

## 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 16 MIPS Throughput at 16 MHz
  - On-chip 2-cycle Multiplier
- Non-volatile Program and Data Memories
  - 16K Bytes of In-System Self-programmable Flash
    - Endurance: 1,000 Write/Erase Cycles
    - Endurance: 10,000 Write/Erase Cycles for ATmega162U
  - Optional Boot Code Section with Independent Lock Bits
    - In-System Programming by On-chip Boot Program
    - True Read-While-Write Operation
  - 512 Bytes EEPROM
    - Endurance: 100,000 Write/Erase Cycles
  - 1K Bytes Internal SRAM
  - Up to 64K Bytes Optional External Memory Space
  - Programming Lock for Software Security
- JTAG (IEEE std. 1149.1 Compliant) Interface
  - Boundary-scan Capabilities According to the JTAG Standard
  - Extensive On-chip Debug Support
  - Programming of Flash, EEPROM, Fuses, and Lock Bits through the JTAG Interface
- Peripheral Features
  - Two 8-bit Timer/Counters with Separate Prescalers and Compare Modes
  - Two 16-bit Timer/Counters with Separate Prescalers, Compare Modes, and Capture Modes
  - Real Time Counter with Separate Oscillator
  - Six PWM Channels
  - Dual Programmable Serial USARTs
  - Master/Slave SPI Serial Interface
  - Programmable Watchdog Timer with Separate On-chip Oscillator
  - On-chip Analog Comparator
- Special Microcontroller Features
  - Power-on Reset and Programmable Brown-out Detection
  - Internal Calibrated RC Oscillator
  - External and Internal Interrupt Sources
  - Five Sleep Modes: Idle, Power-save, Power-down, Standby, and Extended Standby
- I/O and Packages
  - 35 Programmable I/O Lines
  - 40-pin PDIP, 44-lead TQFP, and 44-pad MLF
- Operating Voltages
  - 1.8 - 3.6V for ATmega162V
  - 2.4 - 4.0V for ATmega162U
  - 2.7 - 5.5V for ATmega162L
  - 4.5 - 5.5V for ATmega162
- Speed Grades
  - 0 - 1 MHz for ATmega162V
  - 0 - 8 MHz for ATmega162L/U
  - 0 - 16 MHz for ATmega162



8-bit **AVR<sup>®</sup>**

**ATmega162**  
**ATmega162V**  
**ATmega162U**  
**ATmega162L**

**Advance  
Information**

**Summary**

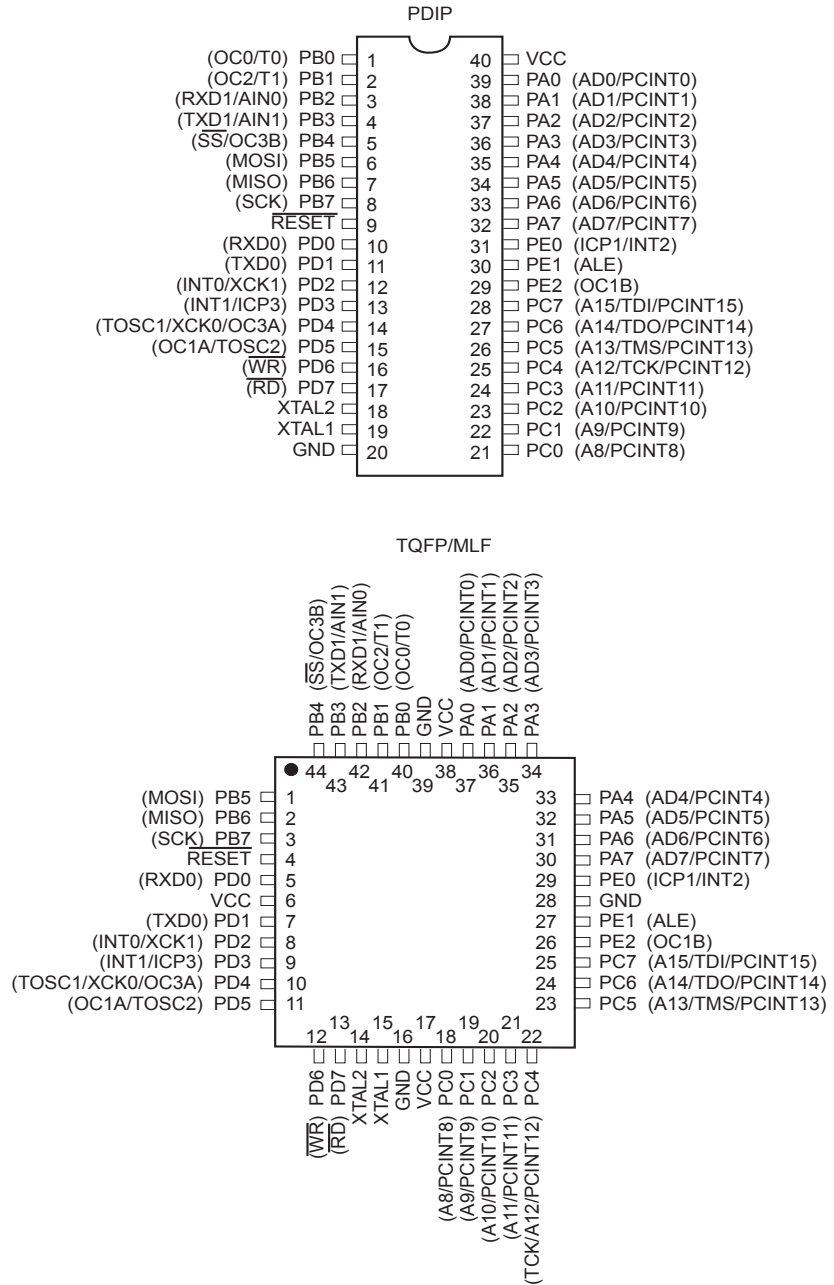
Rev. 2513CS-AVR-09/02



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 ATmega162



## Disclaimer

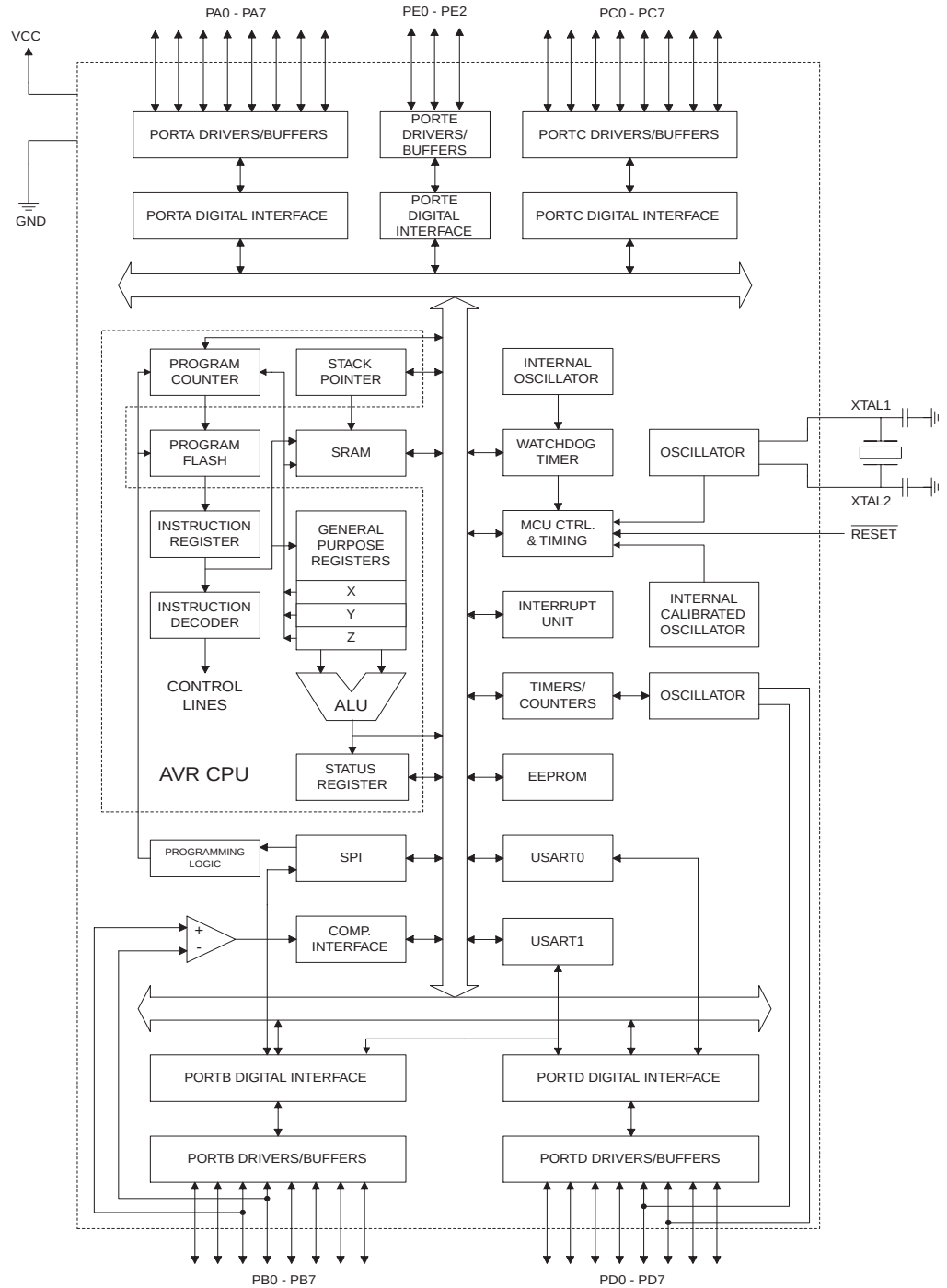
Typical values contained in this data sheet are based on simulations and characterization of other AVR microcontrollers manufactured on the same process technology. Min and Max values will be available after the device is characterized.

## Overview

The ATmega162 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 ATmega162 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 ATmega162 provides the following features: 16K bytes of In-System Programmable Flash with Read-While-Write capabilities, 512 bytes EEPROM, 1K bytes SRAM, an external memory interface, 35 general purpose I/O lines, 32 general purpose working registers, a JTAG interface for Boundary-scan, On-chip Debugging support and programming, four flexible Timer/Counters with compare modes, internal and external interrupts, two serial programmable USARTs, a programmable Watchdog Timer with Internal Oscillator, an SPI serial port, and five software selectable power saving modes. The Idle mode stops the CPU while allowing the SRAM, Timer/Counters, 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. 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. In Extended Standby mode, both the main Oscillator and the Asynchronous Timer continue to run.

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 ATmega162 is a powerful microcontroller that provides a highly flexible and cost effective solution to many embedded control applications.

The ATmega162 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.

### **ATmega161 and ATmega162 Compatibility**

The ATmega162 is a highly complex microcontroller where the number of I/O locations supersedes the 64 I/O locations reserved in the AVR instruction set. To ensure backward compatibility with the ATmega161, all I/O locations present in ATmega161 have the same locations in ATmega162. Some additional I/O locations are added in an Extended I/O space starting from 0x60 to 0xFF, (i.e., in the ATmega162 internal RAM space). These locations can be reached by using LD/LDS/LDD and ST/STS/STD instructions only, not by using IN and OUT instructions. The relocation of the internal RAM space may still be a problem for ATmega161 users. Also, the increased number of Interrupt Vectors might be a problem if the code uses absolute addresses. To solve these problems, an ATmega161 compatibility mode can be selected by programming the fuse M161C. In this mode, none of the functions in the Extended I/O space are in use, so the internal RAM is located as in ATmega161. Also, the Extended Interrupt Vectors are removed. The ATmega162 is 100% pin compatible with ATmega161, and can replace the ATmega161 on current Printed Circuit Boards. However, the location of Fuse bits and the electrical characteristics differs between the two devices.

**ATmega161 Compatibility Mode**

Programming the M161C will change the following functionality:

- The extended I/O map will be configured as internal RAM once the M161C Fuse is programmed.
- The timed sequence for changing the Watchdog Time-out period is disabled. See “Timed Sequences for Changing the Configuration of the Watchdog Timer” on page 55 for details.
- The double buffering of the USART Receive Registers is disabled. See “AVR USART vs. AVR UART – Compatibility” on page 166 for details.
- Pin change interrupts are not supported (Control Registers are located in Extended I/O).
- One 16 bits Timer/Counter (Timer/Counter1) only. Timer/Counter3 is not accessible.

Note that the shared UBRRHI Register in ATmega161 is split into two separate registers in ATmega162, UBRR0H and UBRR1H. The location of these registers will not be affected by the ATmega161 compatibility fuse.

**Pin Descriptions****VCC**

Digital supply voltage

**GND**

Ground

**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. When pins PA0 to PA7 are used as inputs and are externally pulled low, they will source current if the internal 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 ATmega162 as listed on page 71.

**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 also serves the functions of various special features of the ATmega162 as listed on page 71.

**Port C (PC7..PC0)**

Port C is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port C 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. If the JTAG interface is enabled, the pull-up resistors on pins PC7(TDI), PC5(TMS) and PC4(TCK) will be activated even if a Reset occurs.

Port C also serves the functions of the JTAG interface and other special features of the ATmega162 as listed on page 74.

## Port D (PD7..PD0)

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.

Port D also serves the functions of various special features of the ATmega162 as listed on page 77.

## Port E (PE2..PE0)

Port E is an 3-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port E output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port E pins that are externally pulled low will source current if the pull-up resistors are activated. The Port E pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port E also serves the functions of various special features of the ATmega162 as listed on page 80.

## 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 Table 18 on page 47. Shorter pulses are not guaranteed to generate a reset.

## XTAL1

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

## XTAL2

Output from the Inverting Oscillator amplifier.

## About Code Examples

This documentation contains simple code examples that briefly show how to use various parts of the device. These code examples assume that the part specific header file is included before compilation. Be aware that not all C compiler vendors include bit definitions in the header files and interrupt handling in C is compiler dependent. Please confirm with the C compiler documentation for more details.

## **ATmega162 Typical Characteristics – Preliminary Data**

The following charts show typical behavior. These figures are not tested during manufacturing. All current consumption measurements are performed with all I/O pins configured as inputs and with internal pull-ups enabled. A sine wave generator with rail-to-rail output is used as clock source. The CKSEL Fuses are programmed to select external clock.

The power consumption in Power-down mode is independent of clock selection.

The current consumption is a function of several factors such as: Operating voltage, operating frequency, loading of I/O pins, switching rate of I/O pins, code executed and ambient temperature. The dominating factors are operating voltage and frequency.

The current drawn from capacitive loaded pins may be estimated (for one pin) as  $C_L \cdot V_{CC} \cdot f$  where  $C_L$  = load capacitance,  $V_{CC}$  = operating voltage and  $f$  = average switching frequency of I/O pin.

The parts are characterized at frequencies higher than test limits. Parts are not guaranteed to function properly at frequencies higher than the ordering code indicates.

The difference between current consumption in Power-down mode with Watchdog Timer enabled and Power-down mode with Watchdog Timer disabled represents the differential current drawn by the Watchdog Timer.

## Register Summary

| Address | Name     | Bit 7                                                | Bit 6   | Bit 5   | Bit 4   | Bit 3   | Bit 2   | Bit 1  | Bit 0  | Page |
|---------|----------|------------------------------------------------------|---------|---------|---------|---------|---------|--------|--------|------|
| (0xFF)  | Reserved | –                                                    | –       | –       | –       | –       | –       | –      | –      |      |
| ...     | 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)  | TCCR3A   | COM3A1                                               | COM3A0  | COM3B1  | COM3B0  | FOC3A   | FOC3B   | WGM31  | WGM30  | 130  |
| (0x8A)  | TCCR3B   | ICNC3                                                | ICES3   | –       | WGM33   | WGM32   | CS32    | CS31   | CS30   | 127  |
| (0x89)  | TCNT3H   | Timer/Counter3 – Counter Register High Byte          |         |         |         |         |         |        |        | 132  |
| (0x88)  | TCNT3L   | Timer/Counter3 – Counter Register Low Byte           |         |         |         |         |         |        |        | 132  |
| (0x87)  | OCR3AH   | Timer/Counter3 – Output Compare Register A High Byte |         |         |         |         |         |        |        | 132  |
| (0x86)  | OCR3AL   | Timer/Counter3 – Output Compare Register A Low Byte  |         |         |         |         |         |        |        | 132  |
| (0x85)  | OCR3BH   | Timer/Counter3 – Output Compare Register B High Byte |         |         |         |         |         |        |        | 132  |
| (0x84)  | OCR3BL   | Timer/Counter3 – Output Compare Register B Low Byte  |         |         |         |         |         |        |        | 132  |
| (0x83)  | Reserved | –                                                    | –       | –       | –       | –       | –       | –      | –      |      |
| (0x82)  | Reserved | –                                                    | –       | –       | –       | –       | –       | –      | –      |      |
| (0x81)  | ICR3H    | Timer/Counter3 – Input Capture Register High Byte    |         |         |         |         |         |        |        | 133  |
| (0x80)  | ICR3L    | Timer/Counter3 – Input Capture Register Low Byte     |         |         |         |         |         |        |        | 133  |
| (0x7F)  | Reserved | –                                                    | –       | –       | –       | –       | –       | –      | –      |      |
| (0x7E)  | Reserved | –                                                    | –       | –       | –       | –       | –       | –      | –      |      |
| (0x7D)  | ETIMSK   | –                                                    | –       | TICIE3  | OCIE3A  | OCIE3B  | TOIE3   | –      | –      | 134  |
| (0x7C)  | ETIFR    | –                                                    | –       | ICF3    | OCF3A   | OCF3B   | TOV3    | –      | –      | 135  |
| (0x7B)  | Reserved | –                                                    | –       | –       | –       | –       | –       | –      | –      |      |
| (0x7A)  | Reserved | –                                                    | –       | –       | –       | –       | –       | –      | –      |      |
| (0x79)  | Reserved | –                                                    | –       | –       | –       | –       | –       | –      | –      |      |
| (0x78)  | Reserved | –                                                    | –       | –       | –       | –       | –       | –      | –      |      |
| (0x77)  | Reserved | –                                                    | –       | –       | –       | –       | –       | –      | –      |      |
| (0x76)  | Reserved | –                                                    | –       | –       | –       | –       | –       | –      | –      |      |
| (0x75)  | Reserved | –                                                    | –       | –       | –       | –       | –       | –      | –      |      |
| (0x74)  | Reserved | –                                                    | –       | –       | –       | –       | –       | –      | –      |      |
| (0x73)  | Reserved | –                                                    | –       | –       | –       | –       | –       | –      | –      |      |
| (0x72)  | Reserved | –                                                    | –       | –       | –       | –       | –       | –      | –      |      |
| (0x71)  | Reserved | –                                                    | –       | –       | –       | –       | –       | –      | –      |      |
| (0x70)  | Reserved | –                                                    | –       | –       | –       | –       | –       | –      | –      |      |
| (0x6F)  | Reserved | –                                                    | –       | –       | –       | –       | –       | –      | –      |      |
| (0x6E)  | Reserved | –                                                    | –       | –       | –       | –       | –       | –      | –      |      |
| (0x6D)  | Reserved | –                                                    | –       | –       | –       | –       | –       | –      | –      |      |
| (0x6C)  | PCMSK1   | PCINT15                                              | PCINT14 | PCINT13 | PCINT12 | PCINT11 | PCINT10 | PCINT9 | PCINT8 | 87   |
| (0x6B)  | PCMSK0   | PCINT7                                               | PCINT6  | PCINT5  | PCINT4  | PCINT3  | PCINT2  | PCINT1 | PCINT0 | 87   |
| (0x6A)  | Reserved | –                                                    | –       | –       | –       | –       | –       | –      | –      |      |
| (0x69)  | Reserved | –                                                    | –       | –       | –       | –       | –       | –      | –      |      |
| (0x68)  | Reserved | –                                                    | –       | –       | –       | –       | –       | –      | –      |      |
| (0x67)  | Reserved | –                                                    | –       | –       | –       | –       | –       | –      | –      |      |
| (0x66)  | Reserved | –                                                    | –       | –       | –       | –       | –       | –      | –      |      |
| (0x65)  | Reserved | –                                                    | –       | –       | –       | –       | –       | –      | –      |      |
| (0x64)  | Reserved | –                                                    | –       | –       | –       | –       | –       | –      | –      |      |
| (0x63)  | Reserved | –                                                    | –       | –       | –       | –       | –       | –      | –      |      |
| (0x62)  | Reserved | –                                                    | –       | –       | –       | –       | –       | –      | –      |      |
| (0x61)  | CLKPR    | CLKPCE                                               | –       | –       | –       | CLKPS3  | CLKPS2  | CLKPS1 | CLKPS0 | 40   |



| Address                                   | Name     | Bit 7                                                | Bit 6  | Bit 5  | Bit 4  | Bit 3  | Bit 2       | Bit 1  | Bit 0  | Page          |
|-------------------------------------------|----------|------------------------------------------------------|--------|--------|--------|--------|-------------|--------|--------|---------------|
| (0x60)                                    | Reserved | –                                                    | –      | –      | –      | –      | –           | –      | –      |               |
| 0x3F (0x5F)                               | SREG     | I                                                    | T      | H      | S      | V      | N           | Z      | C      | 8             |
| 0x3E (0x5E)                               | SPH      | SP15                                                 | SP14   | SP13   | SP12   | SP11   | SP10        | SP9    | SP8    | 11            |
| 0x3D (0x5D)                               | SPL      | SP7                                                  | SP6    | SP5    | SP4    | SP3    | SP2         | SP1    | SP0    | 11            |
| 0x3C <sup>(2)</sup> (0x5C) <sup>(2)</sup> | UBRR1H   | URSEL1                                               |        |        |        |        | UBRR1[11:8] |        |        | 188           |
|                                           | UCSR1C   | URSEL1                                               | UMSEL1 | UPM11  | UPM10  | USBS1  | UCSZ11      | UCSZ10 | UCPOL1 | 187           |
| 0x3B (0x5B)                               | GICR     | INT1                                                 | INT0   | INT2   | PCIE1  | PCIE0  | –           | IVSEL  | IVCE   | 60, 85        |
| 0x3A (0x5A)                               | GIFR     | INTF1                                                | INTF0  | INTF2  | PCIF1  | PCIF0  | –           | –      | –      | 86            |
| 0x39 (0x59)                               | TIMSK    | TOIE1                                                | OCIE1A | OCIE1B | OCIE2  | TICIE1 | TOIE2       | TOIE0  | OCIE0  | 101, 133, 153 |
| 0x38 (0x58)                               | TIFR     | TOV1                                                 | OCF1A  | OCF1B  | OCF2   | ICF1   | TOV2        | TOV0   | OCF0   | 102, 135, 154 |
| 0x37 (0x57)                               | SPMCR    | SPMIE                                                | RWWSB  | –      | RWWSRE | BLBSET | PGWRT       | PGERS  | SPMEN  | 219           |
| 0x36 (0x56)                               | EMUCUCR  | SM0                                                  | SRL2   | SRL1   | SRL0   | SRW01  | SRW00       | SRW11  | ISC2   | 28,43,84      |
| 0x35 (0x55)                               | MCUCR    | SRE                                                  | SRW10  | SE     | SM1    | ISC11  | ISC10       | ISC01  | ISC00  | 28,42,83      |
| 0x34 (0x54)                               | MCUCSR   | JTD                                                  | –      | SM2    | JTRF   | WDRF   | BORF        | EXTRF  | PORF   | 42,50,205     |
| 0x33 (0x53)                               | TCCR0    | FOC0                                                 | WGM00  | COM01  | COM00  | WGM01  | CS02        | CS01   | CS00   | 99            |
| 0x32 (0x52)                               | TCNT0    | Timer/Counter0 (8 Bits)                              |        |        |        |        |             |        |        | 101           |
| 0x31 (0x51)                               | OCR0     | Timer/Counter0 Output Compare Register               |        |        |        |        |             |        |        | 101           |
| 0x30 (0x50)                               | SFIOR    | TSM                                                  | XMBK   | XMM2   | XMM1   | XMM0   | PUD         | PSR2   | PSR310 | 31,69,104,155 |
| 0x2F (0x4F)                               | TCCR1A   | COM1A1                                               | COM1A0 | COM1B1 | COM1B0 | FOC1A  | FOC1B       | WGM11  | WGM10  | 127           |
| 0x2E (0x4E)                               | TCCR1B   | ICNC1                                                | ICES1  | –      | WGM13  | WGM12  | CS12        | CS11   | CS10   | 130           |
| 0x2D (0x4D)                               | TCNT1H   | Timer/Counter1 – Counter Register High Byte          |        |        |        |        |             |        |        | 132           |
| 0x2C (0x4C)                               | TCNT1L   | Timer/Counter1 – Counter Register Low Byte           |        |        |        |        |             |        |        | 132           |
| 0x2B (0x4B)                               | OCR1AH   | Timer/Counter1 – Output Compare Register A High Byte |        |        |        |        |             |        |        | 132           |
| 0x2A (0x4A)                               | OCR1AL   | Timer/Counter1 – Output Compare Register A Low Byte  |        |        |        |        |             |        |        | 132           |
| 0x29 (0x49)                               | OCR1BH   | Timer/Counter1 – Output Compare Register B High Byte |        |        |        |        |             |        |        | 132           |
| 0x28 (0x48)                               | OCR1BL   | Timer/Counter1 – Output Compare Register B Low Byte  |        |        |        |        |             |        |        | 132           |
| 0x27 (0x47)                               | TCCR2    | FOC2                                                 | WGM20  | COM21  | COM20  | WGM21  | CS22        | CS21   | CS20   | 148           |
| 0x26 (0x46)                               | ASSR     | –                                                    | –      | –      | –      | AS2    | TCO2UB      | OCR2UB | TCR2UB | 151           |
| 0x25 (0x45)                               | ICR1H    | Timer/Counter1 – Input Capture Register High Byte    |        |        |        |        |             |        |        | 133           |
| 0x24 (0x44)                               | ICR1L    | Timer/Counter1 – Input Capture Register Low Byte     |        |        |        |        |             |        |        | 133           |
| 0x23 (0x43)                               | TCNT2    | Timer/Counter2 (8 Bits)                              |        |        |        |        |             |        |        | 150           |
| 0x22 (0x42)                               | OCR2     | Timer/Counter2 Output Compare Register               |        |        |        |        |             |        |        | 150           |
| 0x21 (0x41)                               | WDTCSR   | –                                                    | –      | –      | WDCE   | WDE    | WDP2        | WDP1   | WDP0   | 52            |
| 0x20 <sup>(2)</sup> (0x40) <sup>(2)</sup> | UBRR0H   | URSEL0                                               | –      | –      | –      |        | UBRR0[11:8] |        |        | 188           |
|                                           | UCSR0C   | URSEL0                                               | UMSEL0 | UPM01  | UPM00  | USBS0  | UCSZ01      | UCSZ00 | UCPOL0 | 187           |
| 0x1F (0x3F)                               | EEARH    | –                                                    | –      | –      | –      | –      | –           | –      | EEAR8  | 18            |
| 0x1E (0x3E)                               | EEARL    | EEPROM Address Register Low Byte                     |        |        |        |        |             |        |        | 18            |
| 0x1D (0x3D)                               | EEDR     | EEPROM Data Register                                 |        |        |        |        |             |        |        | 19            |
| 0x1C (0x3C)                               | EEDR     | –                                                    | –      | –      | –      | EERIE  | EEMWE       | EWE    | EERE   | 19            |
| 0x1B (0x3B)                               | PORTA    | PORTA7                                               | PORTA6 | PORTA5 | PORTA4 | PORTA3 | PORTA2      | PORTA1 | PORTA0 | 81            |
| 0x1A (0x3A)                               | DDRA     | DDA7                                                 | DDA6   | DDA5   | DDA4   | DDA3   | DDA2        | DDA1   | DDA0   | 81            |
| 0x19 (0x39)                               | PINA     | PINA7                                                | PINA6  | PINA5  | PINA4  | PINA3  | PINA2       | PINA1  | PINA0  | 81            |
| 0x18 (0x38)                               | PORTB    | PORTB7                                               | PORTB6 | PORTB5 | PORTB4 | PORTB3 | PORTB2      | PORTB1 | PORTB0 | 81            |
| 0x17 (0x37)                               | DDRB     | DDB7                                                 | DDB6   | DDB5   | DDB4   | DDB3   | DDB2        | DDB1   | DDB0   | 81            |
| 0x16 (0x36)                               | PINB     | PINB7                                                | PINB6  | PINB5  | PINB4  | PINB3  | PINB2       | PINB1  | PINB0  | 81            |
| 0x15 (0x35)                               | PORTC    | PORTC7                                               | PORTC6 | PORTC5 | PORTC4 | PORTC3 | PORTC2      | PORTC1 | PORTC0 | 81            |
| 0x14 (0x34)                               | DDRC     | DDC7                                                 | DDC6   | DDC5   | DDC4   | DDC3   | DDC2        | DDC1   | DDC0   | 81            |
| 0x13 (0x33)                               | PINC     | PINC7                                                | PINC6  | PINC5  | PINC4  | PINC3  | PINC2       | PINC1  | PINC0  | 82            |
| 0x12 (0x32)                               | PORTD    | PORTD7                                               | PORTD6 | PORTD5 | PORTD4 | PORTD3 | PORTD2      | PORTD1 | PORTD0 | 82            |
| 0x11 (0x31)                               | DDRD     | DDD7                                                 | DDD6   | DDD5   | DDD4   | DDD3   | DDD2        | DDD1   | DDD0   | 82            |
| 0x10 (0x30)                               | PIND     | PIND7                                                | PIND6  | PIND5  | PIND4  | PIND3  | PIND2       | PIND1  | PIND0  | 82            |
| 0x0F (0x2F)                               | SPDR     | SPI Data Register                                    |        |        |        |        |             |        |        | 162           |
| 0x0E (0x2E)                               | SPSR     | SPIF                                                 | WCOL   | –      | –      | –      | –           | –      | SPI2X  | 162           |
| 0x0D (0x2D)                               | SPCR     | SPIE                                                 | SPE    | DORD   | MSTR   | CPOL   | CPHA        | SPR1   | SPR0   | 160           |
| 0x0C (0x2C)                               | UDR0     | USART0 I/O Data Register                             |        |        |        |        |             |        |        | 184           |
| 0x0B (0x2B)                               | UCSR0A   | RXC0                                                 | TXC0   | UDRE0  | FE0    | DOR0   | UPE0        | U2X0   | MPCM0  | 184           |
| 0x0A (0x2A)                               | UCSR0B   | RXCIE0                                               | TXCIE0 | UDRIE0 | RXEN0  | TXEN0  | UCSZ02      | RXB80  | TXB80  | 185           |
| 0x09 (0x29)                               | UBRR0L   | USART0 Baud Rate Register Low Byte                   |        |        |        |        |             |        |        | 188           |
| 0x08 (0x28)                               | ACSR     | ACD                                                  | ACBG   | ACO    | ACI    | ACIE   | ACIC        | ACIS1  | ACIS0  | 193           |
| 0x07 (0x27)                               | PORTE    | –                                                    | –      | –      | –      | –      | PORTE2      | PORTE1 | PORTE0 | 82            |
| 0x06 (0x26)                               | DDRE     | –                                                    | –      | –      | –      | –      | DDE2        | DDE1   | DDE0   | 82            |
| 0x05 (0x25)                               | PINE     | –                                                    | –      | –      | –      | –      | PINE2       | PINE1  | PINE0  | 82            |
| 0x04 <sup>(1)</sup> (0x24) <sup>(1)</sup> | OSCCAL   | Oscillator Calibration Register                      |        |        |        |        |             |        |        | 38            |
|                                           | OCDR     | On-chip Debug Register                               |        |        |        |        |             |        |        | 200           |
| 0x03 (0x23)                               | UDR1     | USART1 I/O Data Register                             |        |        |        |        |             |        |        | 184           |
| 0x02 (0x22)                               | UCSR1A   | RXC1                                                 | TXC1   | UDRE1  | FE1    | DOR1   | UPE1        | U2X1   | MPCM1  | 184           |

| Address     | Name   | Bit 7                              | Bit 6  | Bit 5  | Bit 4 | Bit 3 | Bit 2  | Bit 1 | Bit 0 | Page |
|-------------|--------|------------------------------------|--------|--------|-------|-------|--------|-------|-------|------|
| 0x01 (0x21) | UCSR1B | RXCIE1                             | TXCIE1 | UDRIE1 | RXEN1 | TXEN1 | UCSZ12 | RXB81 | TXB81 | 185  |
| 0x00 (0x20) | UBRR1L | USART1 Baud Rate Register Low Byte |        |        |       |       |        |       |       | 188  |

- Notes:
1. When the OCDEN Fuse is unprogrammed, the OSCCAL Register is always accessed on this address. Refer to the debugger specific documentation for details on how to use the OCDR Register.
  2. Refer to the USART description for details on how to access UBRRH and UCSRC.
  3. For compatibility with future devices, reserved bits should be written to zero if accessed. Reserved I/O memory addresses should never be written.
  4. Some of the status flags are cleared by writing a logical one to them. Note that the CBI and SBI instructions will operate on all bits in the I/O Register, writing a one back into any flag read as set, thus clearing the flag. The CBI and SBI instructions work with registers 0x00 to 0x1F only.

## 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                                     | RdL, 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       |
| SBW                                      | RdL, 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       |
| IJMP                                     |          | 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   |
| 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>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       |
| POP                                  | Rd       | Pop Register from Stack          | Rd ← STACK                               | None       | 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       |

| Mnemonics                       | Operands | Description                   | Operation                                | Flags | #Clocks |
|---------------------------------|----------|-------------------------------|------------------------------------------|-------|---------|
| CLH                             |          | Clear Half Carry Flag in SREG | $H \leftarrow 0$                         | H     | 1       |
| <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     |

## Ordering Information

| Speed (MHz) | Power Supply | Ordering Code                                      | Package             | Operation Range               |
|-------------|--------------|----------------------------------------------------|---------------------|-------------------------------|
| 1           | 1.8 - 3.6V   | ATmega162V-1AC<br>ATmega162V-1PC<br>ATmega162V-1MC | 44A<br>40P6<br>44M1 | Commercial<br>(0°C to 70°C)   |
| 8           | 2.4 - 4.0V   | ATmega162U-8AC<br>ATmega162U-8PC<br>ATmega162U-8MC | 44A<br>40P6<br>44M1 | Commercial<br>(0°C to 70°C)   |
| 8           | 2.7 - 5.5V   | ATmega162L-8AC<br>ATmega162L-8PC<br>ATmega162L-8MC | 44A<br>40P6<br>44M1 | Commercial<br>(0°C to 70°C)   |
|             |              | ATmega162L-8AI<br>ATmega162L-8PI<br>ATmega162L-8MI | 44A<br>40P6<br>44M1 | Industrial<br>(-40°C to 85°C) |
| 16          | 4.5 - 5.5V   | ATmega162-16AC<br>ATmega162-16PC<br>ATmega162-16MC | 44A<br>40P6<br>44M1 | Commercial<br>(0°C to 70°C)   |
|             |              | ATmega162-16AI<br>ATmega162-16PI<br>ATmega162-16MI | 44A<br>40P6<br>44M1 | Industrial<br>(-40°C to 85°C) |

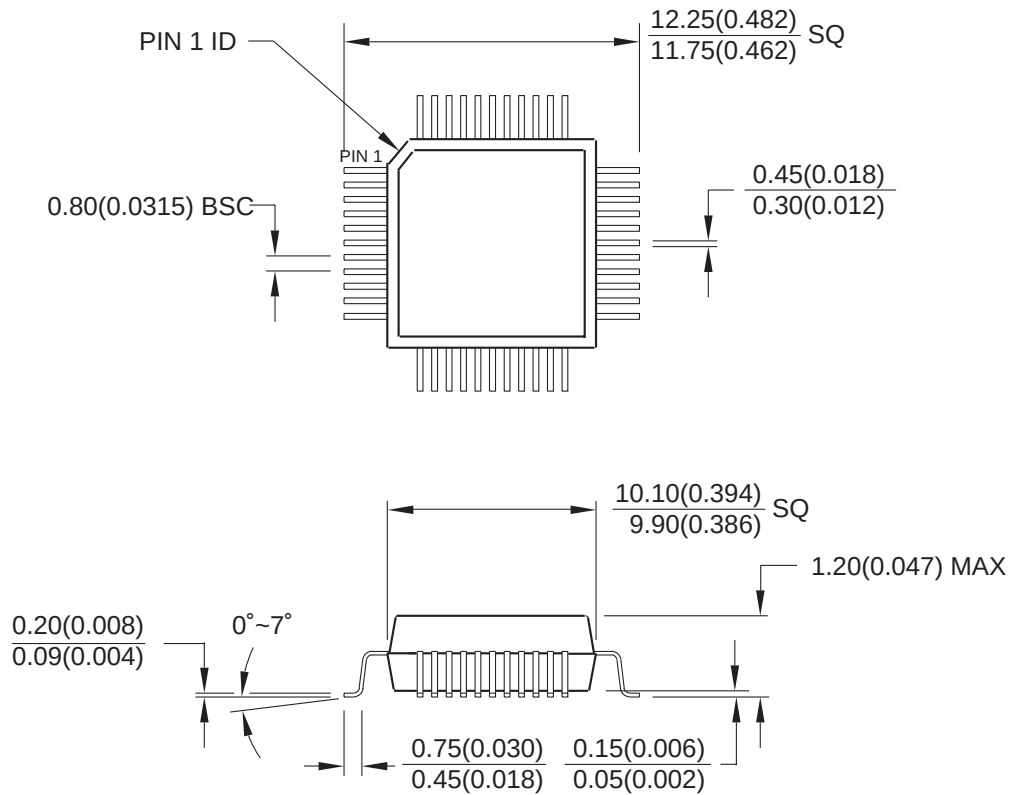
Note: 1. This device can also be supplied in wafer form. Please contact your local Atmel sales office for detailed ordering information and minimum quantities.

| Package Type |                                                                                 |
|--------------|---------------------------------------------------------------------------------|
| <b>44A</b>   | 44-lead, Thin (1.0 mm) Plastic Gull Wing Quad Flat Package (TQFP)               |
| <b>40P6</b>  | 40-pin, 0.600" Wide, Plastic Dual Inline Package (PDIP)                         |
| <b>44M1</b>  | 44-pad, 7 x 7 x 1.0 mm body, lead pitch 0.50 mm, Micro Lead Frame Package (MLF) |

## Packaging Information

### 44A

44-lead, Thin (1.0mm) Plastic Quad Flat Package  
(TQFP), 10x10mm body, 2.0mm footprint, 0.8mm pitch.  
Dimension in Millimeters and (Inches)\*  
JEDEC STANDARD MS-026 ACB

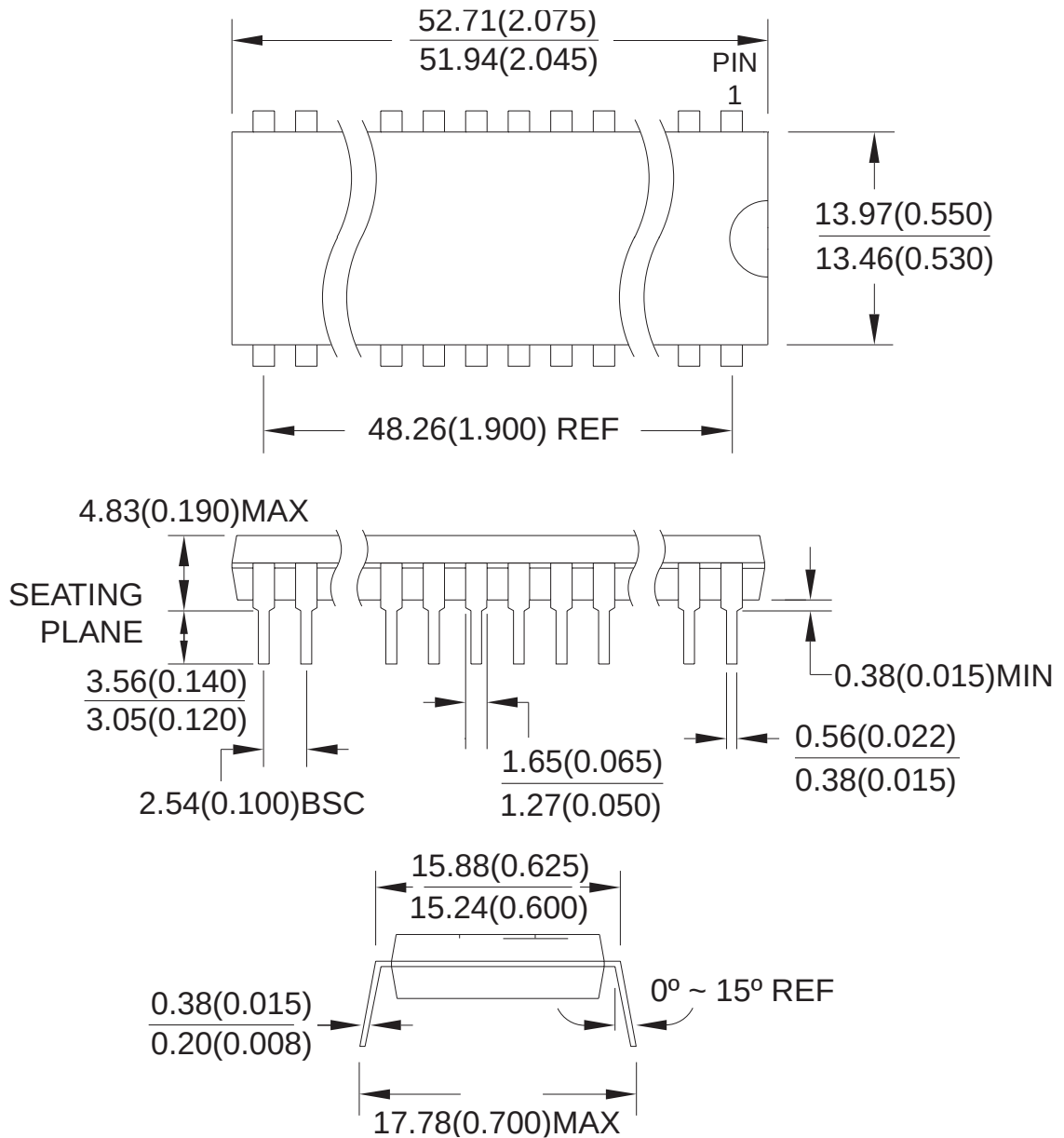


\*Controlling dimension: millimeter

REV. A 04/11/2001

## 40P6

40-lead, Plastic Dual Inline  
Package (PDIP), 0.600" wide  
Dimension in Millimeters and (Inches)\*  
JEDEC STANDARD MS-011 AC

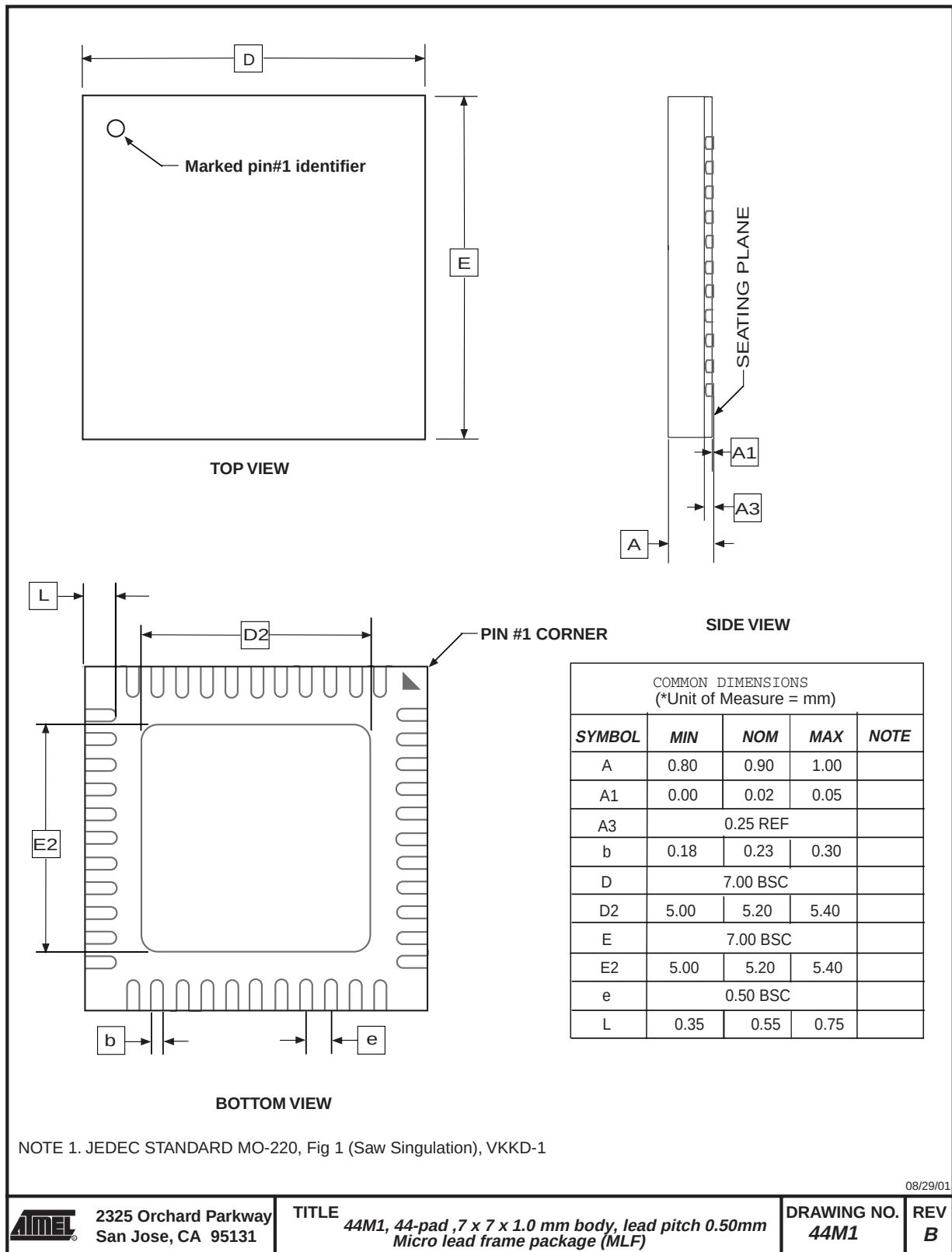


\*Controlling dimension: Inches

REV. A 04/11/2001



44M1



## **Data Sheet Change Log for ATmega162**

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.

### **Changes from Rev. 2513A-05/02 to Rev. 2513B-09/02**

#### **1. Added information for ATmega162U.**

Information about ATmega162U included in "Features" on page 1, Table 19, "BODLEVEL Fuse Coding," on page 49, and "Ordering Information" on page 14.

### **Changes from Rev. 2513B-09/02 to Rev. 2513C-09/02**

#### **1. Changed the Endurance on the Flash to 10,000 Write/Erase Cycles.**



## **Atmel Headquarters**

### ***Corporate Headquarters***

2325 Orchard Parkway  
San Jose, CA 95131  
TEL 1(408) 441-0311  
FAX 1(408) 487-2600

### ***Europe***

Atmel Sarl  
Route des Arsenaux 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 Tsimhatsui  
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  
TEL 1(408) 441-0311  
FAX 1(408) 436-4314

### ***Microcontrollers***

2325 Orchard Parkway  
San Jose, CA 95131  
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  
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  
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

---

### ***e-mail***

literature@atmel.com

### ***Web Site***

<http://www.atmel.com>

## **© Atmel Corporation 2002.**

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®, AVR®, and AVR Studio® are the registered trademarks of Atmel.

Microsoft®, Windows®, and Windows NT® are trademarks of Microsoft Corporation.

Other terms and product names may be the trademarks of others.



Printed on recycled paper.

2513CS-AVR-09/01

OM