

## Features

- High Performance, Low Power AVR<sup>®</sup> 8-Bit Microcontroller
- Advanced RISC Architecture
  - 131 Powerful Instructions – Most Single Clock Cycle Execution
  - 32 x 8 General Purpose Working Registers
  - Fully Static Operation
  - Up to 24 MIPS Throughput at 24 MHz
  - On-chip 2-cycle Multiplier
- Non-volatile Program and Data Memories
  - 4/8/16K Bytes of In-System Self-Programmable Flash (ATmega48/88/168)  
Endurance: 10,000 Write/Erase Cycles
  - Optional Boot Code Section with Independent Lock Bits  
In-System Programming by On-chip Boot Program  
True Read-While-Write Operation
  - 256/512/512 Bytes EEPROM (ATmega48/88/168)  
Endurance: 100,000 Write/Erase Cycles
  - 512/1K/1K Byte Internal SRAM (ATmega48/88/168)
  - Programming Lock for Software Security
- Peripheral Features
  - Two 8-bit Timer/Counters with Separate Prescaler and Compare Mode
  - One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
  - Real Time Counter with Separate Oscillator
  - Six PWM Channels
  - 8-channel 10-bit ADC in TQFP and MLF package
  - 6-channel 10-bit ADC in PDIP Package
  - Programmable Serial USART
  - Master/Slave SPI Serial Interface
  - Byte-oriented 2-wire Serial Interface
  - Programmable Watchdog Timer with Separate On-chip Oscillator
  - On-chip Analog Comparator
  - Interrupt and Wake-up on Pin Change
- Special Microcontroller Features
  - Power-on Reset and Programmable Brown-out Detection
  - Internal Calibrated Oscillator
  - External and Internal Interrupt Sources
  - Five Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, and Standby
- I/O and Packages
  - 23 Programmable I/O Lines
  - 28-pin PDIP, 32-lead TQFP and 32-pad MLF
- Operating Voltage:
  - 1.8 - 5.5V for ATmega48V/88V/168V
  - 2.7 - 5.5V for ATmega48/88/168
- Temperature Range:
  - -40°C to 85°C
- Speed Grade:
  - ATmega48V/88V/168V: 0 - 6 MHz @ 1.8 - 5.5V, 0 - 12 MHz @ 2.7 - 5.5V
  - ATmega48/88/168: 0 - 12 MHz @ 2.7 - 5.5V, 0 - 24 MHz @ 4.5 - 5.5V
- Low Power Consumption
  - Active Mode:
    - 1 MHz, 1.8V: 240µA
    - 32 kHz, 1.8V: 15µA (including Oscillator)
  - Power-down Mode:
    - 0.1µA at 1.8V



8-bit AVR<sup>®</sup>  
Microcontroller  
with 8K Bytes  
In-System  
Programmable  
Flash

ATmega48/V  
ATmega88/V  
ATmega168/V

Preliminary  
Summary

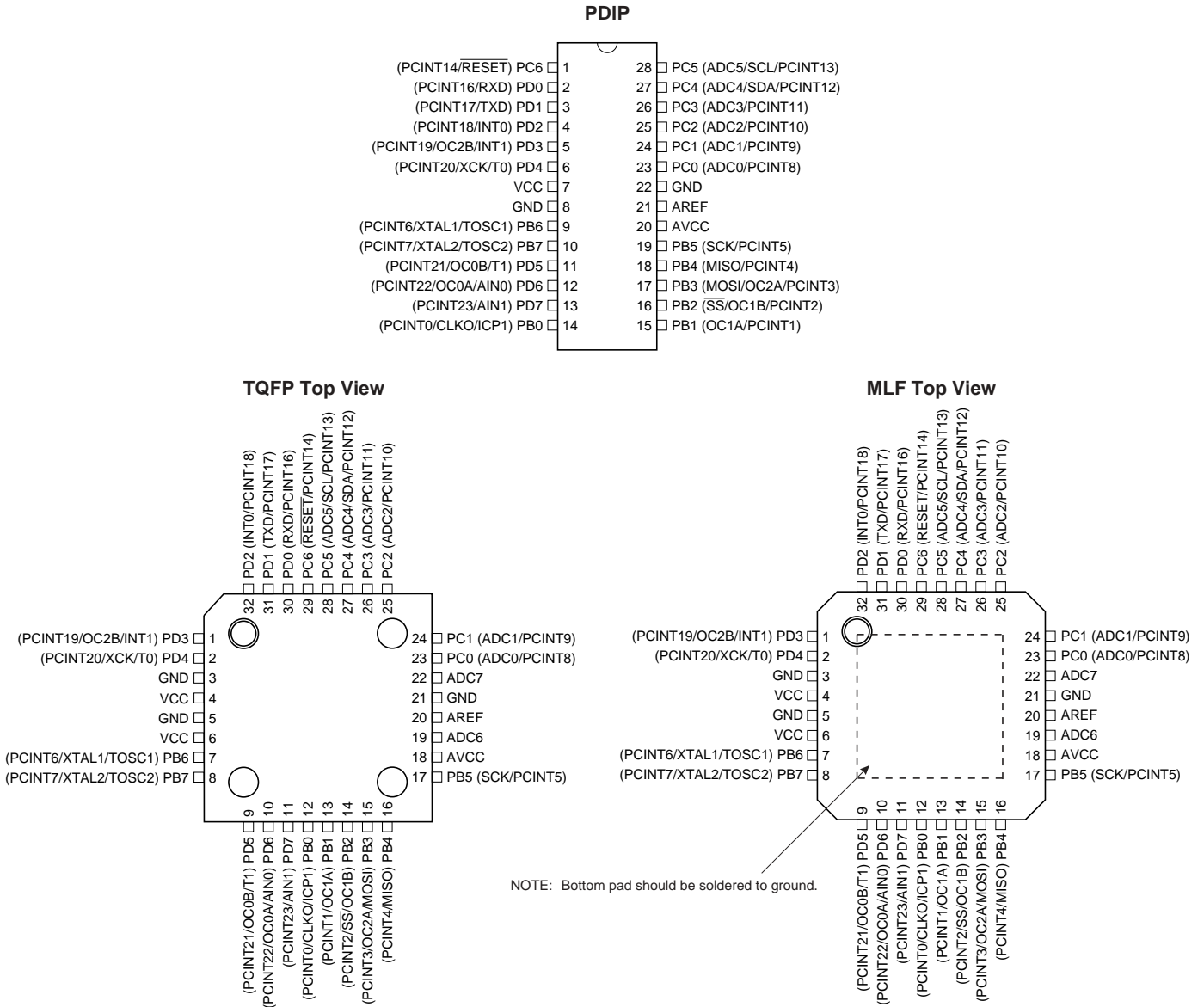
Rev. 2545BS-AVR-01/04



Note: This is a summary document. A complete document is available on our Web site at [www.atmel.com](http://www.atmel.com).

# Pin Configurations

Figure 1. Pinout ATmega48/88/168



## Disclaimer

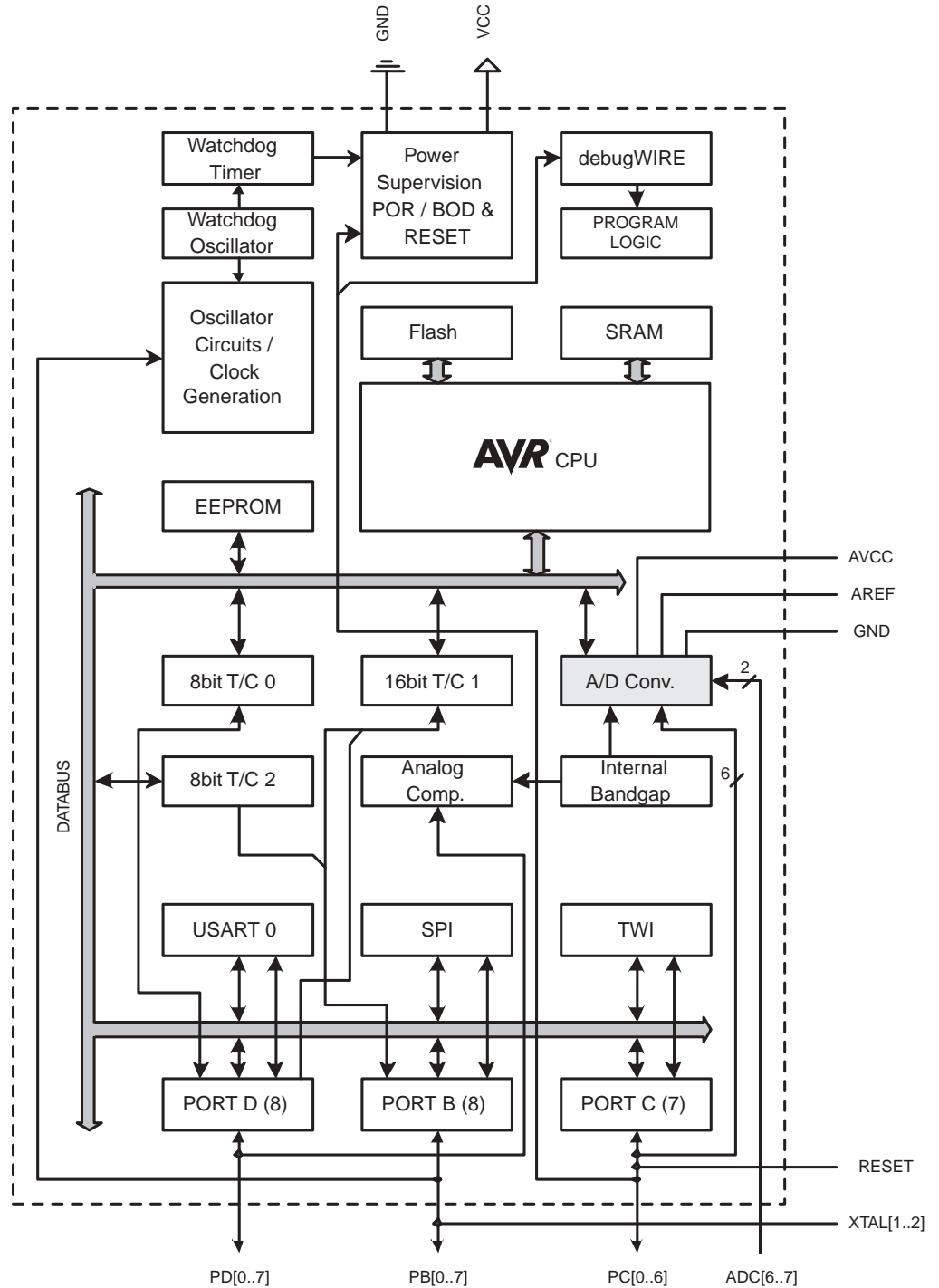
Typical values contained in this datasheet are based on simulations and characterization of other AVR microcontrollers manufactured on the same process technology. Min and Max values will be available after the device is characterized.

## Overview

The ATmega48/88/168 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 ATmega48/88/168 achieves throughputs approaching 1 MIPS per MHz allowing the system designer to optimize power consumption versus processing speed.

## Block Diagram

Figure 2. Block Diagram





The AVR core combines a rich instruction set with 32 general purpose working registers. All the 32 registers are directly connected to the Arithmetic Logic Unit (ALU), allowing two independent registers to be accessed in one single instruction executed in one clock cycle. The resulting architecture is more code efficient while achieving throughputs up to ten times faster than conventional CISC microcontrollers.

The ATmega48/88/168 provides the following features: 4K/8K/16K bytes of In-System Programmable Flash with Read-While-Write capabilities, 256/512/512 bytes EEPROM, 512/1K/1K bytes SRAM, 23 general purpose I/O lines, 32 general purpose working registers, three flexible Timer/Counters with compare modes, internal and external interrupts, a serial programmable USART, a byte-oriented 2-wire Serial Interface, an SPI serial port, a 6-channel 10-bit ADC (8 channels in TQFP and MLF packages), a programmable Watchdog Timer with internal Oscillator, and five software selectable power saving modes. The Idle mode stops the CPU while allowing the SRAM, Timer/Counters, USART, 2-wire Serial Interface, SPI port, and interrupt system to continue functioning. The Power-down mode saves the register contents but freezes the Oscillator, disabling all other chip functions until the next interrupt or hardware reset. In Power-save mode, the asynchronous timer continues to run, allowing the user to maintain a timer base while the rest of the device is sleeping. The ADC Noise Reduction mode stops the CPU and all I/O modules except asynchronous timer and ADC, to minimize switching noise during ADC conversions. In Standby mode, the crystal/resonator Oscillator is running while the rest of the device is sleeping. This allows very fast start-up combined with low power consumption.

The device is manufactured using Atmel's high density non-volatile memory technology. The On-chip ISP Flash allows the program memory to be reprogrammed In-System through an SPI serial interface, by a conventional non-volatile memory programmer, or by an On-chip Boot program running on the AVR core. The Boot program can use any interface to download the application program in the Application Flash memory. Software in the Boot Flash section will continue to run while the Application Flash section is updated, providing true Read-While-Write operation. By combining an 8-bit RISC CPU with In-System Self-Programmable Flash on a monolithic chip, the Atmel ATmega48/88/168 is a powerful microcontroller that provides a highly flexible and cost effective solution to many embedded control applications.

The ATmega48/88/168 AVR is supported with a full suite of program and system development tools including: C Compilers, Macro Assemblers, Program Debugger/Simulators, In-Circuit Emulators, and Evaluation kits.

## Comparison Between ATmega48, ATmega88, and ATmega168

The ATmega48, ATmega88 and ATmega168 differ only in memory sizes, boot loader support, and interrupt vector sizes. Table 1 summarizes the different memory and interrupt vector sizes for the three devices.

**Table 1.** Memory Size Summary

| Device    | Flash     | EEPROM    | RAM       | Interrupt Vector Size      |
|-----------|-----------|-----------|-----------|----------------------------|
| ATmega48  | 4K Bytes  | 256 Bytes | 512 Bytes | 1 instruction word/vector  |
| ATmega88  | 8K Bytes  | 512 Bytes | 1K Bytes  | 1 instruction word/vector  |
| ATmega168 | 16K Bytes | 512 Bytes | 1K Bytes  | 2 instruction words/vector |

ATmega88 and ATmega168 support a real Read-While-Write Self-Programming mechanism. There is a separate Boot Loader Section, and the SPM instruction can only execute from there. In ATmega48, there is no Read-While-Write support and no separate Boot Loader Section. The SPM instruction can execute from the entire Flash.

## Pin Descriptions

|   |   |
|---|---|
| <b>VCC</b>  | Digital supply voltage.   |
| <b>GND</b>  | Ground.   |
| <b>Port B (PB7..0) XTAL1/<br/>XTAL2/TOSC1/TOSC2</b> | <p>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.</p> <p>Depending on the clock selection fuse settings, PB6 can be used as input to the inverting Oscillator amplifier and input to the internal clock operating circuit.</p> <p>Depending on the clock selection fuse settings, PB7 can be used as output from the inverting Oscillator amplifier.</p> <p>If the Internal Calibrated RC Oscillator is used as chip clock source, PB7..6 is used as TOSC2..1 input for the Asynchronous Timer/Counter2 if the AS2 bit in ASSR is set.</p> <p>The various special features of Port B are elaborated in “Alternate Functions of Port B” on page 69 and “System Clock and Clock Options” on page 24.</p> |
| <b>Port C (PC5..0)</b>                              | <p>Port C is a 7-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The PC5..0 output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port C pins that are externally pulled low will source current if the pull-up resistors are activated. The Port C pins are tri-stated when a reset condition becomes active, even if the clock is not running.</p>   |
| <b>PC6/RESET</b>                                    | <p>If the RSTDISBL Fuse is programmed, PC6 is used as an I/O pin. Note that the electrical characteristics of PC6 differ from those of the other pins of Port C.</p> <p>If the RSTDISBL Fuse is unprogrammed, PC6 is used as a 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 Table 20 on page 41. Shorter pulses are not guaranteed to generate a Reset.</p> <p>The various special features of Port C are elaborated in “Alternate Functions of Port C” on page 73.</p>  |
| <b>Port D (PD7..0)</b>                              | <p>Port D is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port D output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port D pins that are externally pulled low will source current if the pull-up resistors are activated. The Port D pins are tri-stated when a reset condition becomes active, even if the clock is not running.</p> <p>The various special features of Port D are elaborated in “Alternate Functions of Port D” on page 75.</p>  |
| <b>AVCC</b>   | <p>AVCC is the supply voltage pin for the A/D Converter, PC3..0, and ADC7..6. It should be externally connected to <math>V_{CC}</math>, even if the ADC is not used. If the ADC is used, it should be connected to <math>V_{CC}</math> through a low-pass filter. Note that PC6..4 use digital supply voltage, <math>V_{CC}</math>.</p>   |
| <b>AREF</b>   | <p>AREF is the analog reference pin for the A/D Converter.</p>  |



**ADC7..6 (TQFP and MLF  
Package Only)**

In the TQFP and MLF package, ADC7..6 serve as analog inputs to the A/D converter. These pins are powered from the analog supply and serve as 10-bit ADC channels.

## Register Summary

| Address | Name     | Bit 7   | Bit 6   | Bit 5  | Bit 4 | Bit 3 | Bit 2           | Bit 1           | Bit 0  | Page    |
|---------|----------|---------|---------|--------|-------|-------|-----------------|-----------------|--------|---------|
| (0xFF)  | Reserved | –       | –       | –      | –     | –     | –               | –               | –      |         |
| (0xFE)  | Reserved | –       | –       | –      | –     | –     | –               | –               | –      |         |
| (0xFD)  | Reserved | –       | –       | –      | –     | –     | –               | –               | –      |         |
| (0xFC)  | Reserved | –       | –       | –      | –     | –     | –               | –               | –      |         |
| (0xFB)  | Reserved | –       | –       | –      | –     | –     | –               | –               | –      |         |
| (0xFA)  | Reserved | –       | –       | –      | –     | –     | –               | –               | –      |         |
| (0xF9)  | Reserved | –       | –       | –      | –     | –     | –               | –               | –      |         |
| (0xF8)  | Reserved | –       | –       | –      | –     | –     | –               | –               | –      |         |
| (0xF7)  | Reserved | –       | –       | –      | –     | –     | –               | –               | –      |         |
| (0xF6)  | Reserved | –       | –       | –      | –     | –     | –               | –               | –      |         |
| (0xF5)  | Reserved | –       | –       | –      | –     | –     | –               | –               | –      |         |
| (0xF4)  | Reserved | –       | –       | –      | –     | –     | –               | –               | –      |         |
| (0xF3)  | Reserved | –       | –       | –      | –     | –     | –               | –               | –      |         |
| (0xF2)  | Reserved | –       | –       | –      | –     | –     | –               | –               | –      |         |
| (0xF1)  | Reserved | –       | –       | –      | –     | –     | –               | –               | –      |         |
| (0xF0)  | Reserved | –       | –       | –      | –     | –     | –               | –               | –      |         |
| (0xEF)  | Reserved | –       | –       | –      | –     | –     | –               | –               | –      |         |
| (0xEE)  | Reserved | –       | –       | –      | –     | –     | –               | –               | –      |         |
| (0xED)  | Reserved | –       | –       | –      | –     | –     | –               | –               | –      |         |
| (0xEC)  | Reserved | –       | –       | –      | –     | –     | –               | –               | –      |         |
| (0xEB)  | Reserved | –       | –       | –      | –     | –     | –               | –               | –      |         |
| (0xEA)  | Reserved | –       | –       | –      | –     | –     | –               | –               | –      |         |
| (0xE9)  | Reserved | –       | –       | –      | –     | –     | –               | –               | –      |         |
| (0xE8)  | Reserved | –       | –       | –      | –     | –     | –               | –               | –      |         |
| (0xE7)  | Reserved | –       | –       | –      | –     | –     | –               | –               | –      |         |
| (0xE6)  | Reserved | –       | –       | –      | –     | –     | –               | –               | –      |         |
| (0xE5)  | Reserved | –       | –       | –      | –     | –     | –               | –               | –      |         |
| (0xE4)  | Reserved | –       | –       | –      | –     | –     | –               | –               | –      |         |
| (0xE3)  | Reserved | –       | –       | –      | –     | –     | –               | –               | –      |         |
| (0xE2)  | Reserved | –       | –       | –      | –     | –     | –               | –               | –      |         |
| (0xE1)  | Reserved | –       | –       | –      | –     | –     | –               | –               | –      |         |
| (0xE0)  | Reserved | –       | –       | –      | –     | –     | –               | –               | –      |         |
| (0xDF)  | Reserved | –       | –       | –      | –     | –     | –               | –               | –      |         |
| (0xDE)  | Reserved | –       | –       | –      | –     | –     | –               | –               | –      |         |
| (0xDD)  | Reserved | –       | –       | –      | –     | –     | –               | –               | –      |         |
| (0xDC)  | Reserved | –       | –       | –      | –     | –     | –               | –               | –      |         |
| (0xDB)  | Reserved | –       | –       | –      | –     | –     | –               | –               | –      |         |
| (0xDA)  | Reserved | –       | –       | –      | –     | –     | –               | –               | –      |         |
| (0xD9)  | Reserved | –       | –       | –      | –     | –     | –               | –               | –      |         |
| (0xD8)  | Reserved | –       | –       | –      | –     | –     | –               | –               | –      |         |
| (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     |         |         |        |       |       |                 |                 |        | 180     |
|         |          |         |         |        |       |       |                 |                 |        |         |
| (0xC5)  | UBRR0H   |         |         |        |       |       |                 |                 |        | 184     |
|         |          |         |         |        |       |       |                 |                 |        |         |
| (0xC4)  | UBRR0L   |         |         |        |       |       |                 |                 |        | 184     |
|         |          |         |         |        |       |       |                 |                 |        |         |
| (0xC3)  | Reserved | –       | –       | –      | –     | –     | –               | –               | –      |         |
|         |          |         |         |        |       |       |                 |                 |        |         |
| (0xC2)  | UCSR0C   | UMSEL01 | UMSEL00 | UPM01  | UPM00 | USBS0 | UCSZ01 / UDORD0 | UCSZ00 / UCPHA0 | UCPOL0 | 183/196 |
|         |          |         |         |        |       |       |                 |                 |        |         |
| (0xC1)  | UCSR0B   | RXCIE0  | TXCIE0  | UDRIE0 | RXEN0 | TXEN0 | UCSZ02          | RXB80           | TXB80  | 182     |
|         |          |         |         |        |       |       |                 |                 |        |         |
| (0xC0)  | UCSR0A   | RXC0    | TXC0    | UDRE0  | FE0   | DOR0  | UPE0            | U2X0            | MPCM0  | 180     |



| Address | Name     | Bit 7  | Bit 6  | Bit 5  | Bit 4  | Bit 3   | Bit 2   | Bit 1   | Bit 0   | Page |     |
|---------|----------|--|--------|--------|--------|---------|---------|---------|---------|------|-----|
| (0xBF)  | Reserved | –  | –      | –      | –      | –       | –       | –       | –       |      |     |
| (0xBE)  | Reserved | –  | –      | –      | –      | –       | –       | –       | –       |      |     |
| (0xBD)  | TWAMR    | TWAM6  | TWAM5  | TWAM4  | TWAM3  | TWAM2   | TWAM1   | TWAM0   | –       | 209  |     |
| (0xBC)  | TWCR     | TWINT  | TWEA   | TWSTA  | TWSTO  | TWWC    | TWEN    | –       | TWIE    | 206  |     |
| (0xBB)  | TWDR     | 2-wire Serial Interface Data Register                |        |        |        |         |         |         |         |      | 208 |
| (0xBA)  | TWAR     | TWA6   | TWA5   | TWA4   | TWA3   | TWA2    | TWA1    | TWA0    | TWGCE   | 208  |     |
| (0xB9)  | TWSR     | TWS7   | TWS6   | TWS5   | TWS4   | TWS3    | –       | TWPS1   | TWPS0   | 207  |     |
| (0xB8)  | TWBR     | 2-wire Serial Interface Bit Rate Register            |        |        |        |         |         |         |         |      | 206 |
| (0xB7)  | Reserved | –  | –      | –      | –      | –       | –       | –       | –       |      |     |
| (0xB6)  | ASSR     | –  | EXCLK  | AS2    | TCN2UB | OCR2AUB | OCR2BUB | TCR2AUB | TCR2BUB | 148  |     |
| (0xB5)  | Reserved | –  | –      | –      | –      | –       | –       | –       | –       |      |     |
| (0xB4)  | OCR2B    | Timer/Counter2 Output Compare Register B             |        |        |        |         |         |         |         |      | 147 |
| (0xB3)  | OCR2A    | Timer/Counter2 Output Compare Register A             |        |        |        |         |         |         |         |      | 147 |
| (0xB2)  | TCNT2    | Timer/Counter2 (8-bit)                               |        |        |        |         |         |         |         |      | 147 |
| (0xB1)  | TCCR2B   | FOC2A  | FOC2B  | –      | –      | WGM22   | CS22    | CS21    | CS20    | 146  |     |
| (0xB0)  | TCCR2A   | COM2A1   | COM2A0 | COM2B1 | COM2B0 | –       | –       | WGM21   | WGM20   | 143  |     |
| (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 Byte |        |        |        |         |         |         |         |      | 129 |
| (0x8A)  | OCR1BL   | Timer/Counter1 - Output Compare Register B Low Byte  |        |        |        |         |         |         |         |      | 129 |
| (0x89)  | OCR1AH   | Timer/Counter1 - Output Compare Register A High Byte |        |        |        |         |         |         |         |      | 129 |
| (0x88)  | OCR1AL   | Timer/Counter1 - Output Compare Register A Low Byte  |        |        |        |         |         |         |         |      | 129 |
| (0x87)  | ICR1H    | Timer/Counter1 - Input Capture Register High Byte    |        |        |        |         |         |         |         |      | 129 |
| (0x86)  | ICR1L    | Timer/Counter1 - Input Capture Register Low Byte     |        |        |        |         |         |         |         |      | 129 |
| (0x85)  | TCNT1H   | Timer/Counter1 - Counter Register High Byte          |        |        |        |         |         |         |         |      | 129 |
| (0x84)  | TCNT1L   | Timer/Counter1 - Counter Register Low Byte           |        |        |        |         |         |         |         |      | 129 |
| (0x83)  | Reserved | –  | –      | –      | –      | –       | –       | –       | –       |      |     |
| (0x82)  | TCCR1C   | FOC1A  | FOC1B  | –      | –      | –       | –       | –       | –       | 128  |     |
| (0x81)  | TCCR1B   | ICNC1  | ICES1  | –      | WGM13  | WGM12   | CS12    | CS11    | CS10    | 127  |     |
| (0x80)  | TCCR1A   | COM1A1   | COM1A0 | COM1B1 | COM1B0 | –       | –       | WGM11   | WGM10   | 125  |     |
| (0x7F)  | DIDR1    | –  | –      | –      | –      | –       | –       | AIN1D   | AIN0D   | 230  |     |
| (0x7E)  | DIDR0    | –  | –      | ADC5D  | ADC4D  | ADC3D   | ADC2D   | ADC1D   | ADC0D   | 245  |     |



| Address     | Name     | Bit 7  | Bit 6                | Bit 5   | Bit 4                | Bit 3   | Bit 2               | Bit 1    | Bit 0     | Page    |
|-------------|----------|--|----------------------|---------|----------------------|---------|---------------------|----------|-----------|---------|
| (0x7D)      | Reserved | –  | –                    | –       | –                    | –       | –                   | –        | –         |         |
| (0x7C)      | ADMUX    | REFS1  | REFS0                | ADLAR   | –                    | MUX3    | MUX2                | MUX1     | MUX0      | 241     |
| (0x7B)      | ADCSRB   | –  | ACME                 | –       | –                    | –       | ADTS2               | ADTS1    | ADTS0     | 244     |
| (0x7A)      | ADCSRA   | ADEN   | ADSC                 | ADATE   | ADIF                 | ADIE    | ADPS2               | ADPS1    | ADPS0     | 242     |
| (0x79)      | ADCH     | ADC Data Register High byte                      |                      |         |                      |         |                     |          |           | 244     |
| (0x78)      | ADCL     | ADC Data Register Low byte                       |                      |         |                      |         |                     |          |           | 244     |
| (0x77)      | Reserved | –  | –                    | –       | –                    | –       | –                   | –        | –         |         |
| (0x76)      | Reserved | –  | –                    | –       | –                    | –       | –                   | –        | –         |         |
| (0x75)      | Reserved | –  | –                    | –       | –                    | –       | –                   | –        | –         |         |
| (0x74)      | Reserved | –  | –                    | –       | –                    | –       | –                   | –        | –         |         |
| (0x73)      | Reserved | –  | –                    | –       | –                    | –       | –                   | –        | –         |         |
| (0x72)      | Reserved | –  | –                    | –       | –                    | –       | –                   | –        | –         |         |
| (0x71)      | Reserved | –  | –                    | –       | –                    | –       | –                   | –        | –         |         |
| (0x70)      | TIMSK2   | –  | –                    | –       | –                    | –       | OCIE2B              | OCIE2A   | TOIE2     | 150     |
| (0x6F)      | TIMSK1   | –  | –                    | ICIE1   | –                    | –       | OCIE1B              | OCIE1A   | TOIE1     | 130     |
| (0x6E)      | TIMSK0   | –  | –                    | –       | –                    | –       | OCIE0B              | OCIE0A   | TOIE0     | 100     |
| (0x6D)      | PCMSK2   | PCINT23  | PCINT22              | PCINT21 | PCINT20              | PCINT19 | PCINT18             | PCINT17  | PCINT16   | 83      |
| (0x6C)      | PCMSK1   | –  | PCINT14              | PCINT13 | PCINT12              | PCINT11 | PCINT10             | PCINT9   | PCINT8    | 83      |
| (0x6B)      | PCMSK0   | PCINT7   | PCINT6               | PCINT5  | PCINT4               | PCINT3  | PCINT2              | PCINT1   | PCINT0    | 84      |
| (0x6A)      | Reserved | –  | –                    | –       | –                    | –       | –                   | –        | –         |         |
| (0x69)      | EICRA    | –  | –                    | –       | –                    | ISC11   | ISC10               | ISC01    | ISC00     | 80      |
| (0x68)      | PCICR    | –  | –                    | –       | –                    | –       | PCIE2               | PCIE1    | PCIE0     |         |
| (0x67)      | Reserved | –  | –                    | –       | –                    | –       | –                   | –        | –         |         |
| (0x66)      | OSCCAL   | Oscillator Calibration Register                  |                      |         |                      |         |                     |          |           | 30      |
| (0x65)      | Reserved | –  | –                    | –       | –                    | –       | –                   | –        | –         |         |
| (0x64)      | PRR      | PRTWI  | PRTIM2               | PRTIM0  | –                    | PRTIM1  | PRSPI               | PRUSART0 | PRADC     | 37      |
| (0x63)      | Reserved | –  | –                    | –       | –                    | –       | –                   | –        | –         |         |
| (0x62)      | Reserved | –  | –                    | –       | –                    | –       | –                   | –        | –         |         |
| (0x61)      | CLKPR    | CLKPCE   | –                    | –       | –                    | CLKPS3  | CLKPS2              | CLKPS1   | CLKPS0    | 33      |
| (0x60)      | WDTCR    | WDIF   | WDIE                 | WDP3    | WDCE                 | WDE     | WDP2                | WDP1     | WDP0      | 49      |
| 0x3F (0x5F) | SREG     | I  | T                    | H       | S                    | V       | N                   | Z        | C         | 9       |
| 0x3E (0x5E) | SPH      | –  | –                    | –       | –                    | –       | (SP10) <sup>5</sup> | SP9      | SP8       | 11      |
| 0x3D (0x5D) | SPL      | SP7  | SP6                  | SP5     | SP4                  | SP3     | SP2                 | SP1      | SP0       | 11      |
| 0x3C (0x5C) | Reserved | –  | –                    | –       | –                    | –       | –                   | –        | –         |         |
| 0x3B (0x5B) | Reserved | –  | –                    | –       | –                    | –       | –                   | –        | –         |         |
| 0x3A (0x5A) | Reserved | –  | –                    | –       | –                    | –       | –                   | –        | –         |         |
| 0x39 (0x59) | Reserved | –  | –                    | –       | –                    | –       | –                   | –        | –         |         |
| 0x38 (0x58) | Reserved | –  | –                    | –       | –                    | –       | –                   | –        | –         |         |
| 0x37 (0x57) | SPMCSR   | SPMIE  | (RWWSB) <sup>5</sup> | –       | (RWWSR) <sup>5</sup> | BLBSET  | PGWRT               | PGERS    | SELFPRGEN | 260     |
| 0x36 (0x56) | Reserved | –  | –                    | –       | –                    | –       | –                   | –        | –         |         |
| 0x35 (0x55) | MCUCR    | –  | –                    | –       | PUD                  | –       | –                   | IVSEL    | IVCE      |         |
| 0x34 (0x54) | MCUSR    | –  | –                    | –       | –                    | WDRF    | BORF                | EXTRF    | PORF      |         |
| 0x33 (0x53) | SMCR     | –  | –                    | –       | –                    | SM2     | SM1                 | SM0      | SE        | 35      |
| 0x32 (0x52) | Reserved | –  | –                    | –       | –                    | –       | –                   | –        | –         |         |
| 0x31 (0x51) | MONDR    | Monitor Data Register                            |                      |         |                      |         |                     |          |           |         |
| 0x30 (0x50) | ACSR     | ACD  | ACBG                 | ACO     | ACI                  | ACIE    | ACIC                | ACIS1    | ACIS0     | 228     |
| 0x2F (0x4F) | Reserved | –  | –                    | –       | –                    | –       | –                   | –        | –         |         |
| 0x2E (0x4E) | SPDR     | SPI Data Register                                |                      |         |                      |         |                     |          |           | 160     |
| 0x2D (0x4D) | SPSR     | SPIF   | WCOL                 | –       | –                    | –       | –                   | –        | SPI2X     | 160     |
| 0x2C (0x4C) | SPCR     | SPIE   | SPE                  | DORD    | MSTR                 | CPOL    | CPHA                | SPR1     | SPR0      | 158     |
| 0x2B (0x4B) | GPOR2    | General Purpose I/O Register 2                   |                      |         |                      |         |                     |          |           | 23      |
| 0x2A (0x4A) | GPOR1    | General Purpose I/O Register 1                   |                      |         |                      |         |                     |          |           | 23      |
| 0x29 (0x49) | Reserved | –  | –                    | –       | –                    | –       | –                   | –        | –         |         |
| 0x28 (0x48) | OCR0B    | Timer/Counter0 Output Compare Register B         |                      |         |                      |         |                     |          |           |         |
| 0x27 (0x47) | OCR0A    | Timer/Counter0 Output Compare Register A         |                      |         |                      |         |                     |          |           |         |
| 0x26 (0x46) | TCNT0    | Timer/Counter0 (8-bit)                           |                      |         |                      |         |                     |          |           |         |
| 0x25 (0x45) | TCCR0B   | FOC0A  | FOC0B                | –       | –                    | WGM02   | CS02                | CS01     | CS00      |         |
| 0x24 (0x44) | TCCR0A   | COM0A1   | COM0A0               | COM0B1  | COM0B0               | –       | –                   | WGM01    | WGM00     |         |
| 0x23 (0x43) | GTCCR    | TSM  | –                    | –       | –                    | –       | –                   | PSRASY   | PSRSYNC   | 103/152 |
| 0x22 (0x42) | EEARH    | (EEPROM Address Register High Byte) <sup>5</sup> |                      |         |                      |         |                     |          |           | 18      |
| 0x21 (0x41) | EEARL    | EEPROM Address Register Low Byte                 |                      |         |                      |         |                     |          |           | 18      |
| 0x20 (0x40) | EEDR     | EEPROM Data Register                             |                      |         |                      |         |                     |          |           | 18      |
| 0x1F (0x3F) | EEDR     | –  | –                    | EEP01   | EEP00                | EERIE   | EEMPE               | EEPE     | EERE      | 18      |
| 0x1E (0x3E) | GPOR0    | General Purpose I/O Register 0                   |                      |         |                      |         |                     |          |           | 23      |
| 0x1D (0x3D) | EIMSK    | –  | –                    | –       | –                    | –       | –                   | INT1     | INT0      | 81      |
| 0x1C (0x3C) | EIFR     | –  | –                    | –       | –                    | –       | –                   | INTF1    | INTF0     | 82      |



| Address     | Name     | Bit 7  | Bit 6  | Bit 5  | Bit 4  | Bit 3  | Bit 2  | Bit 1  | Bit 0  | Page |
|-------------|----------|--------|--------|--------|--------|--------|--------|--------|--------|------|
| 0x1B (0x3B) | PCIFR    | –      | –      | –      | –      | –      | PCIF2  | PCIF1  | PCIF0  |      |
| 0x1A (0x3A) | Reserved | –      | –      | –      | –      | –      | –      | –      | –      |      |
| 0x19 (0x39) | Reserved | –      | –      | –      | –      | –      | –      | –      | –      |      |
| 0x18 (0x38) | Reserved | –      | –      | –      | –      | –      | –      | –      | –      |      |
| 0x17 (0x37) | TIFR2    | –      | –      | –      | –      | –      | OCF2B  | OCF2A  | TOV2   | 151  |
| 0x16 (0x36) | TIFR1    | –      | –      | ICF1   | –      | –      | OCF1B  | OCF1A  | TOV1   | 130  |
| 0x15 (0x35) | TIFR0    | –      | –      | –      | –      | –      | OCF0B  | OCF0A  | TOV0   |      |
| 0x14 (0x34) | Reserved | –      | –      | –      | –      | –      | –      | –      | –      |      |
| 0x13 (0x33) | Reserved | –      | –      | –      | –      | –      | –      | –      | –      |      |
| 0x12 (0x32) | Reserved | –      | –      | –      | –      | –      | –      | –      | –      |      |
| 0x11 (0x31) | Reserved | –      | –      | –      | –      | –      | –      | –      | –      |      |
| 0x10 (0x30) | Reserved | –      | –      | –      | –      | –      | –      | –      | –      |      |
| 0x0F (0x2F) | Reserved | –      | –      | –      | –      | –      | –      | –      | –      |      |
| 0x0E (0x2E) | Reserved | –      | –      | –      | –      | –      | –      | –      | –      |      |
| 0x0D (0x2D) | Reserved | –      | –      | –      | –      | –      | –      | –      | –      |      |
| 0x0C (0x2C) | Reserved | –      | –      | –      | –      | –      | –      | –      | –      |      |
| 0x0B (0x2B) | PORTD    | PORTD7 | PORTD6 | PORTD5 | PORTD4 | PORTD3 | PORTD2 | PORTD1 | PORTD0 | 79   |
| 0x0A (0x2A) | DDRD     | DDD7   | DDD6   | DDD5   | DDD4   | DDD3   | DDD2   | DDD1   | DDD0   | 79   |
| 0x09 (0x29) | PIND     | PIND7  | PIND6  | PIND5  | PIND4  | PIND3  | PIND2  | PIND1  | PIND0  | 79   |
| 0x08 (0x28) | PORTC    | –      | PORTC6 | PORTC5 | PORTC4 | PORTC3 | PORTC2 | PORTC1 | PORTC0 | 79   |
| 0x07 (0x27) | DDRC     | –      | DDC6   | DDC5   | DDC4   | DDC3   | DDC2   | DDC1   | DDC0   | 79   |
| 0x06 (0x26) | PINC     | –      | PINC6  | PINC5  | PINC4  | PINC3  | PINC2  | PINC1  | PINC0  | 79   |
| 0x05 (0x25) | PORTB    | PORTB7 | PORTB6 | PORTB5 | PORTB4 | PORTB3 | PORTB2 | PORTB1 | PORTB0 | 79   |
| 0x04 (0x24) | DDRB     | DDB7   | DDB6   | DDB5   | DDB4   | DDB3   | DDB2   | DDB1   | DDB0   | 79   |
| 0x03 (0x23) | PINB     | PINB7  | PINB6  | PINB5  | PINB4  | PINB3  | PINB2  | PINB1  | PINB0  | 79   |
| 0x02 (0x22) | Reserved | –      | –      | –      | –      | –      | –      | –      | –      |      |
| 0x01 (0x21) | Reserved | –      | –      | –      | –      | –      | –      | –      | –      |      |
| 0x0 (0x20)  | Reserved | –      | –      | –      | –      | –      | –      | –      | –      |      |

- 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 AVR's, 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 ATmega48/88/168 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.
  5. Only valid for ATmega88/168

## 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                                     | Rd,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                                     | Rd,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             | $R1:R0 \leftarrow (Rd \times Rr) \lll 1$              | Z,C        | 2       |
| FMULS                                    | Rd, Rr   | Fractional Multiply Signed               | $R1:R0 \leftarrow (Rd \times Rr) \lll 1$              | Z,C        | 2       |
| FMULSU                                   | Rd, Rr   | Fractional Multiply Signed with Unsigned | $R1:R0 \leftarrow (Rd \times Rr) \lll 1$              | Z,C        | 2       |
| <b>BRANCH INSTRUCTIONS</b>               |          |  |   |            |         |
| RJMP                                     | k        | Relative Jump                            | $PC \leftarrow PC + k + 1$                            | None       | 2       |
| IJMP                                     |          | Indirect Jump to (Z)                     | $PC \leftarrow Z$                                     | None       | 2       |
| JMP <sup>(1)</sup>                       | 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 <sup>(1)</sup>                      | 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   |
| SBRS                                     | 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     |
| BRTC                                     | k        | Branch if T Flag Cleared                 | if $(T = 0)$ then $PC \leftarrow PC + k + 1$          | None       | 1/2     |
| BRVS                                     | k        | Branch if Overflow Flag is Set           | if $(V = 1)$ then $PC \leftarrow PC + k + 1$          | None       | 1/2     |
| BRVC                                     | k        | Branch if Overflow Flag is Cleared       | if $(V = 0)$ then $PC \leftarrow PC + k + 1$          | None       | 1/2     |

| Mnemonics                            | Operands | Description                      | Operation                                | Flags   | #Clocks |
|--------------------------------------|----------|----------------------------------|--|---------|---------|
| 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    | -       |
| IN                                   | Rd, P    | In Port                          | Rd ← P                                   | None    | 1       |
| OUT                                  | P, Rr    | Out Port                         | P ← Rr                                   | None    | 1       |
| PUSH                                 | Rr       | Push Register on Stack           | STACK ← Rr                               | None    | 2       |

| Mnemonics                       | Operands | Description             | Operation                                | Flags | #Clocks |
|---------------------------------|----------|-------------------------|--|-------|---------|
| POP                             | Rd       | Pop Register from Stack | Rd ← 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     |

Note: 1. These instructions are only available in ATmega168.



## Ordering Information

### ATmega48

| Speed (MHz)       | Power Supply | Ordering Code                 | Package | Operation Range               |
|-------------------|--------------|-------------------------------|---------|-------------------------------|
| 12 <sup>(3)</sup> | 1.8 - 5.5    | ATmega48V-12AI                | 32A     | Industrial<br>(-40°C to 85°C) |
|                   |              | ATmega48V-12PI                | 28P3    |                               |
|                   |              | ATmega48V-12MI                | 32M1-A  |                               |
|                   |              | ATmega48V-12AJ <sup>(2)</sup> | 32A     |                               |
|                   |              | ATmega48V-12PJ <sup>(2)</sup> | 28P3    |                               |
|                   |              | ATmega48V-12MJ <sup>(2)</sup> | 32M1-A  |                               |
| 24 <sup>(3)</sup> | 2.7 - 5.5    | ATmega48-24AI                 | 32A     | Industrial<br>(-40°C to 85°C) |
|                   |              | ATmega48-24PI                 | 28P3    |                               |
|                   |              | ATmega48-24MI                 | 32M1-A  |                               |
|                   |              | ATmega48-24AJ <sup>(2)</sup>  | 32A     |                               |
|                   |              | ATmega48-24PJ <sup>(2)</sup>  | 28P3    |                               |
|                   |              | ATmega48-24MJ <sup>(2)</sup>  | 32M1-A  |                               |

- Note:
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
  3. See Figure 131 on page 293 and Figure 132 on page 293.

| Package Type  |   |
|---------------|---|
| <b>32A</b>    | 32-lead, Thin (1.0 mm) Plastic Quad Flat Package (TQFP)                     |
| <b>28P3</b>   | 28-lead, 0.300" Wide, Plastic Dual Inline Package (PDIP)                    |
| <b>32M1-A</b> | 32-pad, 5 x 5 x 1.0 body, Lead Pitch 0.50 mm Micro Lead Frame Package (MLF) |

## ATmega88

| Speed (MHz)       | Power Supply | Ordering Code   | Package  | Operation Range               |
|-------------------|--------------|---|--|-------------------------------|
| 12 <sup>(3)</sup> | 1.8 - 5.5    | ATmega88V-12AI<br>ATmega88V-12PI<br>ATmega88V-12MI<br>ATmega88V-12AJ <sup>(2)</sup><br>ATmega88V-12PJ <sup>(2)</sup><br>ATmega88V-12MJ <sup>(2)</sup> | 32A<br>28P3<br>32M1-A<br>32A<br>28P3<br>32M1-A | Industrial<br>(-40°C to 85°C) |
| 24 <sup>(3)</sup> | 2.7 - 5.5    | ATmega88-24AI<br>ATmega88-24PI<br>ATmega88-24MI<br>ATmega88-24AJ <sup>(2)</sup><br>ATmega88-24PJ <sup>(2)</sup><br>ATmega88-24MJ <sup>(2)</sup>       | 32A<br>28P3<br>32M1-A<br>32A<br>28P3<br>32M1-A | Industrial<br>(-40°C to 85°C) |

- Note:
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
  3. See Figure 131 on page 293 and Figure 132 on page 293.

| Package Type  |   |
|---------------|---|
| <b>32A</b>    | 32-lead, Thin (1.0 mm) Plastic Quad Flat Package (TQFP)                     |
| <b>28P3</b>   | 28-lead, 0.300" Wide, Plastic Dual Inline Package (PDIP)                    |
| <b>32M1-A</b> | 32-pad, 5 x 5 x 1.0 body, Lead Pitch 0.50 mm Micro Lead Frame Package (MLF) |



## ATmega168

| Speed (MHz)       | Power Supply | Ordering Code   | Package  | Operation Range               |
|-------------------|--------------|---|--|-------------------------------|
| 12 <sup>(3)</sup> | 1.8 - 5.5    | ATmega168V-12AI<br>ATmega168V-12PI<br>ATmega168V-12MI<br>ATmega168V-12AJ <sup>(2)</sup><br>ATmega168V-12PJ <sup>(2)</sup><br>ATmega168V-12MJ <sup>(2)</sup> | 32A<br>28P3<br>32M1-A<br>32A<br>28P3<br>32M1-A | Industrial<br>(-40°C to 85°C) |
| 24 <sup>(3)</sup> | 2.7 - 5.5    | ATmega168-24AI<br>ATmega168-24PI<br>ATmega168-24MI<br>ATmega168-24AJ <sup>(2)</sup><br>ATmega168-24PJ <sup>(2)</sup><br>ATmega168-24MJ <sup>(2)</sup>       | 32A<br>28P3<br>32M1-A<br>32A<br>28P3<br>32M1-A | Industrial<br>(-40°C to 85°C) |

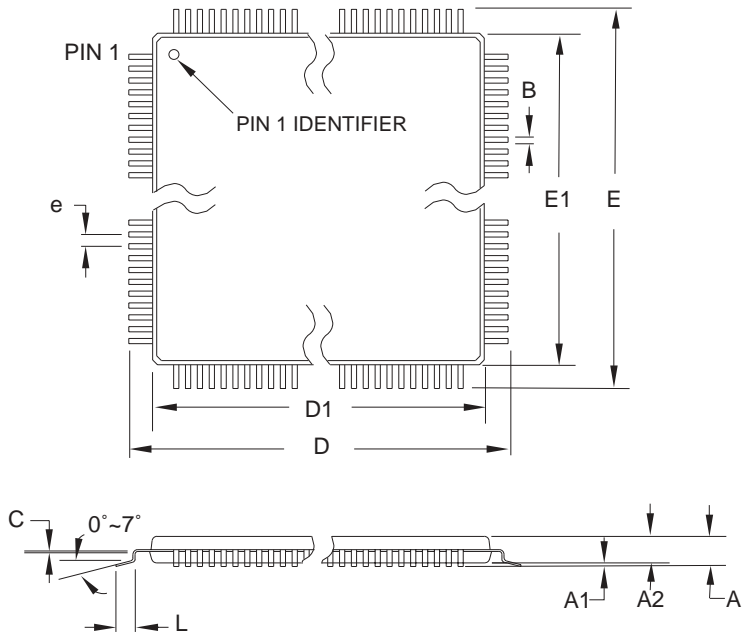
- Note:
1. This device can also be supplied in wafer form. Please contact your local Atmel sales office for detailed ordering information and minimum quantities.
  2. Pb-free packaging alternative
  3. See Figure 131 on page 293 and Figure 132 on page 293.

| Package Type  |   |
|---------------|---|
| <b>32A</b>    | 32-lead, Thin (1.0 mm) Plastic Quad Flat Package (TQFP)                     |
| <b>28P3</b>   | 28-lead, 0.300" Wide, Plastic Dual Inline Package (PDIP)                    |
| <b>32M1-A</b> | 32-pad, 5 x 5 x 1.0 body, Lead Pitch 0.50 mm Micro Lead Frame Package (MLF) |



Packaging Information

32A




COMMON DIMENSIONS  
(Unit of Measure = mm)

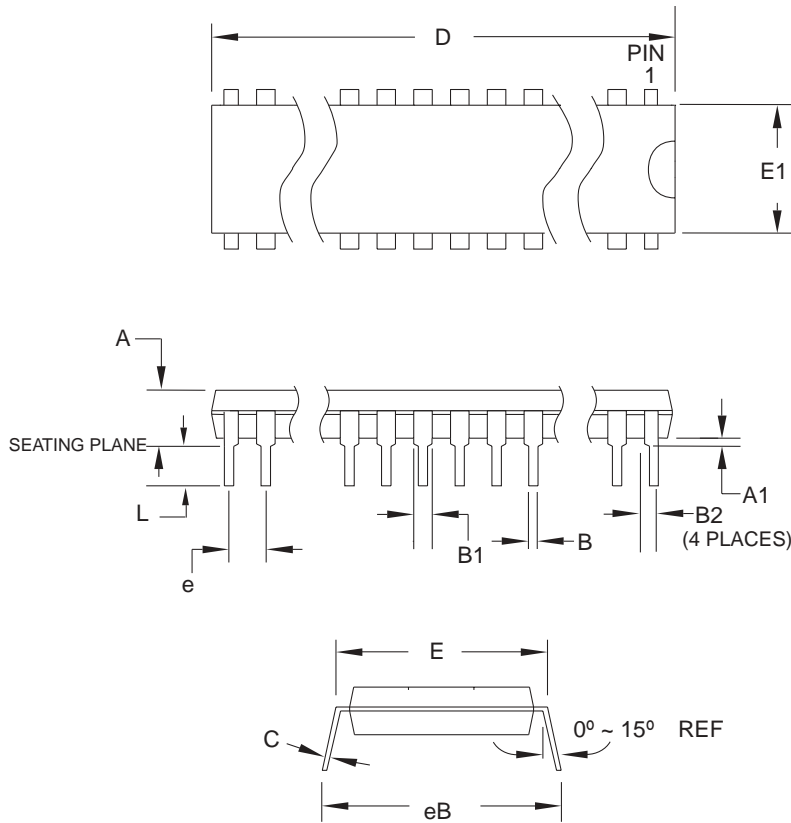
| SYMBOL | MIN      | NOM  | MAX  | NOTE   |
|--------|----------|------|------|--------|
| A      | -        | -    | 1.20 |        |
| A1     | 0.05     | -    | 0.15 |        |
| A2     | 0.95     | 1.00 | 1.05 |        |
| D      | 8.75     | 9.00 | 9.25 |        |
| D1     | 6.90     | 7.00 | 7.10 | Note 2 |
| E      | 8.75     | 9.00 | 9.25 |        |
| E1     | 6.90     | 7.00 | 7.10 | Note 2 |
| B      | 0.30     | -    | 0.45 |        |
| C      | 0.09     | -    | 0.20 |        |
| L      | 0.45     | -    | 0.75 |        |
| e      | 0.80 TYP |      |      |        |

- Notes:
1. This package conforms to JEDEC reference MS-026, Variation ABA.
  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 0.10 mm maximum.

10/5/2001

|  |  |             |      |
|--|--|-------------|------|
|  2325 Orchard Parkway<br>San Jose, CA 95131 | TITLE  | DRAWING NO. | REV. |
|  | <b>32A</b> , 32-lead, 7 x 7 mm Body Size, 1.0 mm Body Thickness,<br>0.8 mm Lead Pitch, Thin Profile Plastic Quad Flat Package (TQFP) | 32A         | B    |

28P3



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

| SYMBOL | MIN       | NOM | MAX    | NOTE   |
|--------|-----------|-----|--------|--------|
| A      | -         | -   | 4.5724 |        |
| A1     | 0.508     | -   | -      |        |
| D      | 34.544    | -   | 34.798 | Note 1 |
| E      | 7.620     | -   | 8.255  |        |
| E1     | 7.112     | -   | 7.493  | Note 1 |
| B      | 0.381     | -   | 0.533  |        |
| B1     | 1.143     | -   | 1.397  |        |
| B2     | 0.762     | -   | 1.143  |        |
| L      | 3.175     | -   | 3.429  |        |
| C      | 0.203     | -   | 0.356  |        |
| eB     | -         | -   | 10.160 |        |
| e      | 2.540 TYP |     |        |        |

Note: 1. Dimensions D and E1 do not include mold Flash or Protrusion.  
Mold Flash or Protrusion shall not exceed 0.25 mm (0.010").

09/28/01



2325 Orchard Parkway  
San Jose, CA 95131

**TITLE**

**28P3**, 28-lead (0.300"/7.62 mm Wide) Plastic Dual  
Inline Package (PDIP)

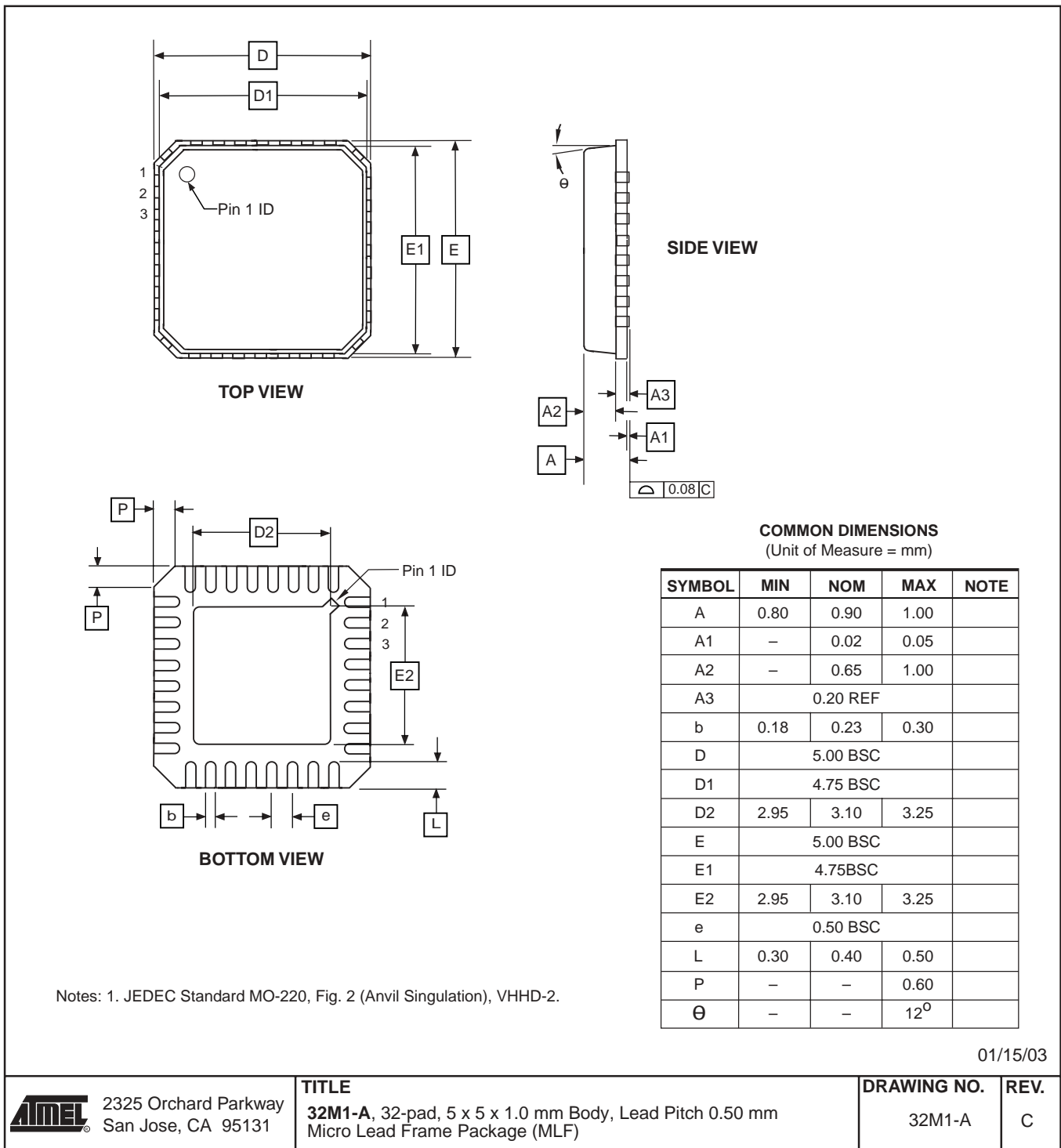
**DRAWING NO.**

28P3

**REV.**

B

## 32M1-A





## Errata ATmega48

The revision letter in this section refers to the revision of the ATmega48 device.

### Rev A

- **Wrong values read after Erase Only operation**
- **Watchdog Timer Interrupt disabled**
- **Start-up time with Crystal Oscillator is higher than expected**
- **High Power Consumption in Power-down with External Clock**
- **Asynchronous Oscillator does not stop in Power-down**

#### 1. **Wrong values read after Erase Only operation**

At supply voltages below 2.7 V, an EEPROM location that is erased by the Erase Only operation may read as programmed (0x00).

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

If it is necessary to read an EEPROM location after Erase Only, use an Atomic Write operation with 0xFF as data in order to erase a location. In any case, the Write Only operation can be used as intended. Thus no special considerations are needed as long as the erased location is not read before it is programmed.

#### 2. **Watchdog Timer Interrupt disabled**

If the watchdog timer interrupt flag is not cleared before a new timeout occurs, the watchdog will be disabled, and the interrupt flag will automatically be cleared. This is only applicable in interrupt only mode. If the Watchdog is configured to reset the device in the watchdog time-out following an interrupt, the device works correctly.

##### **Problem fix / Workaround**

Make sure there is enough time to always service the first timeout event before a new watchdog timeout occurs. This is done by selecting a long enough time-out period.

#### 3. **Start-up time with Crystal Oscillator is higher than expected**

The clock counting part of the start-up time is about 2 times higher than expected for all start-up periods when running on an external Crystal. This applies only when waking up by reset. Wake-up from power down is not affected. For most settings, the clock counting parts is a small fraction of the overall start-up time, and thus, the problem can be ignored. The exception is when using a very low frequency crystal like for instance a 32 kHz clock crystal.

##### **Problem fix / Workaround**

No known workaround.

#### 4. **High Power Consumption in Power-down with External Clock**

The power consumption in power down with an active external clock is about 10 times higher than when using internal RC or external oscillators.

##### **Problem fix / Workaround**

Stop the external clock when the device is in power down.

#### 5. **Asynchronous Oscillator does not stop in Power-down**

The Asynchronous oscillator does not stop when entering power down mode. This leads to higher power consumption than expected.

##### **Problem fix / Workaround**

Manually disable the asynchronous timer before entering power down.

**Errata ATmega88**

The revision letter in this section refers to the revision of the ATmega88 device.

**Rev A**

- **Wrong values read after Erase Only operation**

**1. Wrong values read after Erase Only operation**

At supply voltages below 2.7 V, an EEPROM location that is erased by the Erase Only operation may read as programmed (0x00).

**Problem Fix/Workaround**

If it is necessary to read an EEPROM location after Erase Only, use an Atomic Write operation with 0xFF as data in order to erase a location. In any case, the Write Only operation can be used as intended. Thus no special considerations are needed as long as the erased location is not read before it is programmed.

## Errata ATmega168

The revision letter in this section refers to the revision of the ATmega168 device.

### Rev A

- **Wrong values read after Erase Only operation**

#### 1. **Wrong values read after Erase Only operation**

At supply voltages below 2.7 V, an EEPROM location that is erased by the Erase Only operation may read as programmed (0x00).

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

If it is necessary to read an EEPROM location after Erase Only, use an Atomic Write operation with 0xFF as data in order to erase a location. In any case, the Write Only operation can be used as intended. Thus no special considerations are needed as long as the erased location is not read before it is programmed.

## Datasheet Change Log

Changes from Rev.  
2545A-09/03 to Rev.  
2545B-01/04

Please note that the referring page numbers in this section are referred to this document. The referring revision in this section are referring to the document revision.

1. Added PDIP to “I/O and Packages”, updated “Speed Grade” and Power Consumption Estimates in “Features” on page 1.
2. Updated “Stack Pointer” on page 11 with RAMEND as recommended Stack Pointer value.
3. Added section “Power Reduction Register” on page 37 and a note regarding the use of the PRR bits to 2-wire, Timer/Counters, USART, Analog Comparator and ADC sections.
4. Updated “Watchdog Timer” on page 46.
5. Updated Figure 55 on page 125 and Table 56 on page 126.
6. Extra Compare Match Interrupt OCF2B added to features in section “8-bit Timer/Counter2 with PWM and Asynchronous Operation” on page 132
7. Updated Table 19 on page 37, Table 102 on page 245, Table 118 to Table 121 on page 272 to 273 and Table 98 on page 236. Added note 2 to Table 115 on page 270. Fixed typo in Table 42 on page 81.
8. Updated whole “ATmega48/88/168 Typical Characteristics – Preliminary Data” on page 298.
9. Added item 2 to 5 in “Errata ATmega48” on page 20.
10. Renamed the following bits:
  - SPMEN to SELFPRGEN,
  - PSR2 to PSRASY
  - PSR10 to PSRSYNC
  - Watchdog Reset to Watchdog System Reset.
11. Updated C code examples containing old IAR syntax.
12. Updated BLBSET description in “Store Program Memory Control and Status Register – SPMCSR” on page 260.



## Atmel Corporation

2325 Orchard Parkway  
San Jose, CA 95131, USA  
Tel: 1(408) 441-0311  
Fax: 1(408) 487-2600

## Regional Headquarters

### Europe

Atmel Sarl  
Route des Arsenalux 41  
Case Postale 80  
CH-1705 Fribourg  
Switzerland  
Tel: (41) 26-426-5555  
Fax: (41) 26-426-5500

### Asia

Room 1219  
Chinachem Golden Plaza  
77 Mody Road Tsimshatsui  
East Kowloon  
Hong Kong  
Tel: (852) 2721-9778  
Fax: (852) 2722-1369

### Japan

9F, Tonetsu Shinkawa Bldg.  
1-24-8 Shinkawa  
Chuo-ku, Tokyo 104-0033  
Japan  
Tel: (81) 3-3523-3551  
Fax: (81) 3-3523-7581

## Atmel Operations

### Memory

2325 Orchard Parkway  
San Jose, CA 95131, USA  
Tel: 1(408) 441-0311  
Fax: 1(408) 436-4314

### Microcontrollers

2325 Orchard Parkway  
San Jose, CA 95131, USA  
Tel: 1(408) 441-0311  
Fax: 1(408) 436-4314

La Chantrerie  
BP 70602  
44306 Nantes Cedex 3, France  
Tel: (33) 2-40-18-18-18  
Fax: (33) 2-40-18-19-60

### ASIC/ASSP/Smart Cards

Zone Industrielle  
13106 Rousset Cedex, France  
Tel: (33) 4-42-53-60-00  
Fax: (33) 4-42-53-60-01

1150 East Cheyenne Mtn. Blvd.  
Colorado Springs, CO 80906, USA  
Tel: 1(719) 576-3300  
Fax: 1(719) 540-1759

Scottish Enterprise Technology Park  
Maxwell Building  
East Kilbride G75 0QR, Scotland  
Tel: (44) 1355-803-000  
Fax: (44) 1355-242-743

### RF/Automotive

Theresienstrasse 2  
Postfach 3535  
74025 Heilbronn, Germany  
Tel: (49) 71-31-67-0  
Fax: (49) 71-31-67-2340

1150 East Cheyenne Mtn. Blvd.  
Colorado Springs, CO 80906, USA  
Tel: 1(719) 576-3300  
Fax: 1(719) 540-1759

### Biometrics/Imaging/Hi-Rel MPU/ High Speed Converters/RF Datacom

Avenue de Rochepleine  
BP 123  
38521 Saint-Egreve Cedex, France  
Tel: (33) 4-76-58-30-00  
Fax: (33) 4-76-58-34-80

---

## Literature Requests

[www.atmel.com/literature](http://www.atmel.com/literature)

**Disclaimer:** Atmel Corporation makes no warranty for the use of its products, other than those expressly contained in the Company's standard warranty which is detailed in Atmel's Terms and Conditions located on the Company's web site. The Company assumes no responsibility for any errors which may appear in this document, reserves the right to change devices or specifications detailed herein at any time without notice, and does not make any commitment to update the information contained herein. No licenses to patents or other intellectual property of Atmel are granted by the Company in connection with the sale of Atmel products, expressly or by implication. Atmel's products are not authorized for use as critical components in life support devices or systems.

© Atmel Corporation 2004. All rights reserved. Atmel® and combinations thereof, AVR®, and AVR Studio® are the registered trademarks of Atmel Corporation or its subsidiaries. Microsoft®, Windows®, Windows NT®, and Windows XP® are the registered trademarks of Microsoft Corporation. Other terms and product names may be the trademarks of others



Printed on recycled paper.

2545BS-AVR-01/04