



Welcome to [E-XFL.COM](https://www.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.

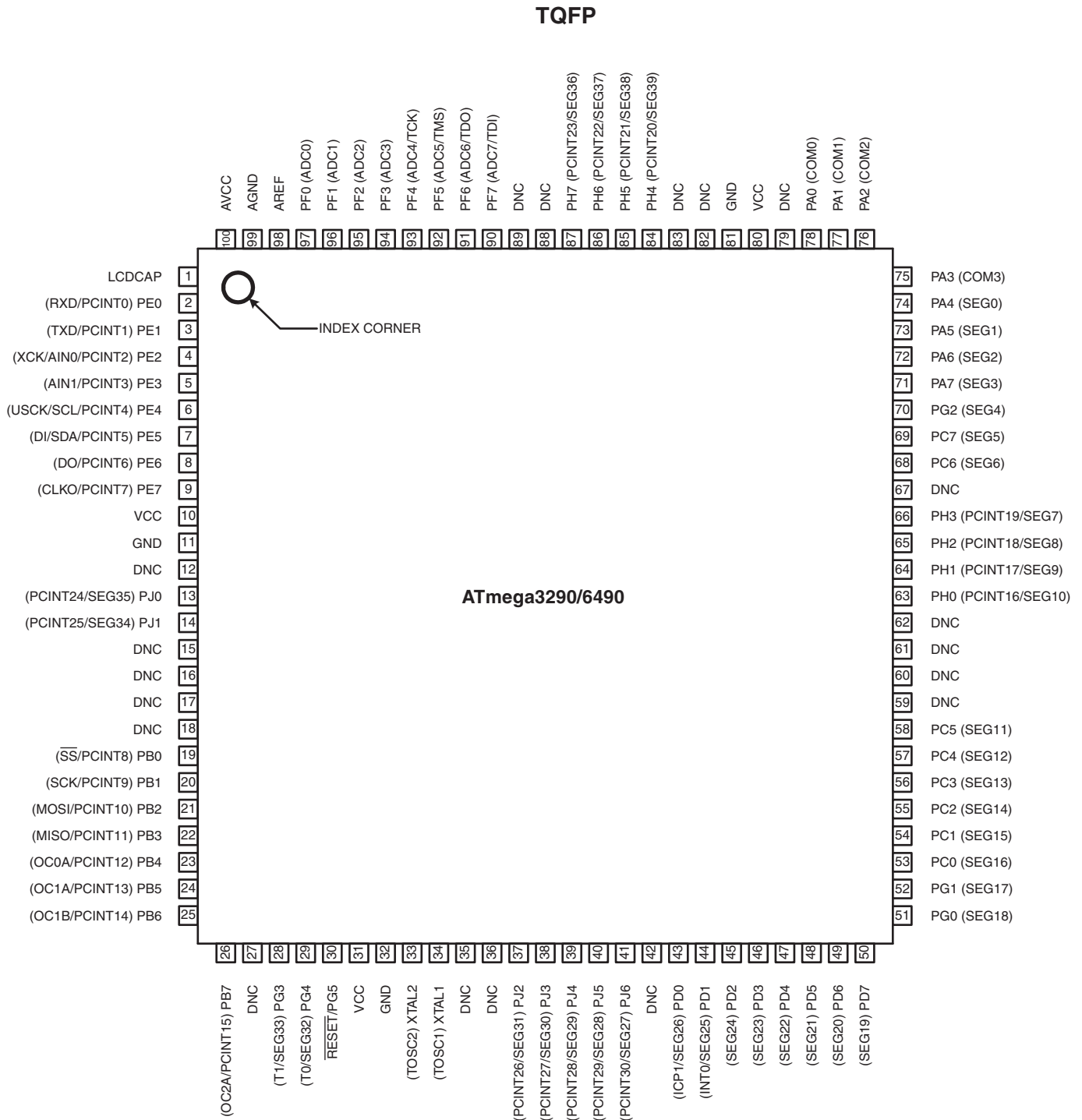
### Applications of "[Embedded - Microcontrollers](#)"

#### Details

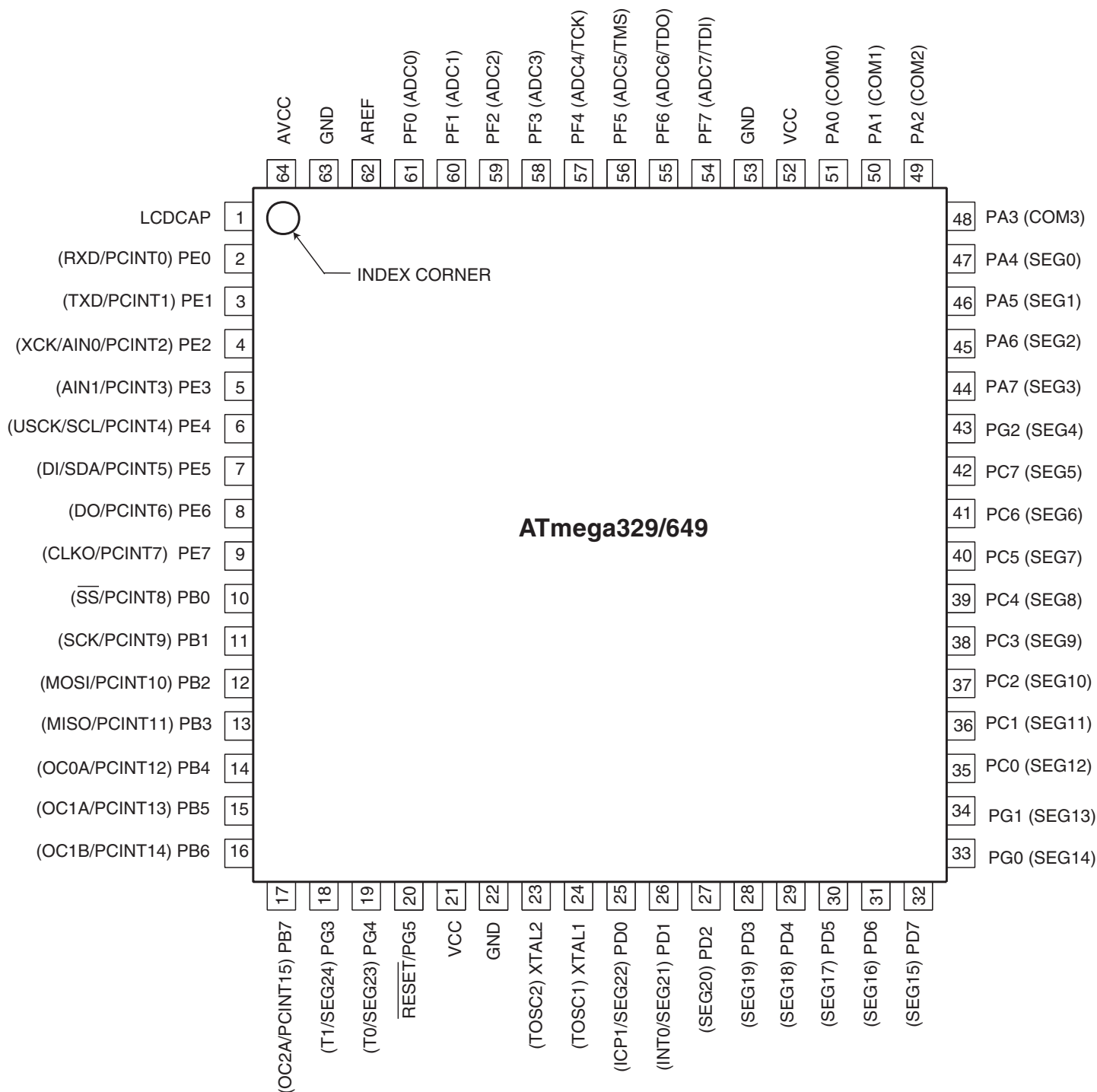
|                            |                                                                                                                                                                 |
|----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Product Status             | Active                                                                                                                                                          |
| Core Processor             | AVR                                                                                                                                                             |
| Core Size                  | 8-Bit                                                                                                                                                           |
| Speed                      | 16MHz                                                                                                                                                           |
| Connectivity               | SPI, UART/USART, USI                                                                                                                                            |
| Peripherals                | Brown-out Detect/Reset, LCD, 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-VFQFN Exposed Pad                                                                                                                                            |
| Supplier Device Package    | 64-QFN (9x9)                                                                                                                                                    |
| Purchase URL               | <a href="https://www.e-xfl.com/product-detail/microchip-technology/atmega649-16mu">https://www.e-xfl.com/product-detail/microchip-technology/atmega649-16mu</a> |

## 1. Pin Configurations

Figure 1-1. Pinout ATmega3290/6490



**Figure 1-2.** Pinout ATmega329/649



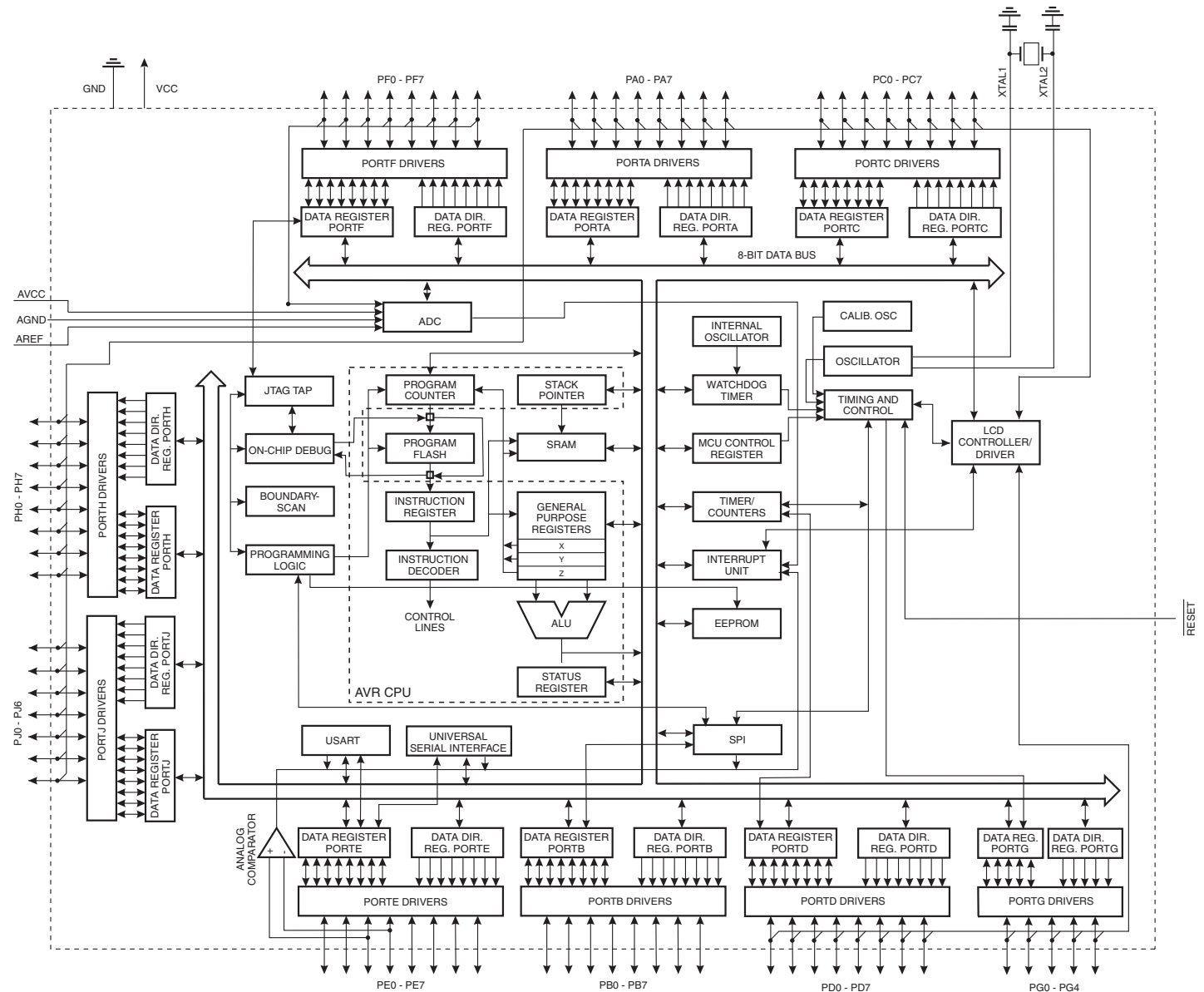
**Note:** The large center pad underneath the QFN/MLF packages is made of metal and internally connected to GND. It should be soldered or glued to the board to ensure good mechanical stability. If the center pad is left unconnected, the package might loosen from the board.

## 2. Overview

The ATmega329/3290/649/6490 is a low-power CMOS 8-bit microcontroller based on the AVR enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the ATmega329/3290/649/6490 achieves throughputs approaching 1 MIPS per MHz allowing the system designer to optimize power consumption versus processing speed.

### 2.1 Block Diagram

Figure 2-1. Block Diagram



## 2.2 Comparison between ATmega329, ATmega3290, ATmega649 and ATmega6490

The ATmega329, ATmega3290, ATmega649, and ATmega6490 differs only in memory sizes, pin count and pinout. [Table 2-1 on page 6](#) summarizes the different configurations for the four devices.

**Table 2-1.** Configuration Summary

| Device     | Flash    | EEPROM   | RAM     | LCD Segments | General Purpose I/O Pins |
|------------|----------|----------|---------|--------------|--------------------------|
| ATmega329  | 32Kbytes | 1Kbytes  | 2Kbytes | 4 x 25       | 54                       |
| ATmega3290 | 32Kbytes | 1K bytes | 2Kbytes | 4 x 40       | 69                       |
| ATmega649  | 64Kbytes | 2Kbytes  | 4Kbytes | 4 x 25       | 54                       |
| ATmega6490 | 64Kbytes | 2Kbytes  | 4Kbytes | 4 x 40       | 69                       |

## 2.3 Pin Descriptions

The following section describes the I/O-pin special functions.

### 2.3.1 $V_{CC}$

Digital supply voltage.

### 2.3.2 GND

Ground.

### 2.3.3 Port A (PA7..PA0)

Port A is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port A output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port A pins that are externally pulled low will source current if the pull-up resistors are activated. The Port A pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port A also serves the functions of various special features of the ATmega329/3290/649/6490 as listed on [page 67](#).

### 2.3.4 Port B (PB7..PB0)

Port B is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port B output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port B pins that are externally pulled low will source current if the pull-up resistors are activated. The Port B pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port B has better driving capabilities than the other ports.

Port B also serves the functions of various special features of the ATmega329/3290/649/6490 as listed on [page 68](#).

**2.3.9 Port G (PG5..PG0)**

Port G is a 6-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port G output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port G pins that are externally pulled low will source current if the pull-up resistors are activated. The Port G pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port G also serves the functions of various special features of the ATmega329/3290/649/6490 as listed on [page 75](#).

**2.3.10 Port H (PH7..PH0)**

Port H is a 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port H output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port H pins that are externally pulled low will source current if the pull-up resistors are activated. The Port H pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port H also serves the functions of various special features of the ATmega3290/6490 as listed on [page 75](#).

**2.3.11 Port J (PJ6..PJ0)**

Port J is a 7-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port J output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port J pins that are externally pulled low will source current if the pull-up resistors are activated. The Port J pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port J also serves the functions of various special features of the ATmega3290/6490 as listed on [page 75](#).

**2.3.12  $\overline{\text{RESET}}$** 

Reset input. A low level on this pin for longer than the minimum pulse length will generate a reset, even if the clock is not running. The minimum pulse length is given in “[System and Reset Characteristics](#)” on [page 330](#). Shorter pulses are not guaranteed to generate a reset.

**2.3.13 XTAL1**

Input to the inverting Oscillator amplifier and input to the internal clock operating circuit.

**2.3.14 XTAL2**

Output from the inverting Oscillator amplifier.

**2.3.15 AVCC**

AVCC is the supply voltage pin for Port F and the A/D Converter. It should be externally connected to  $V_{CC}$ , even if the ADC is not used. If the ADC is used, it should be connected to  $V_{CC}$  through a low-pass filter.

**2.3.16 AREF**

This is the analog reference pin for the A/D Converter.

## 6. Register Summary

Note: Registers with bold type only available in ATmega3290/6490.

| Address | Name           | Bit 7                          | Bit 6         | Bit 5         | Bit 4         | Bit 3          | Bit 2         | Bit 1         | Bit 0         | Page |
|---------|----------------|--------------------------------|---------------|---------------|---------------|----------------|---------------|---------------|---------------|------|
| (0xFF)  | <b>LCDDR19</b> | <b>SEG339</b>                  | <b>SEG338</b> | <b>SEG337</b> | <b>SEG336</b> | <b>SEG335</b>  | <b>SEG334</b> | <b>SEG333</b> | <b>SEG332</b> | 244  |
| (0xFE)  | LCDDR18        | <b>SEG331</b>                  | <b>SEG330</b> | <b>SEG329</b> | <b>SEG328</b> | <b>SEG327</b>  | <b>SEG326</b> | <b>SEG325</b> | SEG324        | 244  |
| (0xFD)  | LCDDR17        | SEG323                         | SEG322        | SEG321        | SEG320        | SEG319         | SEG318        | SEG317        | SEG316        | 244  |
| (0xFC)  | LCDDR16        | SEG315                         | SEG314        | SEG313        | SEG312        | SEG311         | SEG310        | SEG309        | SEG308        | 244  |
| (0xFB)  | LCDDR15        | SEG307                         | SEG306        | SEG305        | SEG304        | SEG303         | SEG302        | SEG301        | SEG300        | 244  |
| (0xFA)  | <b>LCDDR14</b> | <b>SEG239</b>                  | <b>SEG238</b> | <b>SEG237</b> | <b>SEG236</b> | <b>SEG235</b>  | <b>SEG234</b> | <b>SEG233</b> | <b>SEG232</b> | 244  |
| (0xF9)  | LCDDR13        | <b>SEG231</b>                  | <b>SEG230</b> | <b>SEG229</b> | <b>SEG228</b> | <b>SEG227</b>  | <b>SEG226</b> | <b>SEG225</b> | SEG224        | 244  |
| (0xF8)  | LCDDR12        | SEG223                         | SEG222        | SEG221        | SEG220        | SEG219         | SEG218        | SEG217        | SEG216        | 244  |
| (0xF7)  | LCDDR11        | SEG215                         | SEG214        | SEG213        | SEG212        | SEG211         | SEG210        | SEG209        | SEG208        | 244  |
| (0xF6)  | LCDDR10        | SEG207                         | SEG206        | SEG205        | SEG204        | SEG203         | SEG202        | SEG201        | SEG200        | 244  |
| (0xF5)  | <b>LCDDR09</b> | <b>SEG139</b>                  | <b>SEG138</b> | <b>SEG137</b> | <b>SEG136</b> | <b>SEG135</b>  | <b>SEG134</b> | <b>SEG133</b> | <b>SEG132</b> | 244  |
| (0xF4)  | LCDDR08        | <b>SEG131</b>                  | <b>SEG130</b> | <b>SEG129</b> | <b>SEG128</b> | <b>SEG127</b>  | <b>SEG126</b> | <b>SEG125</b> | SEG124        | 244  |
| (0xF3)  | LCDDR07        | SEG123                         | SEG122        | SEG121        | SEG120        | SEG119         | SEG118        | SEG117        | SEG116        | 244  |
| (0xF2)  | LCDDR06        | SEG115                         | SEG114        | SEG113        | SEG112        | SEG111         | SEG110        | SEG109        | SEG108        | 244  |
| (0xF1)  | LCDDR05        | SEG107                         | SEG106        | SEG105        | SEG104        | SEG103         | SEG102        | SEG101        | SEG100        | 244  |
| (0xF0)  | <b>LCDDR04</b> | <b>SEG039</b>                  | <b>SEG038</b> | <b>SEG037</b> | <b>SEG036</b> | <b>SEG035</b>  | <b>SEG034</b> | <b>SEG033</b> | <b>SEG032</b> | 244  |
| (0xEF)  | LCDDR03        | <b>SEG031</b>                  | <b>SEG030</b> | <b>SEG029</b> | <b>SEG028</b> | <b>SEG027</b>  | <b>SEG026</b> | <b>SEG025</b> | SEG024        | 244  |
| (0xEE)  | LCDDR02        | SEG023                         | SEG022        | SEG021        | SEG020        | SEG019         | SEG018        | SEG017        | SEG016        | 244  |
| (0xED)  | LCDDR01        | SEG015                         | SEG014        | SEG013        | SEG012        | SEG011         | SEG010        | SEG009        | SEG008        | 244  |
| (0xEC)  | LCDDR00        | SEG007                         | SEG006        | SEG005        | SEG004        | SEG003         | SEG002        | SEG001        | SEG000        | 244  |
| (0xEB)  | Reserved       | -                              | -             | -             | -             | -              | -             | -             | -             |      |
| (0xEA)  | Reserved       | -                              | -             | -             | -             | -              | -             | -             | -             |      |
| (0xE9)  | Reserved       | -                              | -             | -             | -             | -              | -             | -             | -             |      |
| (0xE8)  | Reserved       | -                              | -             | -             | -             | -              | -             | -             | -             |      |
| (0xE7)  | LCDDCR         | LCDDC2                         | LCDDC1        | LCDDC0        | -             | LCDDC3         | LCDDC2        | LCDDC1        | LCDDC0        | 243  |
| (0xE6)  | LCDFRR         | -                              | LCDPS2        | LCDPS1        | LCDPS0        | -              | LCDCD2        | LCDCD1        | LCDCD0        | 241  |
| (0xE5)  | LCDCRB         | LDCDS                          | LCD2B         | LCDMUX1       | LCDMUX0       | <b>LCDDPM3</b> | LCDDPM2       | LCDDPM1       | LCDDPM0       | 239  |
| (0xE4)  | LCDCRA         | LCDEN                          | LCDAB         | -             | LCDIF         | LCDIE          | -             | -             | LCDDBL        | 239  |
| (0xE3)  | Reserved       | -                              | -             | -             | -             | -              | -             | -             | -             |      |
| (0xE2)  | Reserved       | -                              | -             | -             | -             | -              | -             | -             | -             |      |
| (0xE1)  | Reserved       | -                              | -             | -             | -             | -              | -             | -             | -             |      |
| (0xE0)  | Reserved       | -                              | -             | -             | -             | -              | -             | -             | -             |      |
| (0xDF)  | Reserved       | -                              | -             | -             | -             | -              | -             | -             | -             |      |
| (0xDE)  | Reserved       | -                              | -             | -             | -             | -              | -             | -             | -             |      |
| (0xDD)  | <b>PORTJ</b>   | -                              | <b>PORTJ6</b> | <b>PORTJ5</b> | <b>PORTJ4</b> | <b>PORTJ3</b>  | <b>PORTJ2</b> | <b>PORTJ1</b> | <b>PORTJ0</b> | 90   |
| (0xDC)  | <b>DDRJ</b>    | -                              | <b>DDJ6</b>   | <b>DDJ5</b>   | <b>DDJ4</b>   | <b>DDJ3</b>    | <b>DDJ2</b>   | <b>DDJ1</b>   | <b>DDJ0</b>   | 90   |
| (0xDB)  | <b>PINJ</b>    | -                              | <b>PINJ6</b>  | <b>PINJ5</b>  | <b>PINJ4</b>  | <b>PINJ3</b>   | <b>PINJ2</b>  | <b>PINJ1</b>  | <b>PINJ0</b>  | 90   |
| (0xDA)  | <b>PORTH</b>   | <b>PORTH7</b>                  | <b>PORTH6</b> | <b>PORTH5</b> | <b>PORTH4</b> | <b>PORTH3</b>  | <b>PORTH2</b> | <b>PORTH1</b> | <b>PORTH0</b> | 89   |
| (0xD9)  | <b>DDRH</b>    | <b>DDH7</b>                    | <b>DDH6</b>   | <b>DDH5</b>   | <b>DDH4</b>   | <b>DDH3</b>    | <b>DDH2</b>   | <b>DDH1</b>   | <b>DDH0</b>   | 90   |
| (0xD8)  | <b>PINH</b>    | <b>PINH7</b>                   | <b>PINH6</b>  | <b>PINH5</b>  | <b>PINH4</b>  | <b>PINH3</b>   | <b>PINH2</b>  | <b>PINH1</b>  | <b>PINH0</b>  | 90   |
| (0xD7)  | Reserved       | -                              | -             | -             | -             | -              | -             | -             | -             |      |
| (0xD6)  | Reserved       | -                              | -             | -             | -             | -              | -             | -             | -             |      |
| (0xD5)  | Reserved       | -                              | -             | -             | -             | -              | -             | -             | -             |      |
| (0xD4)  | Reserved       | -                              | -             | -             | -             | -              | -             | -             | -             |      |
| (0xD3)  | Reserved       | -                              | -             | -             | -             | -              | -             | -             | -             |      |
| (0xD2)  | Reserved       | -                              | -             | -             | -             | -              | -             | -             | -             |      |
| (0xD1)  | Reserved       | -                              | -             | -             | -             | -              | -             | -             | -             |      |
| (0xD0)  | Reserved       | -                              | -             | -             | -             | -              | -             | -             | -             |      |
| (0xCF)  | Reserved       | -                              | -             | -             | -             | -              | -             | -             | -             |      |
| (0xCE)  | Reserved       | -                              | -             | -             | -             | -              | -             | -             | -             |      |
| (0xCD)  | Reserved       | -                              | -             | -             | -             | -              | -             | -             | -             |      |
| (0xCC)  | Reserved       | -                              | -             | -             | -             | -              | -             | -             | -             |      |
| (0xCB)  | Reserved       | -                              | -             | -             | -             | -              | -             | -             | -             |      |
| (0xCA)  | Reserved       | -                              | -             | -             | -             | -              | -             | -             | -             |      |
| (0xC9)  | Reserved       | -                              | -             | -             | -             | -              | -             | -             | -             |      |
| (0xC8)  | Reserved       | -                              | -             | -             | -             | -              | -             | -             | -             |      |
| (0xC7)  | Reserved       | -                              | -             | -             | -             | -              | -             | -             | -             |      |
| (0xC6)  | UDR0           | USART0 Data Register           |               |               |               |                |               |               |               | 190  |
| (0xC5)  | UBRR0H         | USART0 Baud Rate Register High |               |               |               |                |               |               |               | 194  |
| (0xC4)  | UBRR0L         | USART0 Baud Rate Register Low  |               |               |               |                |               |               |               | 194  |

| Address | Name     | Bit 7                                         | Bit 6  | Bit 5  | Bit 4  | Bit 3   | Bit 2   | Bit 1   | Bit 0   | Page |
|---------|----------|-----------------------------------------------|--------|--------|--------|---------|---------|---------|---------|------|
| (0xC3)  | Reserved | -                                             | -      | -      | -      | -       | -       | -       | -       |      |
| (0xC2)  | UCSR0C   | -                                             | UMSEL0 | UPM01  | UPM00  | USBS0   | UCSZ01  | UCSZ00  | UCPOL0  | 192  |
| (0xC1)  | UCSR0B   | RXCIE0                                        | TXCIE0 | UDRIE0 | RXEN0  | TXEN0   | UCSZ02  | RXB80   | TXB80   | 191  |
| (0xC0)  | UCSR0A   | RXC0                                          | TXC0   | UDRE0  | FE0    | DOR0    | UPE0    | U2X0    | MPCM0   | 190  |
| (0xBF)  | Reserved | -                                             | -      | -      | -      | -       | -       | -       | -       |      |
| (0xBE)  | Reserved | -                                             | -      | -      | -      | -       | -       | -       | -       |      |
| (0xBD)  | Reserved | -                                             | -      | -      | -      | -       | -       | -       | -       |      |
| (0xBC)  | Reserved | -                                             | -      | -      | -      | -       | -       | -       | -       |      |
| (0xBB)  | Reserved | -                                             | -      | -      | -      | -       | -       | -       | -       |      |
| (0xBA)  | USIDR    | USI Data Register                             |        |        |        |         |         |         |         | 203  |
| (0xB9)  | USISR    | USISIF                                        | USIOIF | USIPF  | USIDC  | USICNT3 | USICNT2 | USICNT1 | USICNT0 | 203  |
| (0xB8)  | USICR    | USISIE                                        | USIOIE | USIWM1 | USIWM0 | USICS1  | USICS0  | USICK   | USITC   | 204  |
| (0xB7)  | Reserved | -                                             | -      | -      | -      | -       | -       | -       | -       |      |
| (0xB6)  | ASSR     | -                                             | -      | -      | EXCLK  | AS2     | TCN2UB  | OCR2UB  | TCR2UB  | 155  |
| (0xB5)  | Reserved | -                                             | -      | -      | -      | -       | -       | -       | -       |      |
| (0xB4)  | Reserved | -                                             | -      | -      | -      | -       | -       | -       | -       |      |
| (0xB3)  | OCR2A    | Timer/Counter 2 Output Compare Register A     |        |        |        |         |         |         |         | 155  |
| (0xB2)  | TCNT2    | Timer/Counter2                                |        |        |        |         |         |         |         | 155  |
| (0xB1)  | Reserved | -                                             | -      | -      | -      | -       | -       | -       | -       |      |
| (0xB0)  | TCCR2A   | FOC2A                                         | WGM20  | COM2A1 | COM2A0 | WGM21   | CS22    | CS21    | CS20    | 153  |
| (0xAF)  | Reserved | -                                             | -      | -      | -      | -       | -       | -       | -       |      |
| (0xAE)  | Reserved | -                                             | -      | -      | -      | -       | -       | -       | -       |      |
| (0xAD)  | Reserved | -                                             | -      | -      | -      | -       | -       | -       | -       |      |
| (0xAC)  | Reserved | -                                             | -      | -      | -      | -       | -       | -       | -       |      |
| (0xAB)  | Reserved | -                                             | -      | -      | -      | -       | -       | -       | -       |      |
| (0xAA)  | Reserved | -                                             | -      | -      | -      | -       | -       | -       | -       |      |
| (0xA9)  | Reserved | -                                             | -      | -      | -      | -       | -       | -       | -       |      |
| (0xA8)  | Reserved | -                                             | -      | -      | -      | -       | -       | -       | -       |      |
| (0xA7)  | Reserved | -                                             | -      | -      | -      | -       | -       | -       | -       |      |
| (0xA6)  | Reserved | -                                             | -      | -      | -      | -       | -       | -       | -       |      |
| (0xA5)  | Reserved | -                                             | -      | -      | -      | -       | -       | -       | -       |      |
| (0xA4)  | Reserved | -                                             | -      | -      | -      | -       | -       | -       | -       |      |
| (0xA3)  | Reserved | -                                             | -      | -      | -      | -       | -       | -       | -       |      |
| (0xA2)  | Reserved | -                                             | -      | -      | -      | -       | -       | -       | -       |      |
| (0xA1)  | Reserved | -                                             | -      | -      | -      | -       | -       | -       | -       |      |
| (0xA0)  | Reserved | -                                             | -      | -      | -      | -       | -       | -       | -       |      |
| (0x9F)  | Reserved | -                                             | -      | -      | -      | -       | -       | -       | -       |      |
| (0x9E)  | Reserved | -                                             | -      | -      | -      | -       | -       | -       | -       |      |
| (0x9D)  | Reserved | -                                             | -      | -      | -      | -       | -       | -       | -       |      |
| (0x9C)  | Reserved | -                                             | -      | -      | -      | -       | -       | -       | -       |      |
| (0x9B)  | Reserved | -                                             | -      | -      | -      | -       | -       | -       | -       |      |
| (0x9A)  | Reserved | -                                             | -      | -      | -      | -       | -       | -       | -       |      |
| (0x99)  | Reserved | -                                             | -      | -      | -      | -       | -       | -       | -       |      |
| (0x98)  | Reserved | -                                             | -      | -      | -      | -       | -       | -       | -       |      |
| (0x97)  | Reserved | -                                             | -      | -      | -      | -       | -       | -       | -       |      |
| (0x96)  | Reserved | -                                             | -      | -      | -      | -       | -       | -       | -       |      |
| (0x95)  | Reserved | -                                             | -      | -      | -      | -       | -       | -       | -       |      |
| (0x94)  | Reserved | -                                             | -      | -      | -      | -       | -       | -       | -       |      |
| (0x93)  | Reserved | -                                             | -      | -      | -      | -       | -       | -       | -       |      |
| (0x92)  | Reserved | -                                             | -      | -      | -      | -       | -       | -       | -       |      |
| (0x91)  | Reserved | -                                             | -      | -      | -      | -       | -       | -       | -       |      |
| (0x90)  | Reserved | -                                             | -      | -      | -      | -       | -       | -       | -       |      |
| (0x8F)  | Reserved | -                                             | -      | -      | -      | -       | -       | -       | -       |      |
| (0x8E)  | Reserved | -                                             | -      | -      | -      | -       | -       | -       | -       |      |
| (0x8D)  | Reserved | -                                             | -      | -      | -      | -       | -       | -       | -       |      |
| (0x8C)  | Reserved | -                                             | -      | -      | -      | -       | -       | -       | -       |      |
| (0x8B)  | OCR1BH   | Timer/Counter1 Output Compare Register B High |        |        |        |         |         |         |         | 136  |
| (0x8A)  | OCR1BL   | Timer/Counter1 Output Compare Register B Low  |        |        |        |         |         |         |         | 136  |
| (0x89)  | OCR1AH   | Timer/Counter1 Output Compare Register A High |        |        |        |         |         |         |         | 136  |
| (0x88)  | OCR1AL   | Timer/Counter1 Output Compare Register A Low  |        |        |        |         |         |         |         | 136  |
| (0x87)  | ICR1H    | Timer/Counter1 Input Capture Register High    |        |        |        |         |         |         |         | 137  |
| (0x86)  | ICR1L    | Timer/Counter1 Input Capture Register Low     |        |        |        |         |         |         |         | 137  |
| (0x85)  | TCNT1H   | Timer/Counter1 High                           |        |        |        |         |         |         |         | 136  |

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

## 7. Instruction Set Summary

| Mnemonics                                | Operands | Description                              | Operation                                             | Flags      | #Clocks |
|------------------------------------------|----------|------------------------------------------|-------------------------------------------------------|------------|---------|
| <b>ARITHMETIC AND LOGIC INSTRUCTIONS</b> |          |                                          |                                                       |            |         |
| ADD                                      | Rd, Rr   | Add two Registers                        | $Rd \leftarrow Rd + Rr$                               | Z,C,N,V,H  | 1       |
| ADC                                      | Rd, Rr   | Add with Carry two Registers             | $Rd \leftarrow Rd + Rr + C$                           | Z,C,N,V,H  | 1       |
| ADIW                                     | RdI,K    | Add Immediate to Word                    | $Rdh:Rdl \leftarrow Rdh:Rdl + K$                      | Z,C,N,V,S  | 2       |
| SUB                                      | Rd, Rr   | Subtract two Registers                   | $Rd \leftarrow Rd - Rr$                               | Z,C,N,V,H  | 1       |
| SUBI                                     | Rd, K    | Subtract Constant from Register          | $Rd \leftarrow Rd - K$                                | Z,C,N,V,H  | 1       |
| SBC                                      | Rd, Rr   | Subtract with Carry two Registers        | $Rd \leftarrow Rd - Rr - C$                           | Z,C,N,V,H  | 1       |
| SBCI                                     | Rd, K    | Subtract with Carry Constant from Reg.   | $Rd \leftarrow Rd - K - C$                            | Z,C,N,V,H  | 1       |
| SBIW                                     | RdI,K    | Subtract Immediate from Word             | $Rdh:Rdl \leftarrow Rdh:Rdl - K$                      | Z,C,N,V,S  | 2       |
| AND                                      | Rd, Rr   | Logical AND Registers                    | $Rd \leftarrow Rd \bullet Rr$                         | Z,N,V      | 1       |
| ANDI                                     | Rd, K    | Logical AND Register and Constant        | $Rd \leftarrow Rd \bullet K$                          | Z,N,V      | 1       |
| OR                                       | Rd, Rr   | Logical OR Registers                     | $Rd \leftarrow Rd \vee Rr$                            | Z,N,V      | 1       |
| ORI                                      | Rd, K    | Logical OR Register and Constant         | $Rd \leftarrow Rd \vee K$                             | Z,N,V      | 1       |
| EOR                                      | Rd, Rr   | Exclusive OR Registers                   | $Rd \leftarrow Rd \oplus Rr$                          | Z,N,V      | 1       |
| COM                                      | Rd       | One's Complement                         | $Rd \leftarrow 0xFF - Rd$                             | Z,C,N,V    | 1       |
| NEG                                      | Rd       | Two's Complement                         | $Rd \leftarrow 0x00 - Rd$                             | Z,C,N,V,H  | 1       |
| SBR                                      | Rd,K     | Set Bit(s) in Register                   | $Rd \leftarrow Rd \vee K$                             | Z,N,V      | 1       |
| CBR                                      | Rd,K     | Clear Bit(s) in Register                 | $Rd \leftarrow Rd \bullet (0xFF - K)$                 | Z,N,V      | 1       |
| INC                                      | Rd       | Increment                                | $Rd \leftarrow Rd + 1$                                | Z,N,V      | 1       |
| DEC                                      | Rd       | Decrement                                | $Rd \leftarrow Rd - 1$                                | Z,N,V      | 1       |
| TST                                      | Rd       | Test for Zero or Minus                   | $Rd \leftarrow Rd \bullet Rd$                         | Z,N,V      | 1       |
| CLR                                      | Rd       | Clear Register                           | $Rd \leftarrow Rd \oplus Rd$                          | Z,N,V      | 1       |
| SER                                      | Rd       | Set Register                             | $Rd \leftarrow 0xFF$                                  | None       | 1       |
| MUL                                      | Rd, Rr   | Multiply Unsigned                        | $R1:R0 \leftarrow Rd \times Rr$                       | Z,C        | 2       |
| MULS                                     | Rd, Rr   | Multiply Signed                          | $R1:R0 \leftarrow Rd \times Rr$                       | Z,C        | 2       |
| MULSU                                    | Rd, Rr   | Multiply Signed with Unsigned            | $R1:R0 \leftarrow Rd \times Rr$                       | Z,C        | 2       |
| FMUL                                     | Rd, Rr   | Fractional Multiply Unsigned             | $R1:R0 \leftarrow (Rd \times Rr) \ll 1$               | Z,C        | 2       |
| FMULS                                    | Rd, Rr   | Fractional Multiply Signed               | $R1:R0 \leftarrow (Rd \times Rr) \ll 1$               | Z,C        | 2       |
| FMULSU                                   | Rd, Rr   | Fractional Multiply Signed with Unsigned | $R1:R0 \leftarrow (Rd \times Rr) \ll 1$               | Z,C        | 2       |
| <b>BRANCH INSTRUCTIONS</b>               |          |                                          |                                                       |            |         |
| RJMP                                     | k        | Relative Jump                            | $PC \leftarrow PC + k + 1$                            | None       | 2       |
| JMP                                      |          | Indirect Jump to (Z)                     | $PC \leftarrow Z$                                     | None       | 2       |
| JMP                                      | k        | Direct Jump                              | $PC \leftarrow k$                                     | None       | 3       |
| RCALL                                    | k        | Relative Subroutine Call                 | $PC \leftarrow PC + k + 1$                            | None       | 3       |
| ICALL                                    |          | Indirect Call to (Z)                     | $PC \leftarrow Z$                                     | None       | 3       |
| CALL                                     | k        | Direct Subroutine Call                   | $PC \leftarrow k$                                     | None       | 4       |
| RET                                      |          | Subroutine Return                        | $PC \leftarrow STACK$                                 | None       | 4       |
| RETI                                     |          | Interrupt Return                         | $PC \leftarrow STACK$                                 | I          | 4       |
| CPSE                                     | Rd,Rr    | Compare, Skip if Equal                   | if $(Rd = Rr)$ $PC \leftarrow PC + 2$ or 3            | None       | 1/2/3   |
| CP                                       | Rd,Rr    | Compare                                  | $Rd - Rr$                                             | Z, N,V,C,H | 1       |
| CPC                                      | Rd,Rr    | Compare with Carry                       | $Rd - Rr - C$                                         | Z, N,V,C,H | 1       |
| CPI                                      | Rd,K     | Compare Register with Immediate          | $Rd - K$                                              | Z, N,V,C,H | 1       |
| SBRC                                     | Rr, b    | Skip if Bit in Register Cleared          | if $(Rr(b)=0)$ $PC \leftarrow PC + 2$ or 3            | None       | 1/2/3   |
| SBRB                                     | Rr, b    | Skip if Bit in Register is Set           | if $(Rr(b)=1)$ $PC \leftarrow PC + 2$ or 3            | None       | 1/2/3   |
| SBIC                                     | P, b     | Skip if Bit in I/O Register Cleared      | if $(P(b)=0)$ $PC \leftarrow PC + 2$ or 3             | None       | 1/2/3   |
| SBIS                                     | P, b     | Skip if Bit in I/O Register is Set       | if $(P(b)=1)$ $PC \leftarrow PC + 2$ or 3             | None       | 1/2/3   |
| BRBS                                     | s, k     | Branch if Status Flag Set                | if $(SREG(s) = 1)$ then $PC \leftarrow PC + k + 1$    | None       | 1/2     |
| BRBC                                     | s, k     | Branch if Status Flag Cleared            | if $(SREG(s) = 0)$ then $PC \leftarrow PC + k + 1$    | None       | 1/2     |
| BREQ                                     | k        | Branch if Equal                          | if $(Z = 1)$ then $PC \leftarrow PC + k + 1$          | None       | 1/2     |
| BRNE                                     | k        | Branch if Not Equal                      | if $(Z = 0)$ then $PC \leftarrow PC + k + 1$          | None       | 1/2     |
| BRCS                                     | k        | Branch if Carry Set                      | if $(C = 1)$ then $PC \leftarrow PC + k + 1$          | None       | 1/2     |
| BRCC                                     | k        | Branch if Carry Cleared                  | if $(C = 0)$ then $PC \leftarrow PC + k + 1$          | None       | 1/2     |
| BRSH                                     | k        | Branch if Same or Higher                 | if $(C = 0)$ then $PC \leftarrow PC + k + 1$          | None       | 1/2     |
| BRLO                                     | k        | Branch if Lower                          | if $(C = 1)$ then $PC \leftarrow PC + k + 1$          | None       | 1/2     |
| BRMI                                     | k        | Branch if Minus                          | if $(N = 1)$ then $PC \leftarrow PC + k + 1$          | None       | 1/2     |
| BRPL                                     | k        | Branch if Plus                           | if $(N = 0)$ then $PC \leftarrow PC + k + 1$          | None       | 1/2     |
| BRGE                                     | k        | Branch if Greater or Equal, Signed       | if $(N \oplus V = 0)$ then $PC \leftarrow PC + k + 1$ | None       | 1/2     |
| BRLT                                     | k        | Branch if Less Than Zero, Signed         | if $(N \oplus V = 1)$ then $PC \leftarrow PC + k + 1$ | None       | 1/2     |
| BRHS                                     | k        | Branch if Half Carry Flag Set            | if $(H = 1)$ then $PC \leftarrow PC + k + 1$          | None       | 1/2     |
| BRHC                                     | k        | Branch if Half Carry Flag Cleared        | if $(H = 0)$ then $PC \leftarrow PC + k + 1$          | None       | 1/2     |
| BRTS                                     | k        | Branch if T Flag Set                     | if $(T = 1)$ then $PC \leftarrow PC + k + 1$          | None       | 1/2     |

| 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     |
| <b>BIT AND BIT-TEST INSTRUCTIONS</b> |          |                                    |                                          |         |         |
| 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       |
| <b>DATA TRANSFER INSTRUCTIONS</b>    |          |                                    |                                          |         |         |
| 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    | -       |

| Mnemonics                       | Operands | Description             | Operation                                | Flags | #Clocks |
|---------------------------------|----------|-------------------------|------------------------------------------|-------|---------|
| IN                              | Rd, P    | In Port                 | $Rd \leftarrow P$                        | None  | 1       |
| OUT                             | P, Rr    | Out Port                | $P \leftarrow Rr$                        | None  | 1       |
| PUSH                            | Rr       | Push Register on Stack  | $STACK \leftarrow Rr$                    | None  | 2       |
| POP                             | Rd       | Pop Register from Stack | $Rd \leftarrow STACK$                    | None  | 2       |
| <b>MCU CONTROL INSTRUCTIONS</b> |          |                         |                                          |       |         |
| NOP                             |          | No Operation            |                                          | None  | 1       |
| SLEEP                           |          | Sleep                   | (see specific descr. for Sleep function) | None  | 1       |
| WDR                             |          | Watchdog Reset          | (see specific descr. for WDR/timer)      | None  | 1       |
| BREAK                           |          | Break                   | For On-chip Debug Only                   | None  | N/A     |

## 8. Ordering Information

### 8.1 ATmega329

| Speed (MHz) <sup>(3)</sup> | Power Supply | Ordering Code <sup>(2)</sup>                                                                         | Package Type <sup>(1)</sup> | Operational Range             |
|----------------------------|--------------|------------------------------------------------------------------------------------------------------|-----------------------------|-------------------------------|
| 8                          | 1.8 - 5.5V   | ATmega329V-8AU<br>ATmega329V-8AUR <sup>(4)</sup><br>ATmega329V-8MU<br>ATmega329V-8MUR <sup>(4)</sup> | 64A<br>64A<br>64M1<br>64M1  | Industrial<br>(-40°C to 85°C) |
| 16                         | 2.7 - 5.5V   | ATmega329-16AU<br>ATmega329-16AUR <sup>(4)</sup><br>ATmega329-16MU<br>ATmega329-16MUR <sup>(4)</sup> | 64A<br>64A<br>64M1<br>64M1  | Industrial<br>(-40°C to 85°C) |

- 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 vs.  $V_{CC}$  see [Figure 28-1 on page 328](#) and [Figure 28-2 on page 328](#).
  4. Tape & Reel

| Package Type |                                                                                              |
|--------------|----------------------------------------------------------------------------------------------|
| <b>64A</b>   | 64-lead, 14 x 14 x 1.0 mm, Thin Profile Plastic Quad Flat Package (TQFP)                     |
| <b>64M1</b>  | 64-pad, 9 x 9 x 1.0 mm, Quad Flat No-Lead/Micro Lead Frame Package (QFN/MLF)                 |
| <b>100A</b>  | 100-lead, 14 x 14 x 1.0 mm, 0.5 mm Lead Pitch, Thin Profile Plastic Quad Flat Package (TQFP) |

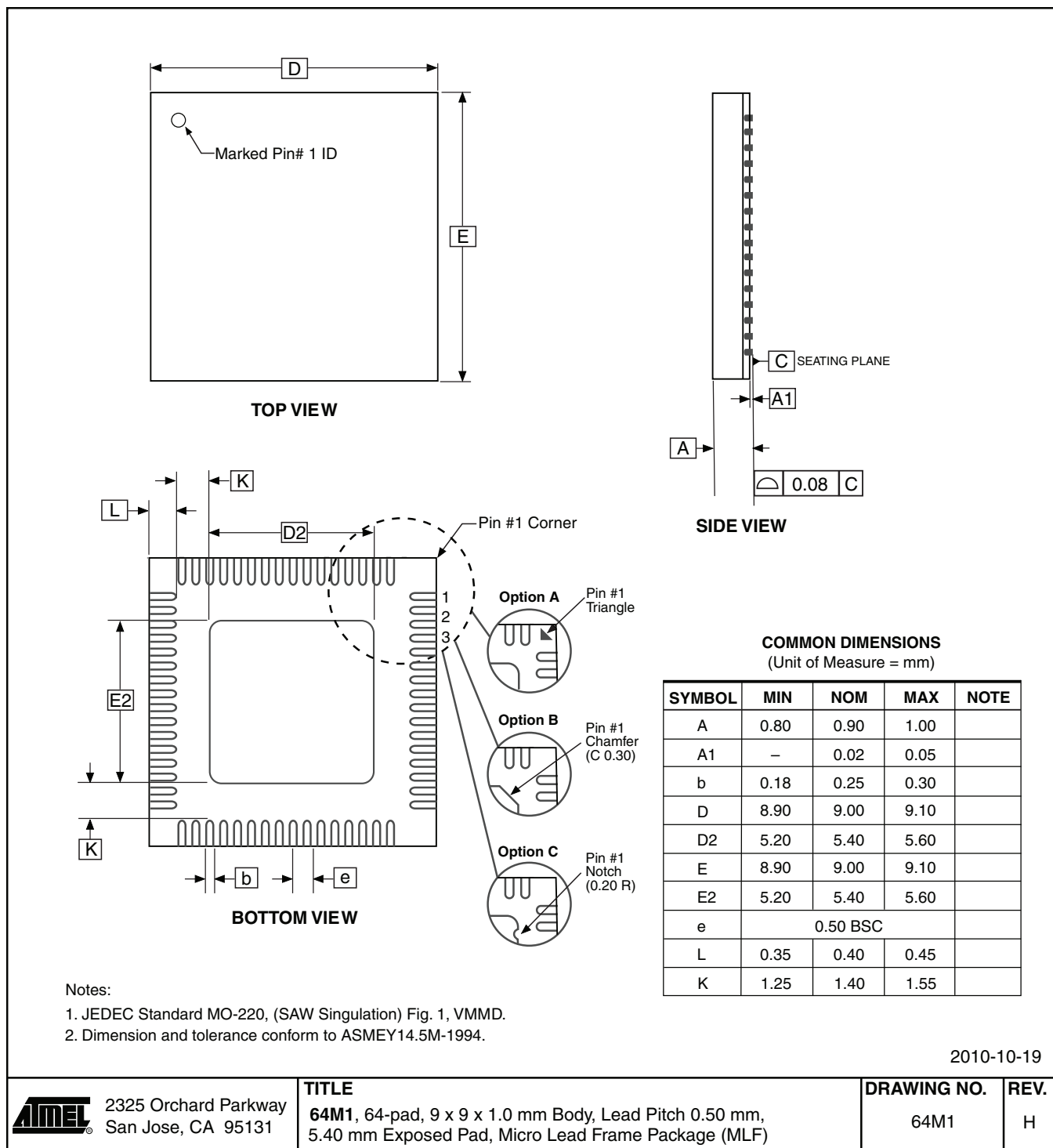
## 8.2 ATmega3290

| Speed (MHz) <sup>(3)</sup> | Power Supply | Ordering Code <sup>(2)</sup>                       | Package Type <sup>(1)</sup> | Operational Range             |
|----------------------------|--------------|----------------------------------------------------|-----------------------------|-------------------------------|
| 8                          | 1.8 - 5.5V   | ATmega3290V-8AU<br>ATmega3290V-8AUR <sup>(4)</sup> | 100A<br>100A                | Industrial<br>(-40°C to 85°C) |
| 16                         | 2.7 - 5.5V   | ATmega3290-16AU<br>ATmega3290-16AUR <sup>(4)</sup> | 100A<br>100A                | Industrial<br>(-40°C to 85°C) |

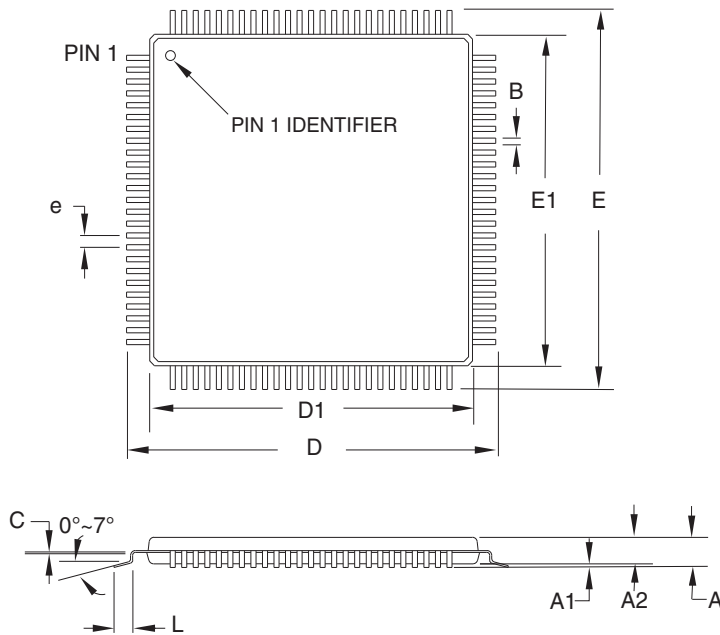
- Notes:
1. This device can also be supplied in wafer form. Please contact your local Atmel sales office for detailed ordering information and minimum quantities.
  2. Pb-free packaging alternative, complies to the European Directive for Restriction of Hazardous Substances (RoHS directive). Also Halide free and fully Green.
  3. For Speed vs.  $V_{CC}$  see [Figure 28-1 on page 328](#) and [Figure 28-2 on page 328](#).
  4. Tape & Reel

| Package Type |                                                                                              |
|--------------|----------------------------------------------------------------------------------------------|
| <b>64A</b>   | 64-lead, 14 x 14 x 1.0 mm, Thin Profile Plastic Quad Flat Package (TQFP)                     |
| <b>64M1</b>  | 64-pad, 9 x 9 x 1.0 mm, Quad Flat No-Lead/Micro Lead Frame Package (QFN/MLF)                 |
| <b>100A</b>  | 100-lead, 14 x 14 x 1.0 mm, 0.5 mm Lead Pitch, Thin Profile Plastic Quad Flat Package (TQFP) |

## 9.2 64M1



## 9.3 100A



**COMMON DIMENSIONS**  
(Unit of Measure = mm)

| SYMBOL | MIN      | NOM   | MAX   | NOTE   |
|--------|----------|-------|-------|--------|
| A      | —        | —     | 1.20  |        |
| A1     | 0.05     | —     | 0.15  |        |
| A2     | 0.95     | 1.00  | 1.05  |        |
| D      | 15.75    | 16.00 | 16.25 |        |
| D1     | 13.90    | 14.00 | 14.10 | Note 2 |
| E      | 15.75    | 16.00 | 16.25 |        |
| E1     | 13.90    | 14.00 | 14.10 | Note 2 |
| B      | 0.17     | —     | 0.27  |        |
| C      | 0.09     | —     | 0.20  |        |
| L      | 0.45     | —     | 0.75  |        |
| e      | 0.50 TYP |       |       |        |

**Notes:**

1. This package conforms to JEDEC reference MS-026, Variation AED.
2. Dimensions D1 and E1 do not include mold protrusion. Allowable protrusion is 0.25 mm per side. Dimensions D1 and E1 are maximum plastic body size dimensions including mold mismatch.
3. Lead coplanarity is 0.08 mm maximum.

2010-10-20



2325 Orchard Parkway  
San Jose, CA 95131

**TITLE**

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

**DRAWING NO.**

100A

**REV.**

D

## 10. Errata

### 10.1 ATmega329

#### 10.1.1 ATmega329 rev. C

- Interrupts may be lost when writing the timer registers in the asynchronous timer

##### 1. Interrupts may be lost when writing the timer registers in the asynchronous timer

The interrupt will be lost if a timer register that is synchronous timer clock is written when the asynchronous Timer/Counter register (TCNTx) is 0x00.

##### Problem Fix/Workaround

Always check that the asynchronous Timer/Counter register neither have the value 0xFF nor 0x00 before writing to the asynchronous Timer Control Register (TCCR<sub>x</sub>), asynchronous Timer Counter Register (TCNT<sub>x</sub>), or asynchronous Output Compare Register (OCR<sub>x</sub>).

#### 10.1.2 ATmega329 rev. B

Not sampled.

#### 10.1.3 ATmega329 rev. A

- LCD contrast voltage too high
- Interrupts may be lost when writing the timer registers in the asynchronous timer

##### 1. LCD contrast voltage too high

When the LCD is active and using low power waveform, the LCD contrast voltage can be too high. This occurs when  $V_{CC}$  is higher than  $V_{LCD}$ , and when using low LCD drivetime.

##### Problem Fix/Workaround

There are several possible workarounds:

- Use normal waveform instead of low power waveform
- Use drivetime of 375  $\mu$ s or longer

##### 2. Interrupts may be lost when writing the timer registers in the asynchronous timer

The interrupt will be lost if a timer register that is synchronous timer clock is written when the asynchronous Timer/Counter register (TCNT<sub>x</sub>) is 0x00.

##### Problem Fix/Workaround

Always check that the asynchronous Timer/Counter register neither have the value 0xFF nor 0x00 before writing to the asynchronous Timer Control Register (TCCR<sub>x</sub>), asynchronous Timer Counter Register (TCNT<sub>x</sub>), or asynchronous Output Compare Register (OCR<sub>x</sub>).

## 10.2 ATmega3290

### 10.2.1 ATmega3290 rev. C

- Interrupts may be lost when writing the timer registers in the asynchronous timer

#### 1. Interrupts may be lost when writing the timer registers in the asynchronous timer

The interrupt will be lost if a timer register that is synchronous timer clock is written when the asynchronous Timer/Counter register (TCNTx) is 0x00.

##### **Problem Fix/Workaround**

Always check that the asynchronous Timer/Counter register neither have the value 0xFF nor 0x00 before writing to the asynchronous Timer Control Register (TCCR<sub>x</sub>), asynchronous Timer Counter Register (TCNT<sub>x</sub>), or asynchronous Output Compare Register (OCR<sub>x</sub>).

### 10.2.2 ATmega3290 rev. B

Not sampled.

### 10.2.3 ATmega3290 rev. A

- LCD contrast voltage too high
- Interrupts may be lost when writing the timer registers in the asynchronous timer

#### 1. LCD contrast voltage too high

When the LCD is active and using low power waveform, the LCD contrast voltage can be too high. This occurs when  $V_{CC}$  is higher than  $V_{LCD}$ , and when using low LCD drivetime.

##### **Problem Fix/Workaround**

There are several possible workarounds:

- Use normal waveform instead of low power waveform
- Use drivetime of 375  $\mu$ s or longer

#### 2. Interrupts may be lost when writing the timer registers in the asynchronous timer

The interrupt will be lost if a timer register that is synchronous timer clock is written when the asynchronous Timer/Counter register (TCNT<sub>x</sub>) is 0x00.

##### **Problem Fix/Workaround**

Always check that the asynchronous Timer/Counter register neither have the value 0xFF nor 0x00 before writing to the asynchronous Timer Control Register (TCCR<sub>x</sub>), asynchronous Timer Counter Register (TCNT<sub>x</sub>), or asynchronous Output Compare Register (OCR<sub>x</sub>).

## 10.3 ATmega649

### 10.3.1 ATmega649 rev. A

- Interrupts may be lost when writing the timer registers in the asynchronous timer

#### 1. Interrupts may be lost when writing the timer registers in the asynchronous timer

The interrupt will be lost if a timer register that is synchronous timer clock is written when the asynchronous Timer/Counter register (TCNTx) is 0x00.

##### **Problem Fix/Wortkaround**

Always check that the asynchronous Timer/Counter register neither have the value 0xFF nor 0x00 before writing to the asynchronous Timer Control Register (TCCR<sub>x</sub>), asynchronous Timer Counter Register (TCNT<sub>x</sub>), or asynchronous Output Compare Register (OCR<sub>x</sub>).

## 10.4 ATmega6490

### 10.4.1 ATmega6490 rev. A

- Interrupts may be lost when writing the timer registers in the asynchronous timer

#### 1. Interrupts may be lost when writing the timer registers in the asynchronous timer

The interrupt will be lost if a timer register that is synchronous timer clock is written when the asynchronous Timer/Counter register (TCNT<sub>x</sub>) is 0x00.

##### **Problem Fix/Wortkaround**

Always check that the asynchronous Timer/Counter register neither have the value 0xFF nor 0x00 before writing to the asynchronous Timer Control Register (TCCR<sub>x</sub>), asynchronous Timer Counter Register (TCNT<sub>x</sub>), or asynchronous Output Compare Register (OCR<sub>x</sub>).

## 11.5 Rev. 2552G – 07/06

1. Updated [Table 14-2 on page 104](#), [Table 14-4 on page 104](#), [Table 16-3 on page 133](#), [Table 16-5 on page 134](#), [Table 16-5 on page 134](#), [Table 17-2 on page 153](#) and [Table 17-4 on page 154](#).
2. Updated [“Fast PWM Mode” on page 124](#).
3. Updated Features in [“USI – Universal Serial Interface” on page 195](#).
4. Added [“Clock speed considerations.” on page 202](#).
5. [“Errata” on page 379](#).

## 11.6 Rev. 2552F – 06/06

1. Updated [“Calibrated Internal RC Oscillator” on page 29](#).
2. Updated [“OSCCAL – Oscillator Calibration Register” on page 32](#)
3. Added [Table 28-2 on page 329](#).

## 11.7 Rev. 2552E – 04/06

1. Updated [“Calibrated Internal RC Oscillator” on page 29](#).

## 11.8 Rev. 2552D – 03/06

1. Updated [“Errata” on page 379](#).

## 11.9 Rev. 2552C – 03/06

1. Added [“Resources” on page 9](#).
2. Added Addresses in Registers.
3. Updated number of General Purpose I/O pins.
4. Updated code example in [“Bit 0 – IVCE: Interrupt Vector Change Enable” on page 53](#).
5. Updated Introduction in [“I/O-Ports” on page 59](#).
6. Updated [“SPI – Serial Peripheral Interface” on page 158](#).
7. Updated [“Bit 6 – ACBG: Analog Comparator Bandgap Select” on page 209](#).
8. Updated Features in [“Analog to Digital Converter” on page 211](#).
9. Updated [“Prescaling and Conversion Timing” on page 214](#).
10. Updated features in [“LCD Controller” on page 228](#).
11. Updated [“ATmega329/3290/649/6490 Boot Loader Parameters” on page 290](#).
12. Updated [“DC Characteristics” on page 310](#).
13. Updated [“” on page 334](#).

## 11.10 Rev. 2552B – 05/05

1. MLF-package alternative changed to “Quad Flat No-Lead/Micro Lead Frame Package QFN/MLF”.
2. Added “Pin Change Interrupt Timing” on page 54.
3. Updated Table 23-6 on page 242, Table 23-7 on page 243 and Table 27-15 on page 310.
4. Added Figure 27-12 on page 312.
5. Updated Figure 22-9 on page 219 and Figure 27-5 on page 304.
6. Updated algorithm “Enter Programming Mode” on page 299.
7. Added “Supply Current of I/O modules” on page 340.
8. Updated “Ordering Information” on page 372.

## 11.11 Rev. 2552A –11/04

1. Initial version.