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"

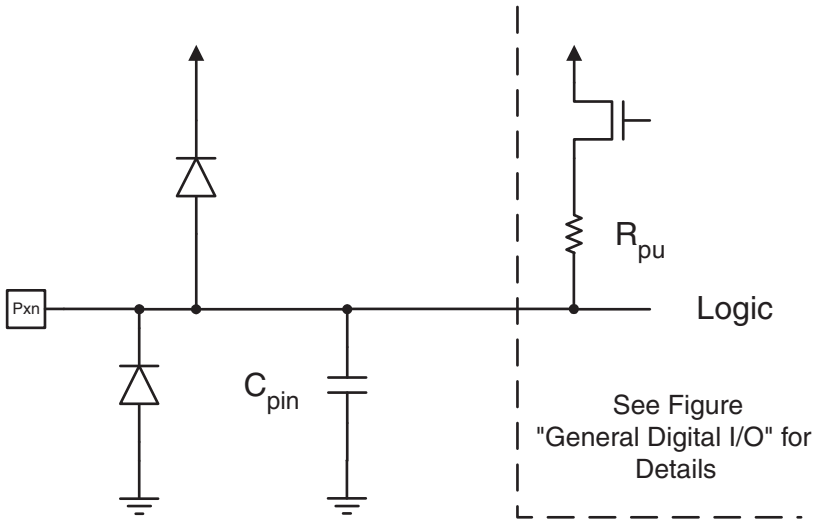
Details	
Product Status	Obsolete
Core Processor	AVR
Core Size	8-Bit
Speed	8MHz
Connectivity	SPI, UART/USART, USI
Peripherals	Brown-out Detect/Reset, LCD, POR, PWM, WDT
Number of I/O	68
Program Memory Size	64KB (32K x 16)
Program Memory Type	FLASH
EEPROM Size	2K x 8
RAM Size	4K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 5.5V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	100-TQFP
Supplier Device Package	100-TQFP (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atmega6490v-8ai

13. I/O-Ports

13.1 Introduction

All AVR ports have true Read-Modify-Write functionality when used as general digital I/O ports. This means that the direction of one port pin can be changed without unintentionally changing the direction of any other pin with the SBI and CBI instructions. The same applies when changing drive value (if configured as output) or enabling/disabling of pull-up resistors (if configured as input). Each output buffer has symmetrical drive characteristics with both high sink and source capability. Port B has a higher pin driver strength than the other ports, but all the pin drivers are strong enough to drive LED displays directly. All port pins have individually selectable pull-up resistors with a supply-voltage invariant resistance. All I/O pins have protection diodes to both V_{CC} and Ground as indicated in Figure 13-1. Refer to "Electrical Characteristics" on page 326 for a complete list of parameters. If exceeding the pin voltage "Absolute Maximum Ratings", resulting currents can harm the device if not limited accordingly. For segment pins used as general I/O, the same situation can also influence the LCD voltage level.

Figure 13-1. I/O Pin Equivalent Schematic



All registers and bit references in this section are written in general form. A lower case "x" represents the numbering letter for the port, and a lower case "n" represents the bit number. However, when using the register or bit defines in a program, the precise form must be used. For example, PORTB3 for bit no. 3 in Port B, here documented generally as PORTxn. The physical I/O Registers and bit locations are listed in "Register Description" on page 87.

Three I/O memory address locations are allocated for each port, one each for the Data Register – PORTx, Data Direction Register – DDRx, and the Port Input Pins – PINx. The Port Input Pins I/O location is read only, while the Data Register and the Data Direction Register are read/write. However, writing a logic one to a bit in the PINx Register, will result in a toggle in the corresponding bit in the Data Register. In addition, the Pull-up Disable – PUD bit in MCUCR disables the pull-up function for all pins in all ports when set.

Using the I/O port as General Digital I/O is described in "Ports as General Digital I/O" on page 60. Most port pins are multiplexed with alternate functions for the peripheral features on the device. How each alternate function interferes with the port pin is described in "Alternate Port

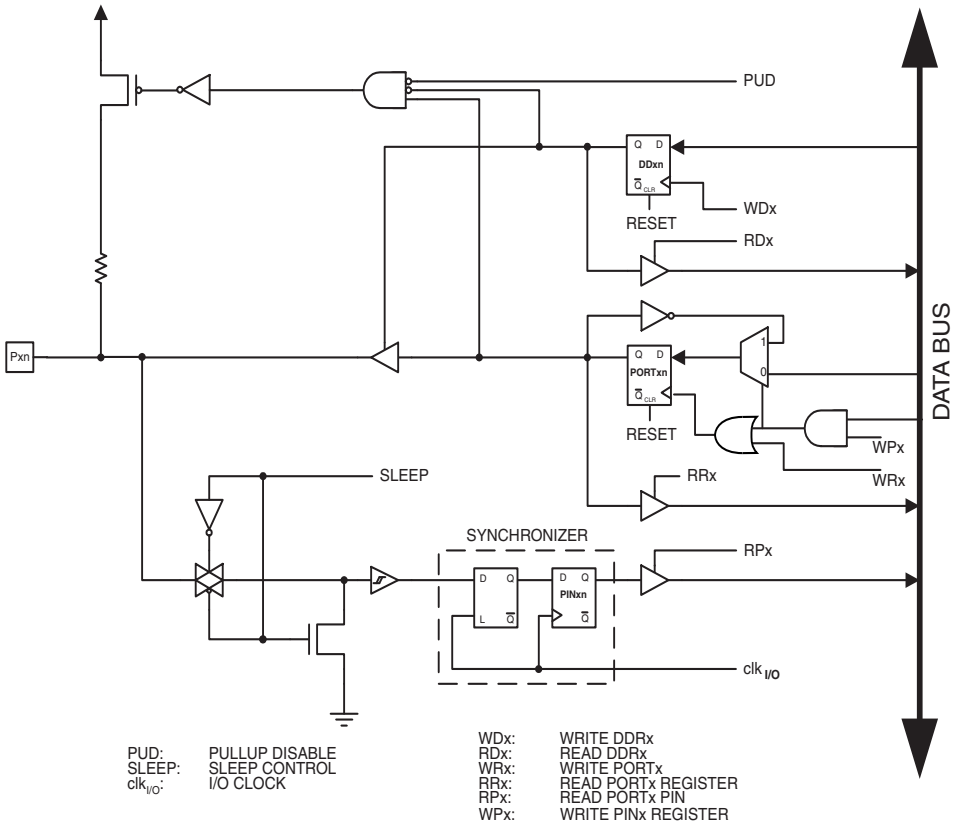
Functions” on page 65. Refer to the individual module sections for a full description of the alternate functions.

Note that enabling the alternate function of some of the port pins does not affect the use of the other pins in the port as general digital I/O.

13.2 Ports as General Digital I/O

The ports are bi-directional I/O ports with optional internal pull-ups. Figure 13-2 shows a functional description of one I/O-port pin, here generically called Pxn.

Figure 13-2. General Digital I/O(1)



Note: 1. WRx, WPx, WDx, RRx, RPx, and RDx are common to all pins within the same port. $clk_{I/O}$, SLEEP, and PUD are common to all ports.

13.2.1 Configuring the Pin

Each port pin consists of three register bits: DDxn, PORTxn, and PINxn. As shown in “Register Description” on page 87, the DDxn bits are accessed at the DDRx I/O address, the PORTxn bits at the PORTx I/O address, and the PINxn bits at the PINx I/O address.

The DDxn bit in the DDRx Register selects the direction of this pin. If DDxn is written logic one, Pxn is configured as an output pin. If DDxn is written logic zero, Pxn is configured as an input pin.

If PORTxn is written logic one when the pin is configured as an input pin, the pull-up resistor is activated. To switch the pull-up resistor off, PORTxn has to be written logic zero or the pin has to

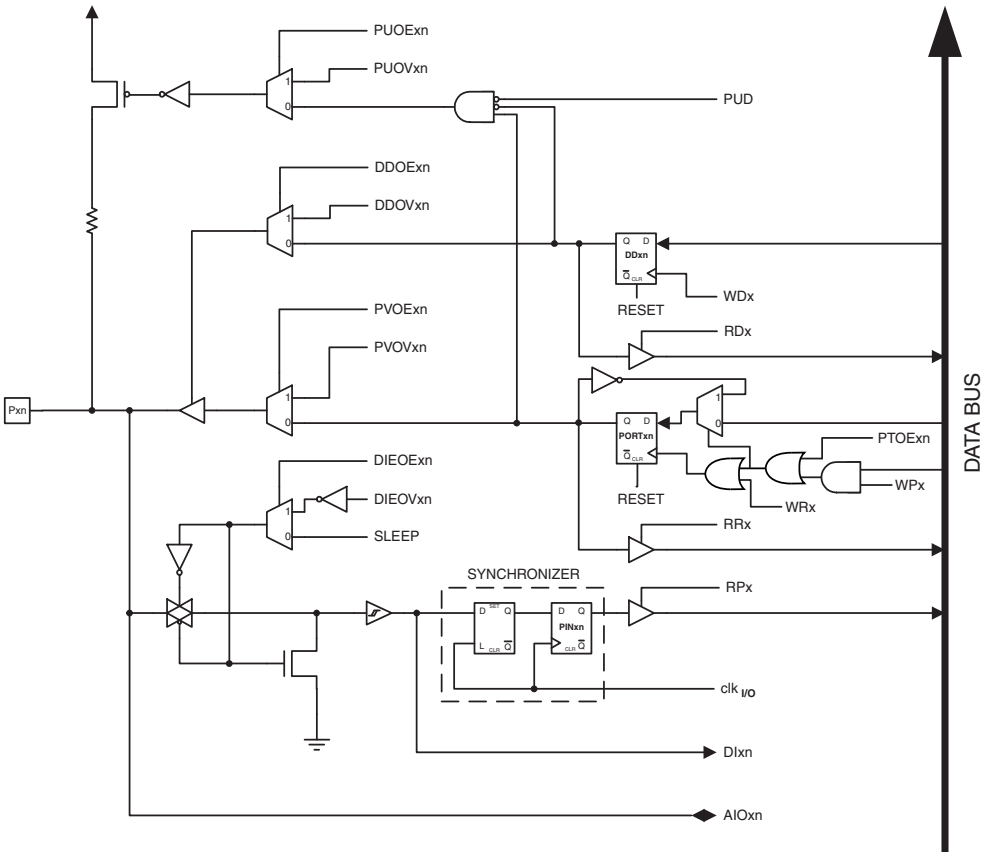
ing inputs should be avoided to reduce current consumption in all other modes where the digital inputs are enabled (Reset, Active mode and Idle mode).

The simplest method to ensure a defined level of an unused pin, is to enable the internal pull-up. In this case, the pull-up will be disabled during reset. If low power consumption during reset is important, it is recommended to use an external pull-up or pull-down. Connecting unused pins directly to V_{CC} or GND is not recommended, since this may cause excessive currents if the pin is accidentally configured as an output.

13.3 Alternate Port Functions

Most port pins have alternate functions in addition to being general digital I/Os. Figure 13-5 shows how the port pin control signals from the simplified Figure 13-2 can be overridden by alternate functions. The overriding signals may not be present in all port pins, but the figure serves as a generic description applicable to all port pins in the AVR microcontroller family.

Figure 13-5. Alternate Port Functions[1]



PUOExn:	Pxn PULL-UP OVERRIDE ENABLE	PUD:	PULLUP DISABLE
PUOVxn:	Pxn PULL-UP OVERRIDE VALUE	WDx:	WRITE DDRx
DDOExn:	Pxn DATA DIRECTION OVERRIDE ENABLE	RDx:	READ DDRx
DDOVxn:	Pxn DATA DIRECTION OVERRIDE VALUE	RRx:	READ PORTx REGISTER
PVOExn:	Pxn PORT VALUE OVERRIDE ENABLE	WRx:	WRITE PORTx
PVOVxn:	Pxn PORT VALUE OVERRIDE VALUE	RPx:	READ PORTx PIN
DIEOExn:	Pxn DIGITAL INPUT-ENABLE OVERRIDE ENABLE	WPx:	WRITE PINx
DIEOVxn:	Pxn DIGITAL INPUT-ENABLE OVERRIDE VALUE	$clk_{I/O}$:	I/O CLOCK
SLEEP:	SLEEP CONTROL	DIxn:	DIGITAL INPUT PIN n ON PORTx
PTOExn:	Pxn, PORT TOGGLE OVERRIDE ENABLE	AIOxn:	ANALOG INPUT/OUTPUT PIN n ON PORTx

Note: 1. WRx, WPx, WDx, RRx, RPx, and RDx are common to all pins within the same port. $clk_{I/O}$, SLEEP, and PUD are common to all ports. All other signals are unique for each pin.

Table 13-8. 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	–	–	–	–

13.3.3 Alternate Functions of Port C

The Port C has an alternate function as SEG for the LCD Controller.

Table 13-9. Port C Pins Alternate Functions (SEG refers to 100-pin/64-pin pinout)

Port Pin	Alternate Function
PC7	SEG (LCD Front Plane 5/5)
PC6	SEG (LCD Front Plane 6/6)
PC5	SEG (LCD Front Plane 11/7)
PC4	SEG (LCD Front Plane 12/8)
PC3	SEG (LCD Front Plane 13/9)
PC2	SEG (LCD Front Plane14/10)
PC1	SEG (LCD Front Plane 15/11)
PC0	SEG (LCD Front Plane 16/12)

The alternate pin configuration is as follows:

- **SEG – Port D, Bit 7:0**

SEG, LCD front plane 5/5, 6/6, 11/7-16/12.

Table 13-10 and Table 13-11 relate the alternate functions of Port C to the overriding signals shown in Figure 13-5 on page 65.

- **TDO, ADC6 – Port F, Bit 6**

ADC6, Analog to Digital Converter, Channel 6.

TDO, JTAG Test Data Out: Serial output data from Instruction Register or Data Register. When the JTAG interface is enabled, this pin can not be used as an I/O pin. In TAP states that shift out data, the TDO pin drives actively. In other states the pin is pulled high.

- **TMS, ADC5 – Port F, Bit 5**

ADC5, Analog to Digital Converter, Channel 5.

TMS, JTAG Test mode Select: This pin is used for navigating through the TAP-controller state machine. When the JTAG interface is enabled, this pin can not be used as an I/O pin.

- **TCK, ADC4 – Port F, Bit 4**

ADC4, Analog to Digital Converter, Channel 4.

TCK, JTAG Test Clock: JTAG operation is synchronous to TCK. When the JTAG interface is enabled, this pin can not be used as an I/O pin.

- **ADC3 - ADC0 – Port F, Bit 3:0**

Analog to Digital Converter, Channel 3-0.

Table 13-19. Overriding Signals for Alternate Functions in PF7:PF4

Signal Name	PF7/ADC7/TDI	PF6/ADC6/TDO	PF5/ADC5/TMS	PF4/ADC4/TCK
PUOE	JTAGEN	JTAGEN	JTAGEN	JTAGEN
PUOV	1	1	1	1
DDOE	JTAGEN	JTAGEN	JTAGEN	JTAGEN
DDOV	0	SHIFT_IR + SHIFT_DR	0	0
PVOE	0	JTAGEN	0	0
PVOV	0	TDO	0	0
PTOE	–	–	–	–
DIEOE	JTAGEN	JTAGEN	JTAGEN	JTAGEN
DIEOV	0	0	0	0
DI	–	–	–	–
AIO	TDI ADC7 INPUT	ADC6 INPUT	TMS ADC5 INPUT	TCK ADC4 INPUT

18.3 $\overline{SS}$ Pin Functionality

18.3.1 Slave Mode

When the SPI is configured as a Slave, the Slave Select ($\overline{SS}$) 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 driven high.

The $\overline{SS}$ pin is useful for packet/byte synchronization to keep the slave bit counter synchronous with the master clock generator. When the $\overline{SS}$ pin is driven high, the SPI slave will immediately reset the send and receive logic, and drop any partially received data in the Shift Register.

18.3.2 Master Mode

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. Typically, the pin will be driving the $\overline{SS}$ pin of the SPI Slave.

If $\overline{SS}$ is configured as an input, it must be held high to ensure Master SPI operation. If the $\overline{SS}$ pin is driven low by peripheral circuitry when the SPI is configured as a 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 is set, the interrupt routine will be executed.

Thus, when interrupt-driven SPI transmission 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. If the MSTR bit has been cleared by a slave select, it must be set by the user to re-enable SPI Master mode.

19. USART0

19.1 Features

The Universal Synchronous and Asynchronous serial Receiver and Transmitter (USART) is a highly flexible serial communication device. The main features are:

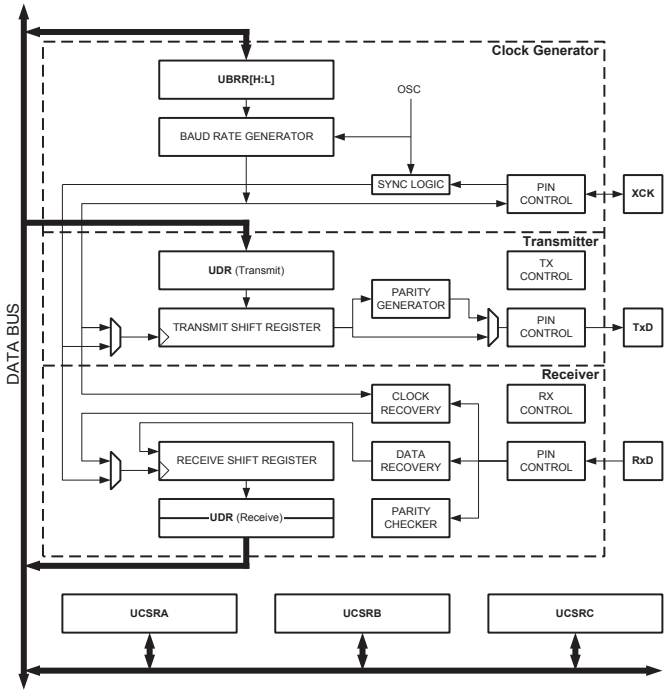
- **Full Duplex Operation (Independent Serial Receive and Transmit Registers)**
- **Asynchronous or Synchronous Operation**
- **Master or Slave Clocked Synchronous Operation**
- **High Resolution Baud Rate Generator**
- **Supports Serial Frames with 5, 6, 7, 8, or 9 Data Bits and 1 or 2 Stop Bits**
- **Odd or Even Parity Generation and Parity Check Supported by Hardware**
- **Data OverRun Detection**
- **Framing Error Detection**
- **Noise Filtering Includes False Start Bit Detection and Digital Low Pass Filter**
- **Three Separate Interrupts on TX Complete, TX Data Register Empty and RX Complete**
- **Multi-processor Communication Mode**
- **Double Speed Asynchronous Communication Mode**

19.2 Overview

A simplified block diagram of the USART Transmitter is shown in Figure 19-1. CPU accessible I/O Registers and I/O pins are shown in bold.

The Power Reduction USART bit, PRUSART0, in “PRR – Power Reduction Register” on page 40 must be written to zero to enable USART0 module.

Figure 19-1. USART Block Diagram[1]



Note: 1. Refer to Figure 1-1 on page 2, Figure 1-2 on page 3, “Alternate Functions of Port E” on page 75 for USART pin placement.

The dashed boxes in the block diagram separate the three main parts of the USART (listed from the top): Clock Generator, Transmitter and Receiver. Control Registers are shared by all units. The Clock Generation logic consists of synchronization logic for external clock input used by synchronous slave operation, and the baud rate generator. The XCK (Transfer Clock) pin is only used by synchronous transfer mode. The Transmitter consists of a single write buffer, a serial Shift Register, Parity Generator and Control logic for handling different serial frame formats. The write buffer allows a continuous transfer of data without any delay between frames. The Receiver is the most complex part of the USART module due to its clock and data recovery units. The recovery units are used for asynchronous data reception. In addition to the recovery units, the Receiver includes a Parity Checker, Control logic, a Shift Register and a two level receive buffer (UDRn). The Receiver supports the same frame formats as the Transmitter, and can detect Frame Error, Data OverRun and Parity Errors.

19.2.1 AVR USART vs. AVR UART – Compatibility

The USART is fully compatible with the AVR UART regarding:

- Bit locations inside all USART Registers.
- Baud Rate Generation.
- Transmitter Operation.
- Transmit Buffer Functionality.
- Receiver Operation.

However, the receive buffering has two improvements that will affect the compatibility in some special cases:

- A second Buffer Register has been added. The two Buffer Registers operate as a circular FIFO buffer. Therefore the UDRn must only be read once for each incoming data! More important is the fact that the Error Flags (FEn and DORn) and the ninth data bit (RXB8n) are buffered with the data in the receive buffer. Therefore the status bits must always be read before the UDRn Register is read. Otherwise the error status will be lost since the buffer state is lost.
- The Receiver Shift Register can now act as a third buffer level. This is done by allowing the received data to remain in the serial Shift Register (see Figure 19-1) if the Buffer Registers are full, until a new start bit is detected. The USART is therefore more resistant to Data OverRun (DORn) error conditions.

The following control bits have changed name, but have same functionality and register location:

- CHR9 is changed to UCSZn2.
- OR is changed to DORn.

19.3 Clock Generation

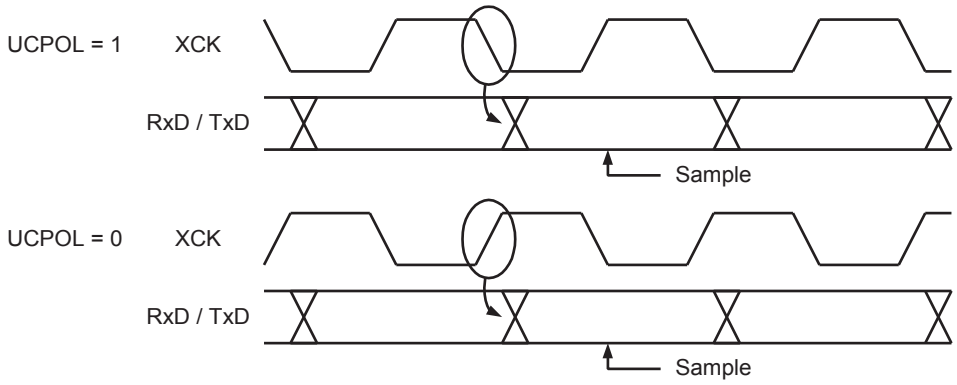
The Clock Generation logic generates the base clock for the Transmitter and Receiver. The USART supports four modes of clock operation: Normal asynchronous, Double Speed asynchronous, Master synchronous and Slave synchronous mode. The UMSELn bit in USART Control and Status Register C (UCSRnC) selects between asynchronous and synchronous operation. Double Speed (asynchronous mode only) is controlled by the U2Xn found in the UCSRnA Register. When using synchronous mode (UMSELn = 1), the Data Direction Register for the XCK pin (DDR_XCK) controls whether the clock source is internal (Master mode) or external (Slave mode). The XCK pin is only active when using synchronous mode.

Figure 19-2 shows a block diagram of the clock generation logic.

19.3.4 Synchronous Clock Operation

When synchronous mode is used (UMSELn = 1), the XCK pin will be used as either clock input (Slave) or clock output (Master). The dependency between the clock edges and data sampling or data change is the same. The basic principle is that data input (on RxD) is sampled at the opposite XCK clock edge of the edge the data output (TxD) is changed.

Figure 19-3. Synchronous Mode XCK Timing.



The UCPOLn bit UCRSC selects which XCK clock edge is used for data sampling and which is used for data change. As Figure 19-3 shows, when UCPOLn is zero the data will be changed at rising XCK edge and sampled at falling XCK edge. If UCPOLn is set, the data will be changed at falling XCK edge and sampled at rising XCK edge.

19.4 Frame Formats

A serial frame is defined to be one character of data bits with synchronization bits (start and stop bits), and optionally a parity bit for error checking. The USART accepts all 30 combinations of the following as valid frame formats:

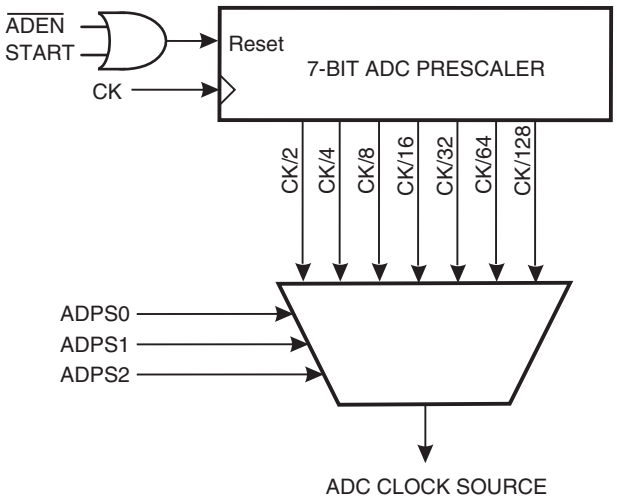
- 1 start bit
- 5, 6, 7, 8, or 9 data bits
- no, even or odd parity bit
- 1 or 2 stop bits

A frame starts with the start bit followed by the least significant data bit. Then the next data bits, up to a total of nine, are succeeding, ending with the most significant bit. If enabled, the parity bit is inserted after the data bits, before the stop bits. When a complete frame is transmitted, it can be directly followed by a new frame, or the communication line can be set to an idle (high) state. Figure 19-4 illustrates the possible combinations of the frame formats. Bits inside brackets are optional.

If Auto Triggering is enabled, single conversions can be started by writing ADSC in ADCSRA to one. ADSC can also be used to determine if a conversion is in progress. The ADSC bit will be read as one during a conversion, independently of how the conversion was started.

22.4 Prescaling and Conversion Timing

Figure 22-3. ADC Prescaler



By default, the successive approximation circuitry requires an input clock frequency between 50kHz and 200kHz to get maximum resolution. If a lower resolution than 10 bits is needed, the input clock frequency to the ADC can be higher than 200kHz to get a higher sample rate.

The ADC module contains a prescaler, which generates an acceptable ADC clock frequency from any CPU frequency above 100kHz. The prescaling is set by the ADPS bits in ADCSRA. The prescaler starts counting from the moment the ADC is switched on by setting the ADEN bit in ADCSRA. The prescaler keeps running for as long as the ADEN bit is set, and is continuously reset when ADEN is low.

When initiating a single ended conversion by setting the ADSC bit in ADCSRA, the conversion starts at the following rising edge of the ADC clock cycle.

A normal conversion takes 13 ADC clock cycles. The first conversion after the ADC is switched on (ADEN in ADCSRA is set) takes 25 ADC clock cycles in order to initialize the analog circuitry.

When the bandgap reference voltage is used as input to the ADC, it will take a certain time for the voltage to stabilize. If not stabilized the first value read after the first conversion may be wrong.

The actual sample-and-hold takes place 1.5 ADC clock cycles after the start of a normal conversion and 13.5 ADC clock cycles after the start of an first conversion. When a conversion is complete, the result is written to the ADC Data Registers, and ADIF is set. In Single Conversion mode, ADSC is cleared simultaneously. The software may then set ADSC again, and a new conversion will be initiated on the first rising ADC clock edge.

When Auto Triggering is used, the prescaler is reset when the trigger event occurs. This assures a fixed delay from the trigger event to the start of conversion. In this mode, the sample-and-hold takes place two ADC clock cycles after the rising edge on the trigger source signal. Three additional CPU clock cycles are used for synchronization logic. When using Differential mode, along

23. LCD Controller

The LCD Controller/driver is intended for monochrome passive liquid crystal display (LCD) with up to four common terminals and up to 25/40 segment terminals.

23.1 Features

- **Display Capacity of 25/40 Segments and Four Common Terminals**
- **Support Static, 1/2, 1/3 and 1/4 Duty**
- **Support Static, 1/2, 1/3 Bias**
- **On-chip LCD Power Supply, only One External Capacitor needed**
- **Display Possible in Power-save Mode for Low Power Consumption**
- **Software Selectable Low Power Waveform Capability**
- **Flexible Selection of Frame Frequency**
- **Software Selection between System Clock or an External Asynchronous Clock Source**
- **Equal Source and Sink Capability to maximize LCD Life Time**
- **LCD Interrupt Can be Used for Display Data Update or Wake-up from Sleep Mode**
- **Segment and Common Pins not Needed for Driving the Display Can be Used as Ordinary I/O Pins**
- **Latching of Display Data gives Full Freedom in Register Update**

23.1.1 Overview

A simplified block diagram of the LCD Controller/Driver is shown in Figure 23-1. For the actual placement of I/O pins, refer to "Pinout ATmega3290/6490" on page 2 and "Pinout ATmega329/649" on page 3.

An LCD consists of several segments (pixels or complete symbols) which can be visible or non visible. A segment has two electrodes with liquid crystal between them. When a voltage above a threshold voltage is applied across the liquid crystal, the segment becomes visible.

The voltage must alternate to avoid an electrophoresis effect in the liquid crystal, which degrades the display. Hence the waveform across a segment must not have a DC-component.

The PRLCD bit in "Power Reduction Register" on page 37 must be written to zero to enable the LCD module.

23.1.2 Definitions

Several terms are used when describing LCD. The definitions in Table 23-1 are used throughout this document.

Table 23-1. Definitions

LCD	A passive display panel with terminals leading directly to a segment
Segment	The least viewing element (pixel) which can be on or off
Common	Denotes how many segments are connected to a segment terminal
Duty	1/(Number of common terminals on a actual LCD display)
Bias	1/(Number of voltage levels used driving a LCD display -1)
Frame Rate	Number of times the LCD segments is energized per second.

When EXCLK in ASSR Register 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. See “Asynchronous Operation of Timer/Counter2” on page 151 for further details.

Before entering Power-down mode, Standby mode or ADC Noise Reduction mode with synchronous LCD clock selected, the user have to disable the LCD. Refer to “Disabling the LCD” on page 237.

23.2.7 Display Blanking

When LCDBL is written to one, the LCD is blanked after completing the current frame. All segments and common pins are connected to GND, discharging the LCD. Display memory is preserved. Display blanking should be used before disabling the LCD to avoid DC voltage across segments, and a slowly fading image.

23.2.8 Port Mask

For LCD with less than 25/40 segment terminals, it is possible to mask some of the unused pins and use them as ordinary port pins instead. Refer to Table 23-3 for details. Unused common pins are automatically configured as port pins.

- **Bit 7 – LCDCS: LCD Clock Select**

When this bit is written to zero, the system clock is used. When this bit is written to one, the external asynchronous clock source is used. The asynchronous clock source is either Timer/Counter Oscillator or external clock, depending on EXCLK in ASSR. See "Asynchronous Operation of Timer/Counter2" on page 151 for further details.

- **Bit 6 – LCD2B: LCD 1/2 Bias Select**

When this bit is written to zero, 1/3 bias is used. When this bit is written to one, ½ bias is used. Refer to the LCD Manufacture for recommended bias selection.

- **Bit 5:4 – LCDMUX1:0: LCD Mux Select**

The LCDMUX1:0 bits determine the duty cycle. Common pins that are not used are ordinary port pins. The different duty selections are shown in Table 23-2.

Table 23-2. LCD Duty Select

LCDMUX1	LCDMUX0	Duty	Bias	COM Pin	I/O Port Pin
0	0	Static	Static	COM0	COM1:3
0	1	1/2	1/2 or 1/3(1)	COM0:1	COM2:3
1	0	1/3	1/2 or 1/3(1)	COM0:2	COM3
1	1	1/4	1/2 or 1/3(1)	COM0:3	None

Note: 1. 1/2 bias when LCD2B is written to one and 1/3 otherwise.

- **Bits 3:0 – LCDPM3:0: LCD Port Mask**

The LCDPM3:0 bits determine the number of port pins to be used as segment drivers. The different selections are shown in Table 23-3. Unused pins can be used as ordinary port pins.

Table 23-3. LCD Port Mask (Values in bold are only available in ATmega3290/6490)

LCDPM3	LCDPM2	LCDPM1	LCDPM0	I/O Port in Use as Segment Driver	Maximum Number of Segments
0	0	0	0	SEG0:12	13
0	0	0	1	SEG0:14	15
0	0	1	0	SEG0:16	17
0	0	1	1	SEG0:18	19
0	1	0	0	SEG0:20	21
0	1	0	1	SEG0:22	23
0	1	1	0	SEG0:23	24
0	1	1	1	SEG0:24	25
1	**0**	**0**	**0**	**SEG0:26**	**27**
1	**0**	**0**	**1**	**SEG0:28**	**29**
1	**0**	**1**	**0**	**SEG0:30**	**31**
1	**0**	**1**	**1**	**SEG0:32**	**33**
1	**1**	**0**	**0**	**SEG0:34**	**35**

Table 23-5. LCD Clock Divide

LCDCD2	LCDCD1	LCDCD0	Output from Prescaler divided by (D) :	clk_{LCD} = 32.768kHz, N = 16, and Duty = 1/4, gives a frame rate of:
0	0	0	1	256Hz
0	0	1	2	128Hz
0	1	0	3	85.3Hz
0	1	1	4	64Hz
1	0	0	5	51.2Hz
1	0	1	6	42.7Hz
1	1	0	7	36.6Hz
1	1	1	8	32Hz

The frame frequency can be calculated by the following equation:

$$f_{frame} = \frac{f_{clk_{LCD}}}{(K \cdot N \cdot D)}$$

Where:

N = prescaler divider (16, 64, 128, 256, 512, 1024, 2048, or 4096).

K = 8 for duty = 1/4, 1/2, and static.

K = 6 for duty = 1/3.

D = Division factor (see Table 23-5)

This is a very flexible scheme, and users are encouraged to calculate their own table to investigate the possible frame rates from the formula above. Note when using 1/3 duty the frame rate is increased with 33% when Frame Rate Register is constant. Example of frame rate calculation is shown in Table 23-6.

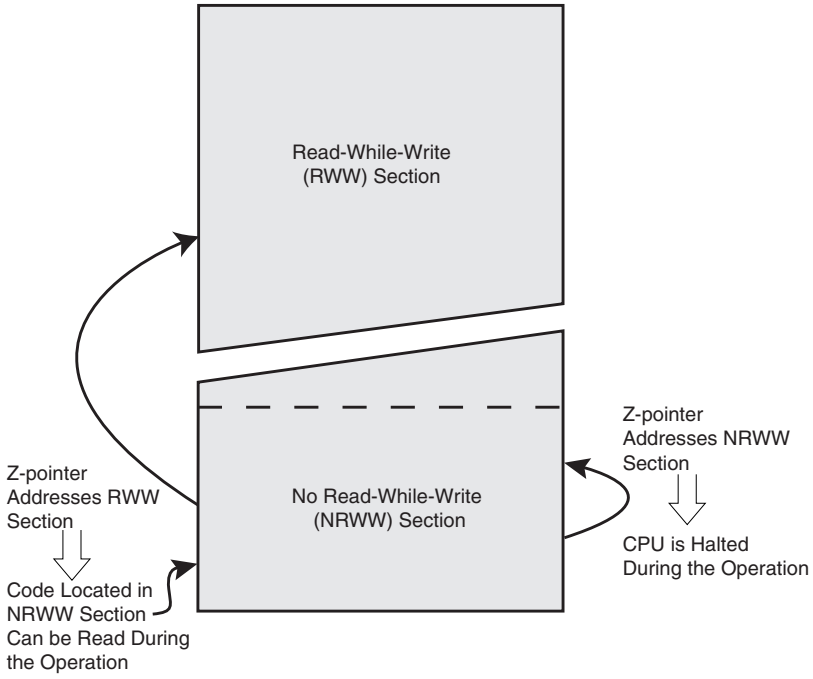
Table 23-6. Example of frame rate calculation

clk_{LCD}	duty	K	N	LCDCD2:0	D	Frame Rate
4MHz	1/4	8	2048	011	4	4000000/(8*2048*4) = 61 Hz
4MHz	1/3	6	2048	011	4	4000000/(6*2048*4) = 81 Hz
32.768kHz	Static	8	16	000	1	32768/(8*16*1) = 256 Hz
32.768kHz	1/2	8	16	100	5	32768/(8*16*5) = 51 Hz

Table 25-8. ATmega3290/6490 Boundary-scan Order, 100-pin (Continued)

Bit Number	Signal Name	Module
99	PD5.Pull-up_Enable	
98	PD6.Data	
97	PD6.Control	
96	PD6.Pull-up_Enable	
95	PD7.Data	
94	PD7.Control	
93	PD7.Pull-up_Enable	
92	PG0.Data	Port G
91	PG0.Control	
90	PG0.Pull-up_Enable	
89	PG1.Data	
88	PG1.Control	
87	PG1.Pull-up_Enable	
86	PC0.Data	Port C
85	PC0.Control	
84	PC0.Pull-up_Enable	
83	PC1.Data	
82	PC1.Control	
81	PC1.Pull-up_Enable	
80	PC2.Data	
79	PC2.Control	
78	PC2.Pull-up_Enable	
77	PC3.Data	
76	PC3.Control	
75	PC3.Pull-up_Enable	
74	PC4.Data	
73	PC4.Control	
72	PC4.Pull-up_Enable	
71	PC5.Data	
70	PC5.Control	
69	PC5.Pull-up_Enable	
68	PH0.Data	Port H
67	PH0.Control	
66	PH0.Pull-up_Enable	
65	PH1.Data	
64	PH1.Control	

Figure 26-1. Read-While-Write vs. No Read-While-Write

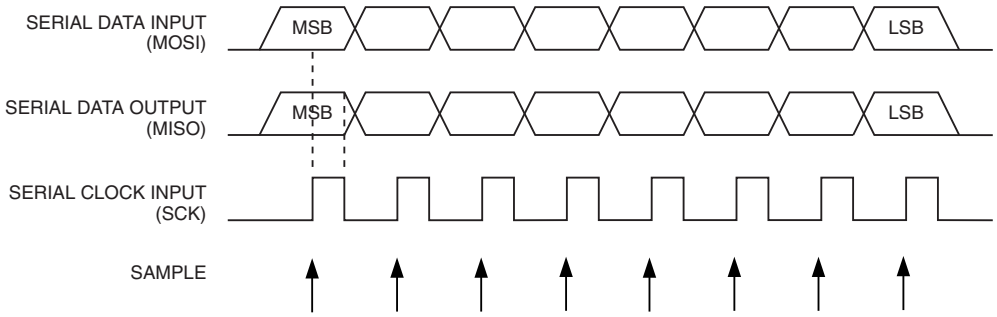


8. Power-off sequence (if needed):
   Set $\overline{\text{RESET}}$ to “1”.
   Turn V_{CC} power off.

Table 27-14. Minimum Wait Delay Before Writing the Next Flash or EEPROM Location

Symbol	Minimum Wait Delay
t_{WD_FUSE}	4.5ms
t_{WD_FLASH}	4.5ms
t_{WD_EEPROM}	9.0ms
t_{WD_ERASE}	9.0ms

Figure 27-11. Serial Programming Waveforms



27.7.3 Serial Programming Instruction set

Table 27-15 and Figure 27-12 on page 312 describes the Instruction set.

Table 27-15. Serial Programming Instruction Set

Instruction/Operation	Instruction Format			
	Byte 1	Byte 2	Byte 3	Byte4
Programming Enable	$AC	$53	$00	$00
Chip Erase (Program Memory/EEPROM)	$AC	$80	$00	$00
Poll RDY/$\overline{\text{BSY}}$	$F0	$00	$00	data byte out
Load Instructions				
Load Extended Address byte[1]	$4D	$00	Extended adr	$00
Load Program Memory Page, High byte	$48	$00	adr LSB	high data byte in
Load Program Memory Page, Low byte	$40	$00	adr LSB	low data byte in
Load EEPROM Memory Page (page access)	$C1	$00	0000 00aa / 0000 0aaa	data byte in
Read Instructions				
Read Program Memory, High byte	$28	adr MSB	adr LSB	high data byte out
Read Program Memory, Low byte	$20	adr MSB	adr LSB	low data byte out
Read EEPROM Memory	$A0	0000 00aa / 0000 0aaa	aaaa aaaa	data byte out

29.0.7 Pin Pull-up

Figure 29-15. I/O Pin Pull-up Resistor Current vs. Input Voltage (V_{CC} = 5V)

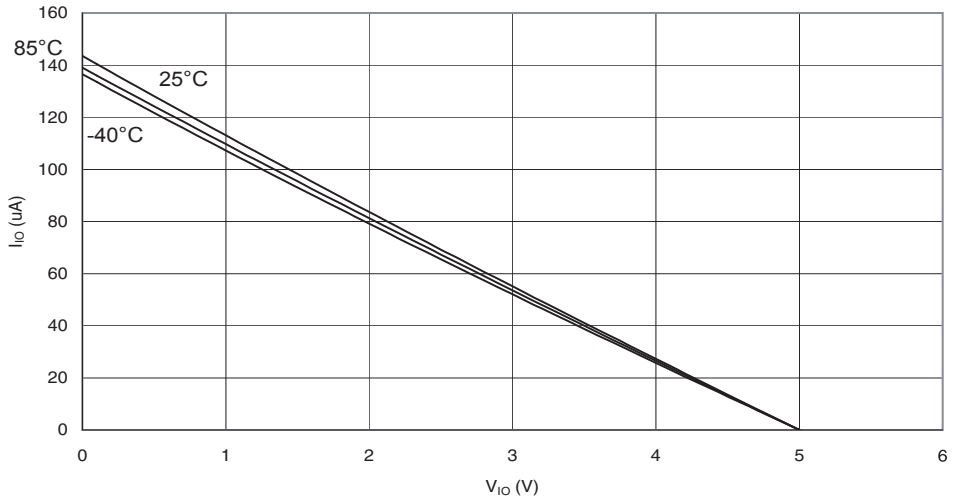
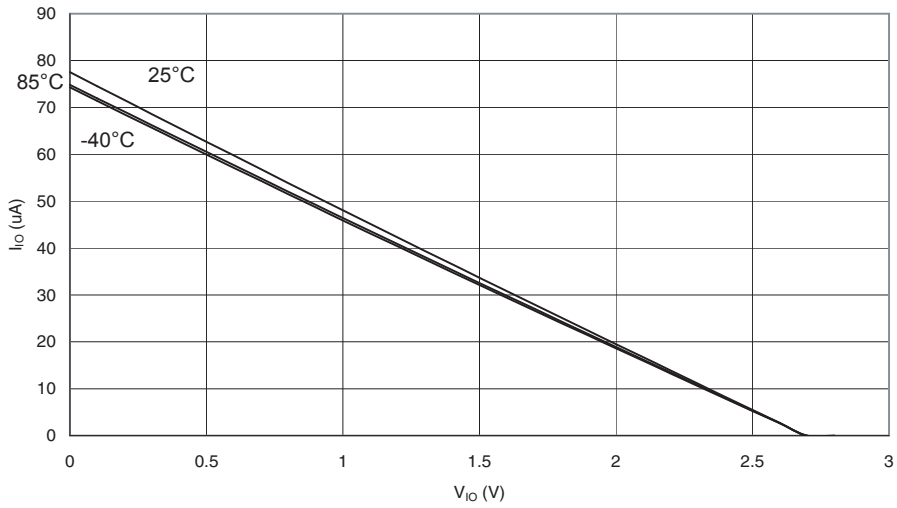


Figure 29-16. I/O Pin Pull-up Resistor Current vs. Input Voltage (V_{CC} = 2.7V)



Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page
0x25 (0x45)	Reserved	-	-	-	-	-	-	-	-	
0x24 (0x44)	TCCR0A	FOC0A	WGM00	COM0A1	COM0A0	WGM01	CS02	CS01	CS00	103
0x23 (0x43)	GTCCR	TSM	-	-	-	-	-	PSR2	PSR10	108/157
0x22 (0x42)	EEARH	-	-	-	-	-	EEPROM Address Register High			22
0x21 (0x41)	EEARL	EEPROM Address Register Low								22
0x20 (0x40)	EEDR	EEPROM Data Register								22
0x1F (0x3F)	EECR	-	-	-	-	EERIE	EEMWE	EEWE	EERE	22
0x1E (0x3E)	GPIOR0	General Purpose I/O Register								25
0x1D (0x3D)	EIMSK	**PCIE3**	**PCIE2**	PCIE1	PCIE0	-	-	-	INT0	55
0x1C (0x3C)	EIFR	**PCIF3**	**PCIF2**	PCIF1	PCIF0	-	-	-	INTF0	56
0x1B (0x3B)	Reserved	-	-	-	-	-	-	-	-	
0x1A (0x3A)	Reserved	-	-	-	-	-	-	-	-	
0x19 (0x39)	Reserved	-	-	-	-	-	-	-	-	
0x18 (0x38)	Reserved	-	-	-	-	-	-	-	-	
0x17 (0x37)	TIFR2	-	-	-	-	-	-	OCF2A	TOV2	157
0x16 (0x36)	TIFR1	-	-	ICF1	-	-	OCF1B	OCF1A	TOV1	138
0x15 (0x35)	TIFR0	-	-	-	-	-	-	OCF0A	TOV0	106
0x14 (0x34)	PORTG	-	-	-	PORTG4	PORTG3	PORTG2	PORTG1	PORTG0	89
0x13 (0x33)	DDRG	-	-	-	DDG4	DDG3	DDG2	DDG1	DDG0	89
0x12 (0x32)	PING	-	-	PING5	PING4	PING3	PING2	PING1	PING0	89
0x11 (0x31)	PORTF	PORTF7	PORTF6	PORTF5	PORTF4	PORTF3	PORTF2	PORTF1	PORTF0	89
0x10 (0x30)	DDRF	DDF7	DDF6	DDF5	DDF4	DDF3	DDF2	DDF1	DDF0	89
0x0F (0x2F)	PINF	PINF7	PINF6	PINF5	PINF4	PINF3	PINF2	PINF1	PINF0	89
0x0E (0x2E)	PORTE	PORTE7	PORTE6	PORTE5	PORTE4	PORTE3	PORTE2	PORTE1	PORTE0	88
0x0D (0x2D)	DDRE	DDE7	DDE6	DDE5	DDE4	DDE3	DDE2	DDE1	DDE0	88
0x0C (0x2C)	PINE	PINE7	PINE6	PINE5	PINE4	PINE3	PINE2	PINE1	PINE0	89
0x0B (0x2B)	PORTD	PORTD7	PORTD6	PORTD5	PORTD4	PORTD3	PORTD2	PORTD1	PORTD0	88
0x0A (0x2A)	DDRD	DDD7	DDD6	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0	88
0x09 (0x29)	PIND	PIND7	PIND6	PIND5	PIND4	PIND3	PIND2	PIND1	PIND0	88
0x08 (0x28)	PORTC	PORTC7	PORTC6	PORTC5	PORTC4	PORTC3	PORTC2	PORTC1	PORTC0	88
0x07 (0x27)	DDRC	DDC7	DDC6	DDC5	DDC4	DDC3	DDC2	DDC1	DDC0	88
0x06 (0x26)	PINC	PINC7	PINC6	PINC5	PINC4	PINC3	PINC2	PINC1	PINC0	88
0x05 (0x25)	PORTB	PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0	87
0x04 (0x24)	DDRB	DDB7	DDB6	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0	87
0x03 (0x23)	PINB	PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0	87
0x02 (0x22)	PORTA	PORTA7	PORTA6	PORTA5	PORTA4	PORTA3	PORTA2	PORTA1	PORTA0	87
0x01 (0x21)	DDRA	DDA7	DDA6	DDA5	DDA4	DDA3	DDA2	DDA1	DDA0	87
0x00 (0x20)	PINA	PINA7	PINA6	PINA5	PINA4	PINA3	PINA2	PINA1	PINA0	87

Note:
1. For compatibility with future devices, reserved bits should be written to zero if accessed. Reserved I/O memory addresses should never be written.
2. I/O Registers within the address range 0x00 - 0x1F are directly bit-accessible using the SBI and CBI instructions. In these registers, the value of single bits can be checked by using the SBIS and SBIC instructions.
3. Some of the Status Flags are cleared by writing a logical one to them. Note that, unlike most other AVRs, the CBI and SBI instructions will only operate on the specified bit, and can therefore be used on registers containing such Status Flags. The CBI and SBI instructions work with registers 0x00 to 0x1F only.
4. When using the I/O specific commands IN and OUT, the I/O addresses 0x00 - 0x3F must be used. When addressing I/O Registers as data space using LD and ST instructions, 0x20 must be added to these addresses. The ATmega329/3290/649/6490 is a complex microcontroller with more peripheral units than can be supported within the 64 location reserved in Opcode for the IN and OUT instructions. For the Extended I/O space from 0x60 - 0xFF in SRAM, only the ST/STS/STD and LD/LDS/LDD instructions can be used.