ PCINT12
PUOE	0	0	0	0
PUOV	0	0	0	0
DDOE	0	0	0	0
DDOV	0	0	0	0
PVOE	OC2A ENABLE	OC1B ENABLE	OC1A ENABLE	OC0A ENABLE
PVOV	OC2A	OC1B	OC1A	OC0A
PTOE	–	–	–	–
DIEOE	PCINT15 • PCIE1	PCINT14 • PCIE1	PCINT13 • PCIE1	PCINT12 • PCIE1
DIEOV	1	1	1	1
DI	PCINT15 INPUT	PCINT14 INPUT	PCINT13 INPUT	PCINT12 INPUT
AIO	–	–	–	–

Table 14-5. Overriding Signals for Alternate Functions in PB3:PB0

Signal Name	PB3/MISO/ PCINT11	PB2/MOSI/ PCINT10	PB1/SCK/ PCINT9	PB0/$\overline{SS}$/ PCINT8
PUOE	SPE • MSTR	SPE • $\overline{MSTR}$	SPE • $\overline{MSTR}$	SPE • $\overline{MSTR}$
PUOV	PORTB3 • $\overline{PUD}$	PORTB2 • $\overline{PUD}$	PORTB1 • $\overline{PUD}$	PORTB0 • $\overline{PUD}$
DDOE	SPE • MSTR	SPE • $\overline{MSTR}$	SPE • $\overline{MSTR}$	SPE • $\overline{MSTR}$
DDOV	0	0	0	0
PVOE	SPE • $\overline{MSTR}$	SPE • MSTR	SPE • MSTR	0
PVOV	SPI SLAVE OUTPUT	SPI MSTR OUTPUT	SCK OUTPUT	0
PTOE	–	–	–	–
DIEOE	PCINT11 • PCIE1	PCINT10 • PCIE1	PCINT9 • PCIE1	PCINT8 • PCIE1
DIEOV	1	1	1	1
DI	PCINT11 INPUT SPI MSTR INPUT	PCINT10 INPUT SPI SLAVE INPUT	PCINT9 INPUT SCK INPUT	PCINT8 INPUT SPI $\overline{SS}$
AIO	–	–	–	–

- **XCK/AIN0/PCINT2 – Port E, Bit 2**

XCK, USART External Clock. The Data Direction Register (DDE2) controls whether the clock is output (DDE2 set) or input (DDE2 cleared). The XCK pin is active only when the USART operates in synchronous mode.

AIN0 – Analog Comparator Positive input. This pin is directly connected to the positive input of the Analog Comparator.

PCINT2, Pin Change Interrupt Source 2: The PE2 pin can serve as an external interrupt source.

- **TXD/PCINT1 – Port E, Bit 1**

TXD0, UART0 Transmit pin.

PCINT1, Pin Change Interrupt Source 1: The PE1 pin can serve as an external interrupt source.

- **RXD/PCINT0 – Port E, Bit 0**

RXD, USART Receive pin. Receive Data (Data input pin for the USART). When the USART Receiver is enabled this pin is configured as an input regardless of the value of DDE0. When the USART forces this pin to be an input, a logical one in PORTE0 will turn on the internal pull-up.

PCINT0, Pin Change Interrupt Source 0: The PE0 pin can serve as an external interrupt source.

Table 14-9 and Table 14-10 relates the alternate functions of Port E to the overriding signals shown in Figure 14-5 on page 66.

Table 14-9. Overriding Signals for Alternate Functions PE7:PE4

Signal Name	PE7/PCINT7	PE6/DO/ PCINT6	PE5/DI/SDA/ PCINT5	PE4/USCK/SCL/ PCINT4
PUOE	0	0	USI_TWO-WIRE	USI_TWO-WIRE
PUOV	0	0	0	0
DDOE	CKOUT(1)	0	USI_TWO-WIRE	USI_TWO-WIRE
DDOV	1	0	($\overline{\text{SDA}}$ + $\overline{\text{PORTE5}}$) • DDE5	(USI_SCL_HOLD + $\overline{\text{PORTE4}}$) • DDE4
PVOE	CKOUT(1)	USI_THREE-WIRE	USI_TWO-WIRE • DDE5	USI_TWO-WIRE • DDE4
PVOV	$clk_{I/O}$	DO	0	0
PTOE	–	–	0	USITC
DIEOE	PCINT7 • PCIE0	PCINT6 • PCIE0	(PCINT5 • PCIE0) + USISIE	(PCINT4 • PCIE0) + USISIE
DIEOV	1	1	1	1
DI	PCINT7 INPUT	PCINT6 INPUT	DI/SDA INPUT PCINT5 INPUT	USCKL/SCL INPUT PCINT4 INPUT
AIO	–	–	–	–

Note: 1. CKOUT is one if the CKOUT Fuse is programmed

Table 14-10. Overriding Signals for Alternate Functions in PE3:PE0

Signal Name	PE3/AIN1/ PCINT3	PE2/XCK/AIN0/ PCINT2	PE1/TXD/ PCINT1	PE0/RXD/PCINT 0
PUOE	0	XCK OUTPUT ENABLE	TXEN	RXEN
PUOV	0	XCK	0	PORTE0 • $\overline{PUD}$
DDOE	0	0	TXEN	RXEN
DDOV	0	0	1	0
PVOE	0	0	TXEN	0
PVOV	0	0	TXD	0
PTOE	–	–	–	–
DIEOE	(PCINT3 • PCIE0) + AIN1D[1]	(PCINT2 • PCIE0) + AIN0D[1]	PCINT1 • PCIE0	PCINT0 • PCIE0
DIEOV	PCINT3 • PCIE0	PCINT2 • PCIE0	1	1
DI	PCINT3 INPUT	XCK/PCINT2 INPUT	PCINT1 INPUT	RXD/PCINT0 INPUT
AIO	AIN1 INPUT	AIN0 INPUT	–	–

Note: 1. AIN0D and AIN1D is described in "DIDR1 – Digital Input Disable Register 1" on page 200.

14.3.4 Alternate Functions of Port F

The Port F has an alternate function as analog input for the ADC as shown in Table 14-11. If some Port F pins are configured as outputs, it is essential that these do not switch when a conversion is in progress. This might corrupt the result of the conversion. If the JTAG interface is enabled, the pull-up resistors on pins PF7(TDI), PF5(TMS) and PF4(TCK) will be activated even if a reset occurs.

Table 14-11. Port F Pins Alternate Functions

Port Pin	Alternate Function
PF7	ADC7/TDI (ADC input channel 7 or JTAG Test Data Input)
PF6	ADC6/TDO (ADC input channel 6 or JTAG Test Data Output)
PF5	ADC5/TMS (ADC input channel 5 or JTAG Test mode Select)
PF4	ADC4/TCK (ADC input channel 4 or JTAG Test ClocK)
PF3	ADC3 (ADC input channel 3)
PF2	ADC2 (ADC input channel 2)
PF1	ADC1 (ADC input channel 1)
PF0	ADC0 (ADC input channel 0)

- **TDI, ADC7 – Port F, Bit 7**

ADC7, Analog to Digital Converter, Channel 7.

TDI, JTAG Test Data In: Serial input data to be shifted in to the Instruction Register or Data Register (scan chains). When the JTAG interface is enabled, this pin can not be used as an I/O pin.

Table 14-18. Overriding Signals for Alternate Functions in PH3:0

Signal Name	PH3/PCINT19	PH2/PCINT18	PH1/PCINT17	PH0/PCINT16
PUOE	0	0	0	0
PUOV	0	0	0	0
DDOE	0	0	0	0
DDOV	0	0	0	0
PVOE	0	0	0	0
PVOV	0	0	0	0
PTOE	–	–	–	–
DIEOE	PCINT19 • PCIE0	PCINT18 • PCIE0	PCINT17 • PCIE0	PCINT16 • PCIE0
DIEOV	0	0	0	0
DI	PCINT19 INPUT	PCINT18 INPUT	PCINT17 INPUT	PCINT16 INPUT
AIO	–	–	–	–

14.3.7 Alternate Functions of Port J

Port J is only present in ATmega3250/6450. The alternate pin configuration is as follows:

Table 14-19. Port J Pins Alternate Functions

Port Pin	Alternate Function
PJ6	PCINT30 (Pin Change Interrupt30)
PJ5	PCINT29 (Pin Change Interrupt29)
PJ4	PCINT28 (Pin Change Interrupt28)
PJ3	PCINT27 (Pin Change Interrupt27)
PJ2	PCINT26(Pin Change Interrupt26)
PJ1	PCINT25(Pin Change Interrupt25)
PJ0	PCINT24 (Pin Change Interrupt26)

The alternate pin configuration is as follows:

- **PCINT30 – Port J, Bit 6**

PCINT30, Pin Change Interrupt Source 30: The PE30 pin can serve as an external interrupt source.

- **PCINT29 – Port J, Bit 5**

PCINT29, Pin Change Interrupt Source 29: The PE29 pin can serve as an external interrupt source.

- **PCINT28 – Port J, Bit 4**

PCINT28, Pin Change Interrupt Source 28: The PE28 pin can serve as an external interrupt source.

14.4.21 DDRG – Port G Data Direction Register

Bit	7	6	5	4	3	2	1	0	
0x13 (0x33)	–	–	–	DDG4	DDG3	DDG2	DDG1	DDG0	DDRG
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	

14.4.22 PING – Port G Input Pins Address

Bit	7	6	5	4	3	2	1	0	
0x12 (0x32)	–	–	PING5	PING4	PING3	PING2	PING1	PING0	PING
Read/Write	R	R	R	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	N/A	N/A	N/A	N/A	N/A	

14.4.23 PORTH – Port H Data Register(1)

Bit	7	6	5	4	3	2	1	0	
(0xDA)	PORTH7	PORTH6	PORTH5	PORTH4	PORTH3	PORTH2	PORTH1	PORTH0	PORTH
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	

14.4.24 DDRH – Port H Data Direction Register(1)

Bit	7	6	5	4	3	2	1	0	
(0xD9)	DDH7	DDH6	DDH5	DDH4	DDH3	DDH2	DDH1	DDH0	DDRH
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	

14.4.25 PINH – Port H Input Pins Address(1)

Bit	7	6	5	4	3	2	1	0	
(0xD8)	PINH7	PINH6	PINH5	PINH4	PINH3	PINH2	PINH1	PINH0	PINH
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	

14.4.26 PORTJ – Port J Data Register(1)

Bit	7	6	5	4	3	2	1	0	
(0xDD)	–	PORTJ6	PORTJ5	PORTJ4	PORTJ3	PORTJ2	PORTJ1	PORTJ0	PORTJ
Read/Write	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

14.4.27 DDRJ – Port J Data Direction Register(1)

Bit	7	6	5	4	3	2	1	0	
(0xDC)	–	DDJ6	DDJ5	DDJ4	DDJ3	DDJ2	DDJ1	DDJ0	DDRJ
Read/Write	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

14.4.28 PINJ – Port J Input Pins Address(1)

Bit	7	6	5	4	3	2	1	0	
(0xDB)	–	PINJ6	PINJ5	PINJ4	PINJ3	PINJ2	PINJ1	PINJ0	PINJ
Read/Write	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	N/A	N/A	N/A	N/A	N/A	N/A	N/A	

Note: 1. Register only available in ATmega3250/6450.

The definitions in Table 15-1 are also used extensively throughout the document.

Table 15-1. Definitions of Timer/Counter values.

BOTTOM	The counter reaches the BOTTOM when it becomes 0x00.
MAX	The counter reaches its MAXimum when it becomes 0xFF (decimal 255).
TOP	The counter reaches the TOP when it becomes equal to the highest value in the count sequence. The TOP value can be assigned to be the fixed value 0xFF (MAX) or the value stored in the OCR0A Register. The assignment is dependent on the mode of operation.

15.2.2 Registers

The Timer/Counter (TCNT0) and Output Compare Register (OCR0A) are 8-bit registers. Interrupt request (abbreviated to Int.Req. in the figure) signals are all visible in the Timer Interrupt Flag Register (TIFR0). All interrupts are individually masked with the Timer Interrupt Mask Register (TIMSK0). TIFR0 and TIMSK0 are not shown in the figure.

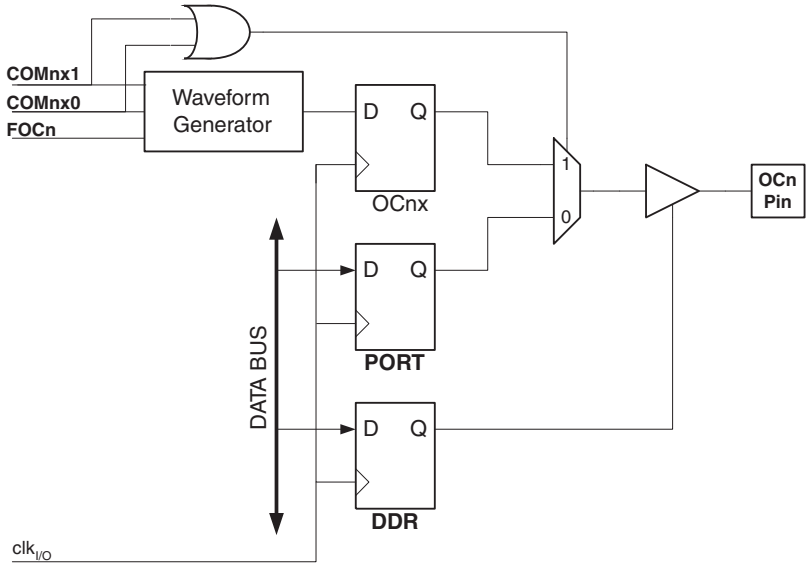
The Timer/Counter can be clocked internally, via the prescaler, or by an external clock source on the T0 pin. The Clock Select logic block controls which clock source and edge the Timer/Counter uses to increment (or decrement) its value. The Timer/Counter is inactive when no clock source is selected. The output from the Clock Select logic is referred to as the timer clock (clk_{T0}).

The double buffered Output Compare Register (OCR0A) is compared with the Timer/Counter value at all times. The result of the compare can be used by the Waveform Generator to generate a PWM or variable frequency output on the Output Compare pin (OC0A). See “Output Compare Unit” on page 87. for details. The compare match event will also set the Compare Flag (OCF0A) which can be used to generate an Output Compare interrupt request.

15.3 Timer/Counter Clock Sources

The Timer/Counter can be clocked by an internal or an external clock source. The clock source is selected by the Clock Select logic which is controlled by the Clock Select (CS02:0) bits located in the Timer/Counter Control Register (TCCR0A). For details on clock sources and prescaler, see “Timer/Counter0 and Timer/Counter1 Prescalers” on page 99.

Figure 15-4. Compare Match Output Unit, Schematic



The general I/O port function is overridden by the Output Compare (OC0A) from the Waveform Generator if either of the COM0A1:0 bits are set. However, the OC0A pin direction (input or output) is still controlled by the Data Direction Register (DDR) for the port pin. The Data Direction Register bit for the OC0A pin (DDR_OC0A) must be set as output before the OC0A value is visible on the pin. The port override function is independent of the Waveform Generation mode.

The design of the Output Compare pin logic allows initialization of the OC0A state before the output is enabled. Note that some COM0A1:0 bit settings are reserved for certain modes of operation. See “Register Description” on page 96.

15.6.1 Compare Output Mode and Waveform Generation

The Waveform Generator uses the COM0A1:0 bits differently in Normal, CTC, and PWM modes. For all modes, setting the COM0A1:0 = 0 tells the Waveform Generator that no action on the OC0A Register is to be performed on the next compare match. For compare output actions in the non-PWM modes refer to Table 15-3 on page 97. For fast PWM mode, refer to Table 15-4 on page 97, and for phase correct PWM refer to Table 15-5 on page 98.

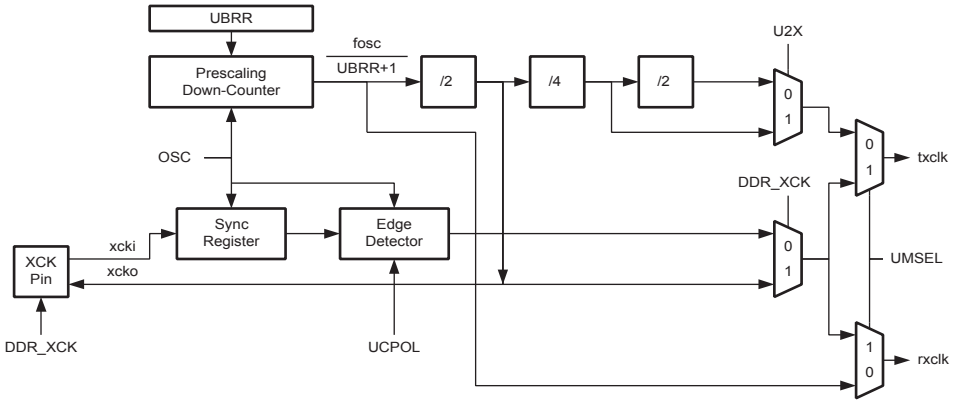
A change of the COM0A1:0 bits state will have effect at the first compare match after the bits are written. For non-PWM modes, the action can be forced to have immediate effect by using the FOC0A strobe bits.

15.7 Modes of Operation

The mode of operation, i.e., the behavior of the Timer/Counter and the Output Compare pins, is defined by the combination of the Waveform Generation mode (WGM01:0) and Compare Output mode (COM0A1:0) bits. The Compare Output mode bits do not affect the counting sequence, while the Waveform Generation mode bits do. The COM0A1:0 bits control whether the PWM output generated should be inverted or not (inverted or non-inverted PWM). For non-PWM modes the COM0A1:0 bits control whether the output should be set, cleared, or toggled at a compare match (See “Compare Match Output Unit” on page 89.).

For detailed timing information refer to Figure 15-8, Figure 15-9, Figure 15-10 and Figure 15-11 in “Timer/Counter Timing Diagrams” on page 95.

Figure 20-2. Clock Generation Logic, Block Diagram



Signal description:

txclk Transmitter clock (Internal Signal).

rxclk Receiver base clock (Internal Signal).

xcki Input from XCK pin (internal Signal). Used for synchronous slave operation.

xcko Clock output to XCK pin (Internal Signal). Used for synchronous master operation.

fosc XTAL pin frequency (System Clock).

20.3.1 Internal Clock Generation – The Baud Rate Generator

Internal clock generation is used for the asynchronous and the synchronous master modes of operation. The description in this section refers to Figure 20-2.

The USART Baud Rate Register (UBRR) and the down-counter connected to it function as a programmable prescaler or baud rate generator. The down-counter, running at system clock (f_{osc}), is loaded with the UBRR value each time the counter has counted down to zero or when the UBRRL Register is written. A clock is generated each time the counter reaches zero. This clock is the baud rate generator clock output (= f_{osc}/(UBRR+1)). The Transmitter divides the baud rate generator clock output by 2, 8 or 16 depending on mode. The baud rate generator output is used directly by the Receiver's clock and data recovery units. However, the recovery units use a state machine that uses 2, 8 or 16 states depending on mode set by the state of the UMSELn, U2Xn and DDR_XCK bits.

Table 20-1 contains equations for calculating the baud rate (in bits per second) and for calculating the UBRR value for each mode of operation using an internally generated clock source.

nine data bits, then the ninth bit (RXB8n) is used for identifying address and data frames. When the frame type bit (the first stop or the ninth bit) is one, the frame contains an address. When the frame type bit is zero the frame is a data frame.

The Multi-processor Communication mode enables several slave MCUs to receive data from a master MCU. This is done by first decoding an address frame to find out which MCU has been addressed. If a particular slave MCU has been addressed, it will receive the following data frames as normal, while the other slave MCUs will ignore the received frames until another address frame is received.

20.9.1 Using MPCMn

For an MCU to act as a master MCU, it can use a 9-bit character frame format (UCSZ = 7). The ninth bit (TXB8n) must be set when an address frame (TXB8n = 1) or cleared when a data frame (TXB = 0) is being transmitted. The slave MCUs must in this case be set to use a 9-bit character frame format.

The following procedure should be used to exchange data in Multi-processor Communication mode:

1. All Slave MCUs are in Multi-processor Communication mode (MPCMn in UCSRnA is set).
2. The Master MCU sends an address frame, and all slaves receive and read this frame. In the Slave MCUs, the RXCn Flag in UCSRnA will be set as normal.
3. Each Slave MCU reads the UDRn Register and determines if it has been selected. If so, it clears the MPCMn bit in UCSRnA, otherwise it waits for the next address byte and keeps the MPCMn setting.
4. The addressed MCU will receive all data frames until a new address frame is received. The other Slave MCUs, which still have the MPCMn bit set, will ignore the data frames.
5. When the last data frame is received by the addressed MCU, the addressed MCU sets the MPCMn bit and waits for a new address frame from master. The process then repeats from 2.

Using any of the 5- to 8-bit character frame formats is possible, but impractical since the Receiver must change between using n and n+1 character frame formats. This makes full-duplex operation difficult since the Transmitter and Receiver uses the same character size setting. If 5- to 8-bit character frames are used, the Transmitter must be set to use two stop bit (USBSn = 1) since the first stop bit is used for indicating the frame type.

Do not use Read-Modify-Write instructions (SBI and CBI) to set or clear the MPCMn bit. The MPCMn bit shares the same I/O location as the TXCn Flag and this might accidentally be cleared when using SBI or CBI instructions.

21. USI – Universal Serial Interface

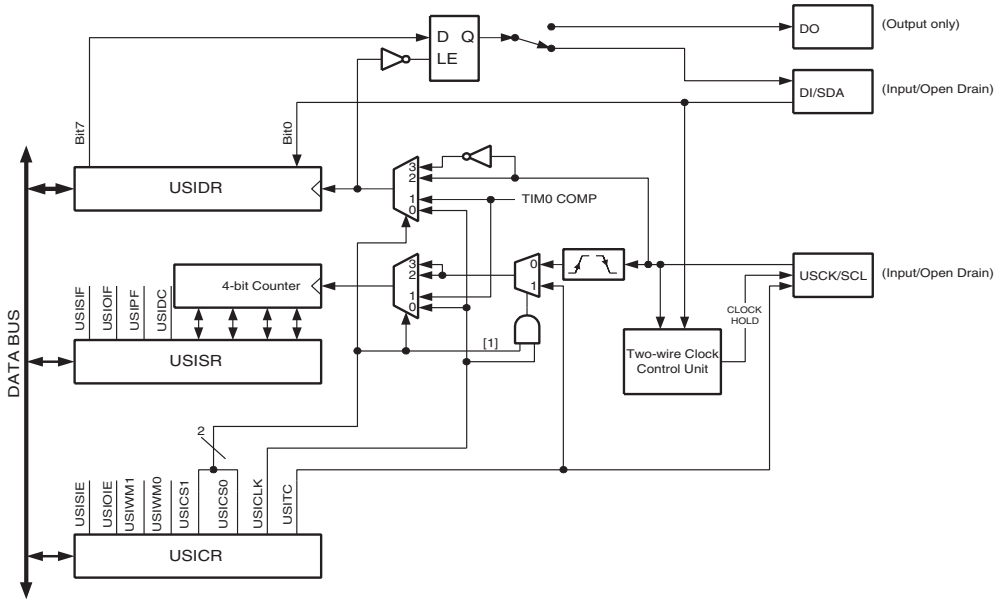
The Universal Serial Interface, or USI, provides the basic hardware resources needed for serial communication. Combined with a minimum of control software, the USI allows significantly higher transfer rates and uses less code space than solutions based on software only. Interrupts are included to minimize the processor load. The main features of the USI are:

- **Two-wire Synchronous Data Transfer (Master or Slave)**
- **Three-wire Synchronous Data Transfer (Master or Slave)**
- **Data Received Interrupt**
- **Wake up from Idle Mode**
- **In Two-wire Mode: Wake-up from All Sleep Modes, Including Power-down Mode**
- **Two-wire Start Condition Detector with Interrupt Capability**

21.1 Overview

A simplified block diagram of the USI is shown on Figure 21-1. For the actual placement of I/O pins, refer to “Pinout ATmega3250/6450” on page 2 and “Pinout ATmega325/645” on page 3. CPU accessible I/O Registers, including I/O bits and I/O pins, are shown in bold. The device-specific I/O Register and bit locations are listed in the “Register Descriptions” on page 192.

Figure 21-1. Universal Serial Interface, Block Diagram



The 8-bit Shift Register is directly accessible via the data bus and contains the incoming and outgoing data. The register has no buffering so the data must be read as quickly as possible to ensure that no data is lost. The most significant bit is connected to one of two output pins depending of the wire mode configuration. A transparent latch is inserted between the Serial Register Output and output pin, which delays the change of data output to the opposite clock edge of the data input sampling. The serial input is always sampled from the Data Input (DI) pin independent of the configuration.

The 4-bit counter can be both read and written via the data bus, and can generate an overflow interrupt. Both the Serial Register and the counter are clocked simultaneously by the same clock source. This allows the counter to count the number of bits received or transmitted and generate

Figure 27-1. Parallel Programming

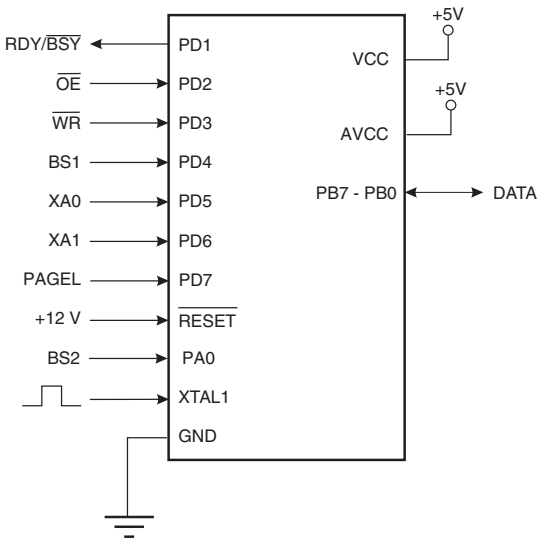


Table 27-6. Pin Name Mapping

Signal Name in Programming Mode	Pin Name	I/O	Function
RDY/$\overline{\text{BSY}}$	PD1	O	0: Device is busy programming, 1: Device is ready for new command.
$\overline{\text{OE}}$	PD2	I	Output Enable (Active low).
$\overline{\text{WR}}$	PD3	I	Write Pulse (Active low).
BS1	PD4	I	Byte Select 1 (“0” selects low byte, “1” selects high byte).
XA0	PD5	I	XTAL Action Bit 0
XA1	PD6	I	XTAL Action Bit 1
PAGEL	PD7	I	Program Memory and EEPROM data Page Load.
BS2	PA0	I	Byte Select 2 (“0” selects low byte, “1” selects 2’nd high byte).
DATA	PB7-0	I/O	Bi-directional Data bus (Output when $\overline{\text{OE}}$ is low).

Table 27-7. Pin Values Used to Enter Programming Mode

Pin	Symbol	Value
PAGEL	Prog_enable[3]	0
XA1	Prog_enable[2]	0
XA0	Prog_enable[1]	0
BS1	Prog_enable[0]	0

Table 27-16. JTAG Programming Instruction Set (Continued)

a = address high bits, **b** = address low bits, **H** = 0 - Low byte, 1 - High Byte, **o** = data out, **i** = data in, x = don't care

Instruction	TDI Sequence	TDO Sequence	Notes
8f. Read Fuses and Lock Bits	0111010_00000000 0111110_00000000 0110010_00000000 0110110_00000000 0110111_00000000	xxxxxxx_xxxxxxxx xxxxxxx_**oooooooo** xxxxxxx_**oooooooo** xxxxxxx_**oooooooo** xxxxxxx_**oooooooo**	(5) Fuse Ext. byte Fuse High byte Fuse Low byte Lock bits
9a. Enter Signature Byte Read	0100011_00001000	xxxxxxx_xxxxxxxx	
9b. Load Address Byte	0000011_**bbbbbbbb**	xxxxxxx_xxxxxxxx	
9c. Read Signature Byte	0110010_00000000 0110011_00000000	xxxxxxx_xxxxxxxx xxxxxxx_**oooooooo**	
10a. Enter Calibration Byte Read	0100011_00001000	xxxxxxx_xxxxxxxx	
10b. Load Address Byte	0000011_**bbbbbbbb**	xxxxxxx_xxxxxxxx	
10c. Read Calibration Byte	0110110_00000000 0110111_00000000	xxxxxxx_xxxxxxxx xxxxxxx_**oooooooo**	
11a. Load No Operation Command	0100011_00000000 0110011_00000000	xxxxxxx_xxxxxxxx xxxxxxx_xxxxxxxx	

Note:
1. This command sequence is not required if the seven MSB are correctly set by the previous command sequence (which is normally the case).
2. Repeat until **o** = "1".
3. Set bits to "0" to program the corresponding Fuse, "1" to unprogram the Fuse.
4. Set bits to "0" to program the corresponding Lock bit, "1" to leave the Lock bit unchanged.
5. "0" = programmed, "1" = unprogrammed.
6. The bit mapping for Fuses Extended byte is listed in Table 27-3 on page 266
7. The bit mapping for Fuses High byte is listed in Table 27-4 on page 267
8. The bit mapping for Fuses Low byte is listed in Table 27-5 on page 267
9. The bit mapping for Lock bits byte is listed in Table 27-1 on page 265
10. Address bits exceeding PCMSB and EEAMSB (Table 27-10 and Table 27-11) are don't care
11. All TDI and TDO sequences are represented by binary digits (0b...).

28.5 System and Reset Characteristics

Table 28-4. Reset, Brown-out and Internal Voltage Reference Characteristics

Symbol	Parameter	Condition	Min	Typ	Max	Units
V_{POT}[1]	Power-on Reset Threshold Voltage (rising)	T_A = -40°C to 85°C	0.7	1.0	1.4	V
	Power-on Reset Threshold Voltage (falling)[1]	T_A = -40°C to 85°C	0.05	0.9	1.3	V
V_{PSR}	Power-on Slope Rate		0.01		4.5	V/ms
V_{RST}	$\overline{\text{RESET}}$ Pin Threshold Voltage	V_{CC} = 3V	$0.2V_{CC}$		$0.85V_{CC}$	V
t_{RST}	Minimum pulse width on $\overline{\text{RESET}}$ Pin	V_{CC} = 3V		800		ns
V_{HYST}	Brown-out Detector Hysteresis			50		mV
t_{BOD}	Min Pulse Width on Brown-out Reset			2		µs
V_{BG}	Bandgap reference voltage	V_{CC}= 2.7V, T_A = 25°C	1.0	1.1	1.2	V
t_{BG}	Bandgap reference start-up time	V_{CC}= 2.7V, T_A = 25°C		40	70	µs
I_{BG}	Bandgap reference current consumption	V_{CC}= 2.7V, T_A = 25°C		15		µA

Notes: 1. The Power-on Reset will not work unless the supply voltage has been below V_{POT} (falling)

Table 28-5. BODLEVEL Fuse Coding[1]

BODLEVEL 2:0 Fuses	Min V_{BOT}	Typ V_{BOT}	Max V_{BOT}	Units
11	BOD Disabled			
10	1.7	1.8	2.0	V
01	2.5	2.7	2.9	
00	4.1	4.3	4.5	

Note: 1. V_{BOT} may be below nominal minimum operating voltage for some devices. For devices where this is the case, the device is tested down to V_{CC} = V_{BOT} during the production test. This guarantees that a Brown-Out Reset will occur before V_{CC} drops to a voltage where correct operation of the microcontroller is no longer guaranteed. The test is performed using BODLEVEL = 10 for Atmel ATmega325/3250/645/6450 and BODLEVEL = 01 for Atmel ATmega325/3250/645/6450V.

Figure 29-31. I/O Pin Sink Current vs. Output Voltage, Port B (V_{CC} = 2.7V)

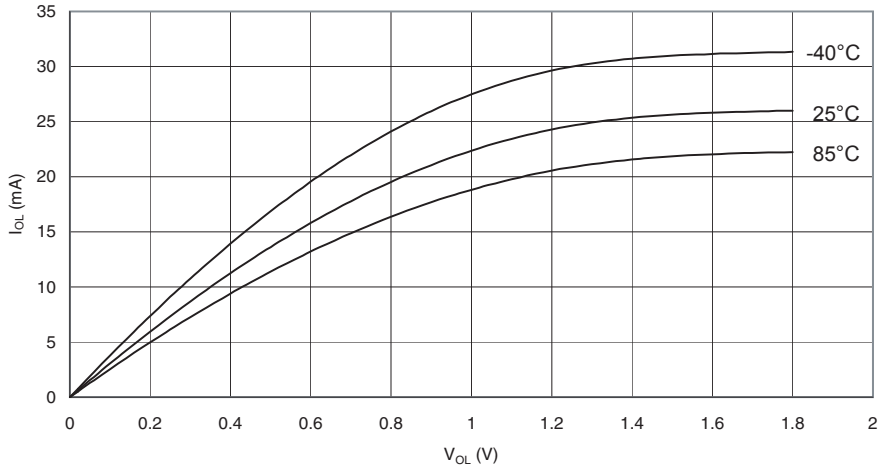


Figure 29-32. I/O Pin Sink Current vs. Output Voltage, Port B (V_{CC} = 1.8V)

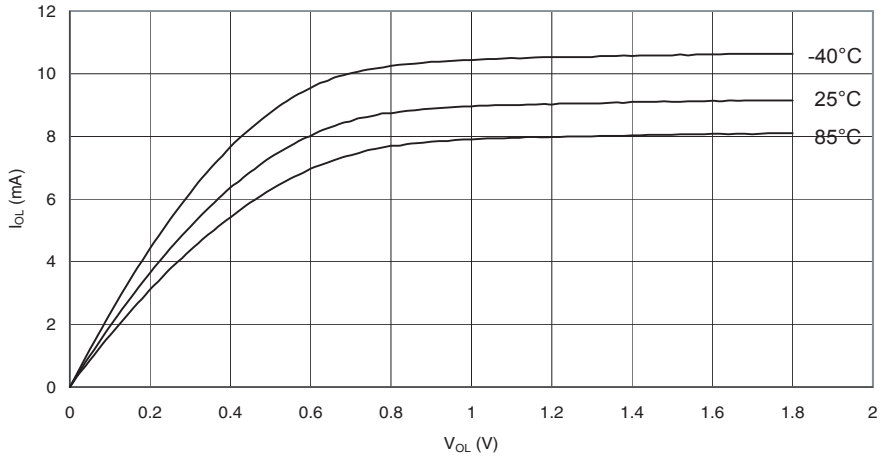
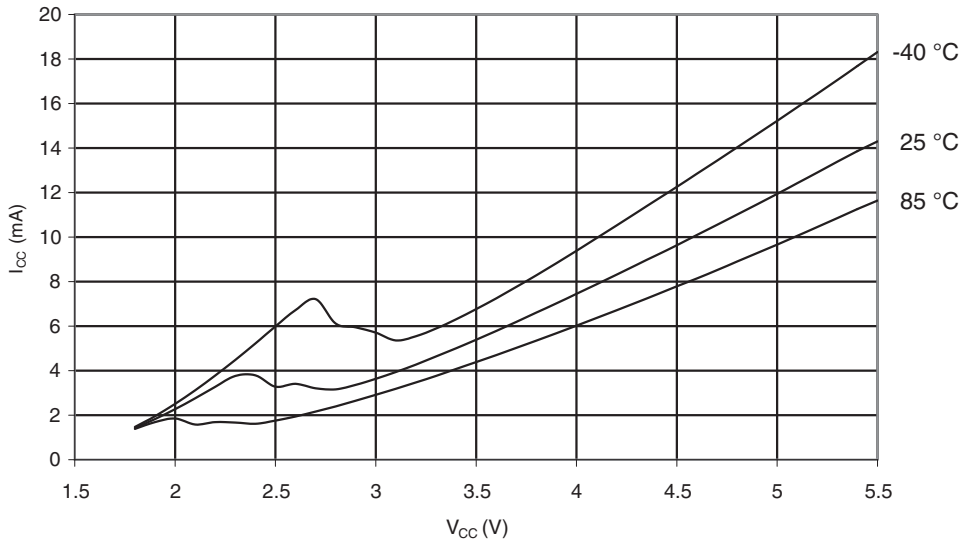
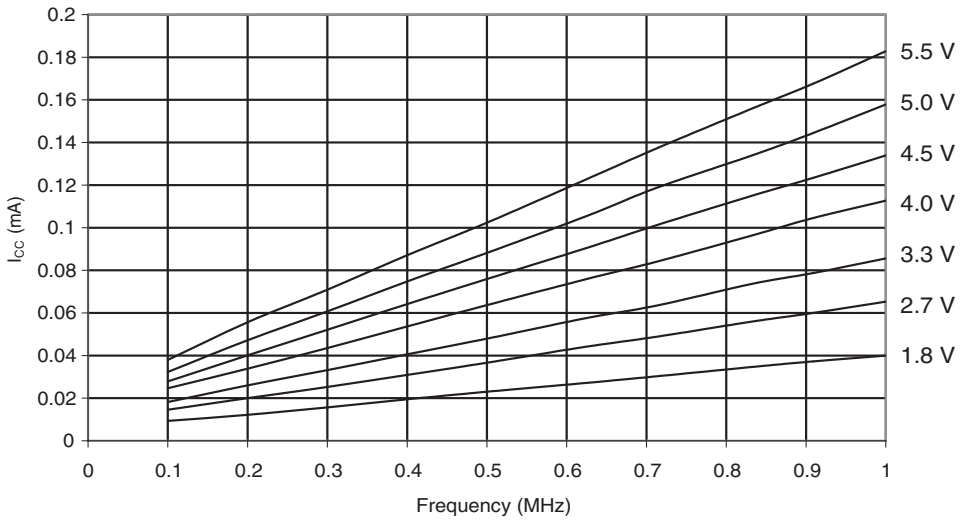


Figure 29-55. Programming Current vs. V_{CC}



29.13 Current Consumption in Reset and Reset Pulsewidth

Figure 29-56. Reset Supply Current vs. V_{CC} (0.1 - 1.0MHz, Excluding Current Through The Reset Pull-up)



Mnemonics	Operands	Description	Operation	Flags	#Clocks
BRTC	k	Branch if T Flag Cleared	if (T = 0) then PC ← PC + k + 1	None	1/2
BRVS	k	Branch if Overflow Flag is Set	if (V = 1) then PC ← PC + k + 1	None	1/2
BRVC	k	Branch if Overflow Flag is Cleared	if (V = 0) then PC ← PC + k + 1	None	1/2
BRIE	k	Branch if Interrupt Enabled	if (I = 1) then PC ← PC + k + 1	None	1/2
BRID	k	Branch if Interrupt Disabled	if (I = 0) then PC ← PC + k + 1	None	1/2
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
DATA TRANSFER INSTRUCTIONS					
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-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
LPM	Rd, Z	Load Program Memory	Rd ← (Z)	None	3
LPM	Rd, Z+	Load Program Memory and Post-Inc	Rd ← (Z), Z ← Z+1	None	3
SPM		Store Program Memory	(Z) ← R1:R0	None	-

32.2 ATmega3250

Speed (MHz)[3]	Power Supply	Ordering Code[2]	Package Type[1]	Operational Range
8	1.8 - 5.5V	ATmega3250V-8AU ATmega3250V-8AUR[4]	100A 100A	Industrial (-40°C to 85°C)
16	2.7 - 5.5V	ATmega3250-16AU ATmega3250-16AUR[4]	100A 100A	

Notes:
1. This device can also be supplied in wafer form. Please contact your local Atmel sales office for detailed ordering information and minimum quantities.
2. Pb-free packaging alternative, complies to the European Directive for Restriction of Hazardous Substances (RoHS directive). Also Halide free and fully Green.
3. For Speed Grades see Figure 28-1 on page 299 and Figure 28-2 on page 299.
4. Tape & Reel

Package Type	
100A	100-lead, 14 x 14 x 1.0mm, 0.5mm Lead Pitch, Thin Profile Plastic Quad Flat Package (TQFP)

32.3 ATmega645

Speed (MHz)[3]	Power Supply	Ordering Code[2]	Package Type[1]	Operational Range
8	1.8 - 5.5V	ATmega645V-8AU ATmega645V-8AUR[4] ATmega645V-8MU ATmega645V-8MUR[4]	64A 64A 64M1 64M1	Industrial (-40°C to 85°C)
16	2.7 - 5.5V	ATmega645-16AU ATmega645-16AUR[4] ATmega645-16MU ATmega645-16MUR[4]	64A 64A 64M1 64M1	

Notes:
1. This device can also be supplied in wafer form. Please contact your local Atmel sales office for detailed ordering information and minimum quantities.
2. Pb-free packaging alternative, complies to the European Directive for Restriction of Hazardous Substances (RoHS directive). Also Halide free and fully Green.
3. For Speed Grades see Figure 28-1 on page 299 and Figure 28-2 on page 299.
4. Tape & Reel

Package Type	
64A	64-lead, 14 x 14 x 1.0mm, Thin Profile Plastic Quad Flat Package (TQFP)
64M1	64-pad, 9 x 9 x 1.0mm, Quad Flat No-Lead/Micro Lead Frame Package (QFN/MLF)