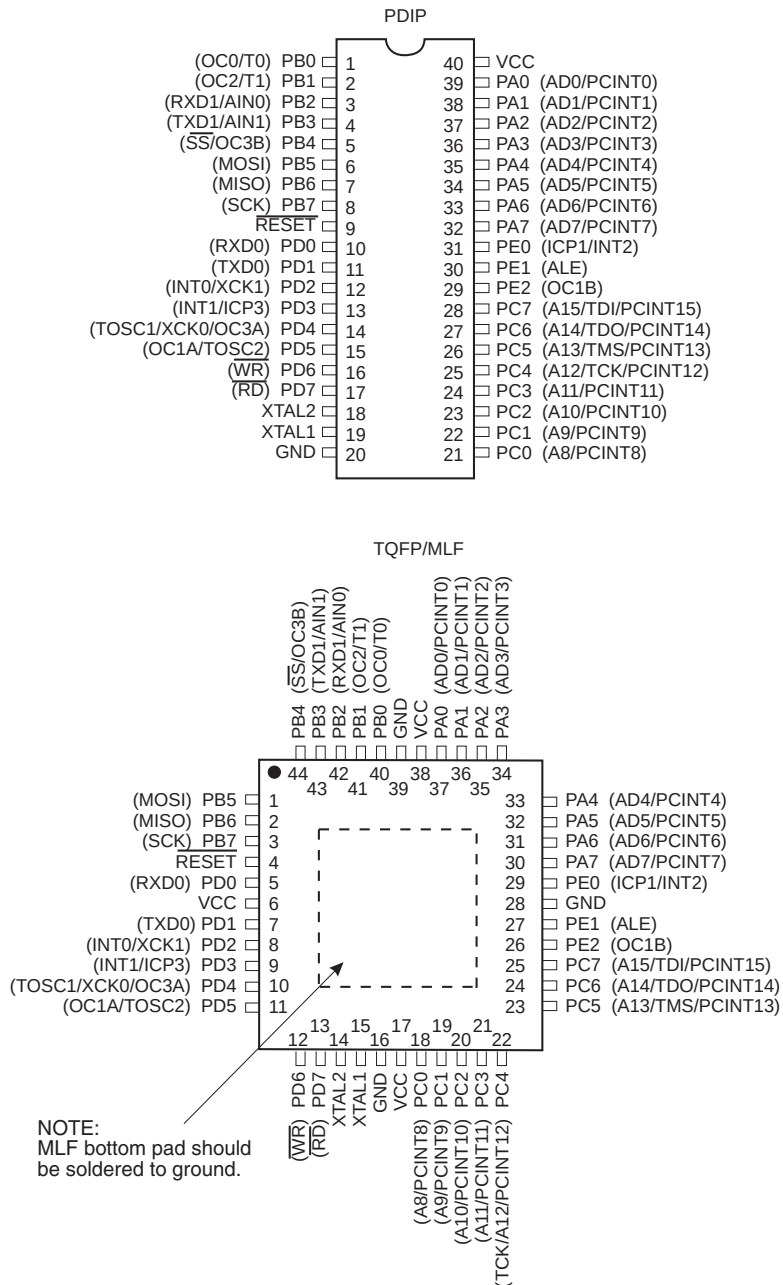


## Pin Configurations

Figure 1. Pinout ATmega162



## Disclaimer

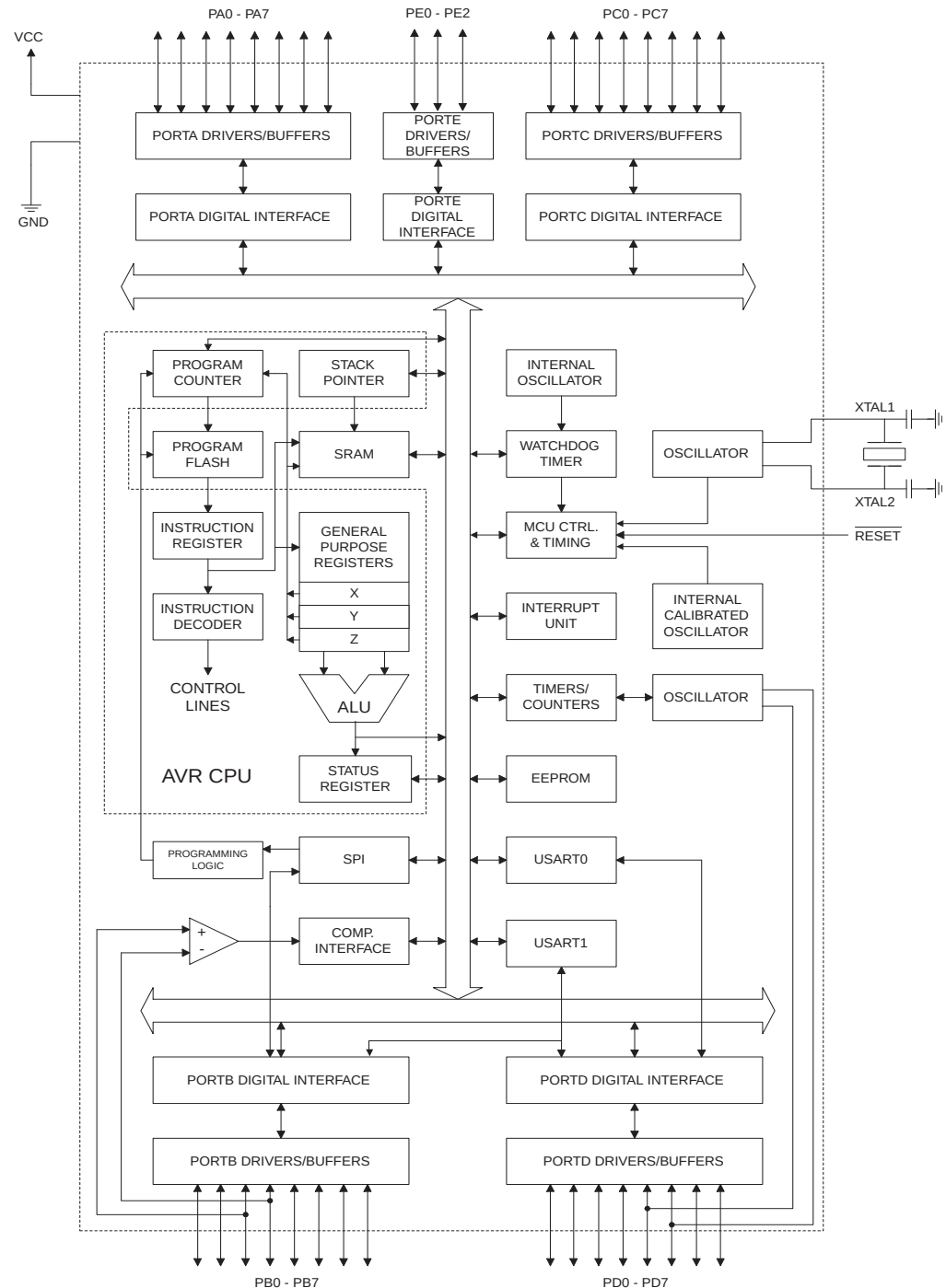
Typical values contained in this datasheet are based on simulations and characterization of other AVR microcontrollers manufactured on the same process technology. Min and Max values will be available after the device is characterized.

## Overview

The 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 back-ward 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 56](#) for details.
- The double buffering of the USART Receive Registers is disabled. See [“AVR USART vs. AVR UART – Compatibility” on page 168](#) 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 72](#).

**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 72](#).

**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 75](#).

**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 78](#).

**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 81](#).

**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 48](#). 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.

## Resources

A comprehensive set of development tools, application notes and datasheets are available for download on <http://www.atmel.com/avr>.

## Data Retention

Reliability Qualification results show that the projected data retention failure rate is much less than 1 PPM over 20 years at 85°C or 100 years at 25°C.

## 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  | 131  |
| (0x8A)  | TCCR3B   | ICNC3                                                | ICES3   | –       | WGM33   | WGM32   | CS32    | CS31   | CS30   | 128  |
| (0x89)  | TCNT3H   | Timer/Counter3 – Counter Register High Byte          |         |         |         |         |         |        |        | 133  |
| (0x88)  | TCNT3L   | Timer/Counter3 – Counter Register Low Byte           |         |         |         |         |         |        |        | 133  |
| (0x87)  | OCR3AH   | Timer/Counter3 – Output Compare Register A High Byte |         |         |         |         |         |        |        | 133  |
| (0x86)  | OCR3AL   | Timer/Counter3 – Output Compare Register A Low Byte  |         |         |         |         |         |        |        | 133  |
| (0x85)  | OCR3BH   | Timer/Counter3 – Output Compare Register B High Byte |         |         |         |         |         |        |        | 133  |
| (0x84)  | OCR3BL   | Timer/Counter3 – Output Compare Register B Low Byte  |         |         |         |         |         |        |        | 133  |
| (0x83)  | Reserved | –                                                    | –       | –       | –       | –       | –       | –      | –      |      |
| (0x82)  | Reserved | –                                                    | –       | –       | –       | –       | –       | –      | –      |      |
| (0x81)  | ICR3H    | Timer/Counter3 – Input Capture Register High Byte    |         |         |         |         |         |        |        | 134  |
| (0x80)  | ICR3L    | Timer/Counter3 – Input Capture Register Low Byte     |         |         |         |         |         |        |        | 134  |
| (0x7F)  | Reserved | –                                                    | –       | –       | –       | –       | –       | –      | –      |      |
| (0x7E)  | Reserved | –                                                    | –       | –       | –       | –       | –       | –      | –      |      |
| (0x7D)  | ETIMSK   | –                                                    | –       | TICIE3  | OCIE3A  | OCIE3B  | TOIE3   | –      | –      | 135  |
| (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 | 88   |
| (0x6B)  | PCMSK0   | PCINT7                                               | PCINT6  | PCINT5  | PCINT4  | PCINT3  | PCINT2  | PCINT1 | PCINT0 | 88   |
| (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 | 41   |

| 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      | 10            |
| 0x3E (0x5E)                               | SPH      | SP15                                                 | SP14   | SP13   | SP12   | SP11        | SP10   | SP9    | SP8    | 13            |
| 0x3D (0x5D)                               | SPL      | SP7                                                  | SP6    | SP5    | SP4    | SP3         | SP2    | SP1    | SP0    | 13            |
| 0x3C <sup>(2)</sup> (0x5C) <sup>(2)</sup> | UBRR1H   | URSEL1                                               |        |        |        | UBRR1[11:8] |        |        |        | 190           |
|                                           | UCSR1C   | URSEL1                                               | UMSEL1 | UPM11  | UPM10  | USBS1       | UCSZ11 | UCSZ10 | UCPOL1 | 189           |
| 0x3B (0x5B)                               | GICR     | INT1                                                 | INT0   | INT2   | PCIE1  | PCIE0       | –      | IVSEL  | IVCE   | 61, 86        |
| 0x3A (0x5A)                               | GIFR     | INTF1                                                | INTF0  | INTF2  | PCIF1  | PCIF0       | –      | –      | –      | 87            |
| 0x39 (0x59)                               | TIMSK    | TOIE1                                                | OCIE1A | OCIE1B | OCIE2  | TICIE1      | TOIE2  | TOIE0  | OCIE0  | 102, 134, 154 |
| 0x38 (0x58)                               | TIFR     | TOV1                                                 | OCF1A  | OCF1B  | OCF2   | ICF1        | TOV2   | TOV0   | OCF0   | 103, 135, 155 |
| 0x37 (0x57)                               | SPMCR    | SPMIE                                                | RWWSB  | –      | RWWSRE | BLBSET      | PGWRT  | PGERS  | SPMEN  | 221           |
| 0x36 (0x56)                               | EMUCR    | SM0                                                  | SRL2   | SRL1   | SRL0   | SRW01       | SRW00  | SRW11  | ISC2   | 30,44,85      |
| 0x35 (0x55)                               | MCUCR    | SRE                                                  | SRW10  | SE     | SM1    | ISC11       | ISC10  | ISC01  | ISC00  | 30,43,84      |
| 0x34 (0x54)                               | MCUCSR   | JTD                                                  | –      | SM2    | JTRF   | WDRF        | BORF   | EXTRF  | PORF   | 43,51,207     |
| 0x33 (0x53)                               | TCCR0    | FOC0                                                 | WGM00  | COM01  | COM00  | WGM01       | CS02   | CS01   | CS00   | 100           |
| 0x32 (0x52)                               | TCNT0    | Timer/Counter0 (8 Bits)                              |        |        |        |             |        |        |        | 102           |
| 0x31 (0x51)                               | OCR0     | Timer/Counter0 Output Compare Register               |        |        |        |             |        |        |        | 102           |
| 0x30 (0x50)                               | SFIOR    | TSM                                                  | XMBK   | XMM2   | XMM1   | XMM0        | PUD    | PSR2   | PSR310 | 32,70,105,156 |
| 0x2F (0x4F)                               | TCCR1A   | COM1A1                                               | COM1A0 | COM1B1 | COM1B0 | FOC1A       | FOC1B  | WGM11  | WGM10  | 128           |
| 0x2E (0x4E)                               | TCCR1B   | ICNC1                                                | ICES1  | –      | WGM13  | WGM12       | CS12   | CS11   | CS10   | 131           |
| 0x2D (0x4D)                               | TCNT1H   | Timer/Counter1 – Counter Register High Byte          |        |        |        |             |        |        |        | 133           |
| 0x2C (0x4C)                               | TCNT1L   | Timer/Counter1 – Counter Register Low Byte           |        |        |        |             |        |        |        | 133           |
| 0x2B (0x4B)                               | OCR1AH   | Timer/Counter1 – Output Compare Register A High Byte |        |        |        |             |        |        |        | 133           |
| 0x2A (0x4A)                               | OCR1AL   | Timer/Counter1 – Output Compare Register A Low Byte  |        |        |        |             |        |        |        | 133           |
| 0x29 (0x49)                               | OCR1BH   | Timer/Counter1 – Output Compare Register B High Byte |        |        |        |             |        |        |        | 133           |
| 0x28 (0x48)                               | OCR1BL   | Timer/Counter1 – Output Compare Register B Low Byte  |        |        |        |             |        |        |        | 133           |
| 0x27 (0x47)                               | TCCR2    | FOC2                                                 | WGM20  | COM21  | COM20  | WGM21       | CS22   | CS21   | CS20   | 149           |
| 0x26 (0x46)                               | ASSR     | –                                                    | –      | –      | –      | AS2         | TCN2UB | OCR2UB | TCR2UB | 152           |
| 0x25 (0x45)                               | ICR1H    | Timer/Counter1 – Input Capture Register High Byte    |        |        |        |             |        |        |        | 134           |
| 0x24 (0x44)                               | ICR1L    | Timer/Counter1 – Input Capture Register Low Byte     |        |        |        |             |        |        |        | 134           |
| 0x23 (0x43)                               | TCNT2    | Timer/Counter2 (8 Bits)                              |        |        |        |             |        |        |        | 151           |
| 0x22 (0x42)                               | OCR2     | Timer/Counter2 Output Compare Register               |        |        |        |             |        |        |        | 151           |
| 0x21 (0x41)                               | WDTCSR   | –                                                    | –      | –      | WDCE   | WDE         | WDP2   | WDP1   | WDP0   | 53            |
| 0x20 <sup>(2)</sup> (0x40) <sup>(2)</sup> | UBRR0H   | URSEL0                                               | –      | –      | –      | UBRR0[11:8] |        |        |        | 190           |
|                                           | UCSR0C   | URSEL0                                               | UMSEL0 | UPM01  | UPM00  | USBS0       | UCSZ01 | UCSZ00 | UCPOL0 | 189           |
| 0x1F (0x3F)                               | EEARH    | –                                                    | –      | –      | –      | –           | –      | –      | EEAR8  | 20            |
| 0x1E (0x3E)                               | EEARL    | EEPROM Address Register Low Byte                     |        |        |        |             |        |        |        | 20            |
| 0x1D (0x3D)                               | EEDR     | EEPROM Data Register                                 |        |        |        |             |        |        |        | 21            |
| 0x1C (0x3C)                               | EEDR     | –                                                    | –      | –      | –      | EERIE       | EEMWE  | EWE    | EERE   | 21            |
| 0x1B (0x3B)                               | PORTA    | PORTA7                                               | PORTA6 | PORTA5 | PORTA4 | PORTA3      | PORTA2 | PORTA1 | PORTA0 | 82            |
| 0x1A (0x3A)                               | DDRA     | DDA7                                                 | DDA6   | DDA5   | DDA4   | DDA3        | DDA2   | DDA1   | DDA0   | 82            |
| 0x19 (0x39)                               | PINA     | PINA7                                                | PINA6  | PINA5  | PINA4  | PINA3       | PINA2  | PINA1  | PINA0  | 82            |
| 0x18 (0x38)                               | PORTB    | PORTB7                                               | PORTB6 | PORTB5 | PORTB4 | PORTB3      | PORTB2 | PORTB1 | PORTB0 | 82            |
| 0x17 (0x37)                               | DDRB     | DDB7                                                 | DDB6   | DDB5   | DDB4   | DDB3        | DDB2   | DDB1   | DDB0   | 82            |
| 0x16 (0x36)                               | PINB     | PINB7                                                | PINB6  | PINB5  | PINB4  | PINB3       | PINB2  | PINB1  | PINB0  | 82            |
| 0x15 (0x35)                               | PORTC    | PORTC7                                               | PORTC6 | PORTC5 | PORTC4 | PORTC3      | PORTC2 | PORTC1 | PORTC0 | 82            |
| 0x14 (0x34)                               | DDRC     | DDC7                                                 | DDC6   | DDC5   | DDC4   | DDC3        | DDC2   | DDC1   | DDC0   | 82            |
| 0x13 (0x33)                               | PINC     | PINC7                                                | PINC6  | PINC5  | PINC4  | PINC3       | PINC2  | PINC1  | PINC0  | 83            |
| 0x12 (0x32)                               | PORTD    | PORTD7                                               | PORTD6 | PORTD5 | PORTD4 | PORTD3      | PORTD2 | PORTD1 | PORTD0 | 83            |
| 0x11 (0x31)                               | DDRD     | DDD7                                                 | DDD6   | DDD5   | DDD4   | DDD3        | DDD2   | DDD1   | DDD0   | 83            |
| 0x10 (0x30)                               | PIND     | PIND7                                                | PIND6  | PIND5  | PIND4  | PIND3       | PIND2  | PIND1  | PIND0  | 83            |
| 0x0F (0x2F)                               | SPDR     | SPI Data Register                                    |        |        |        |             |        |        |        | 164           |
| 0x0E (0x2E)                               | SPSR     | SPIF                                                 | WCOL   | –      | –      | –           | –      | –      | SPI2X  | 164           |
| 0x0D (0x2D)                               | SPCR     | SPIE                                                 | SPE    | DORD   | MSTR   | CPOL        | CPHA   | SPR1   | SPR0   | 162           |
| 0x0C (0x2C)                               | UDR0     | USART0 I/O Data Register                             |        |        |        |             |        |        |        | 186           |
| 0x0B (0x2B)                               | UCSR0A   | RXC0                                                 | TXC0   | UDRE0  | FE0    | DOR0        | UPE0   | U2X0   | MPCM0  | 186           |
| 0x0A (0x2A)                               | UCSR0B   | RXCIE0                                               | TXCIE0 | UDRIE0 | RXEN0  | TXEN0       | UCSZ02 | RXB80  | TXB80  | 187           |
| 0x09 (0x29)                               | UBRR0L   | USART0 Baud Rate Register Low Byte                   |        |        |        |             |        |        |        | 190           |
| 0x08 (0x28)                               | ACSR     | ACD                                                  | ACBG   | ACO    | ACI    | ACIE        | ACIC   | ACIS1  | ACIS0  | 195           |
| 0x07 (0x27)                               | PORTE    | –                                                    | –      | –      | –      | –           | PORTE2 | PORTE1 | PORTE0 | 83            |
| 0x06 (0x26)                               | DDRE     | –                                                    | –      | –      | –      | –           | DDE2   | DDE1   | DDE0   | 83            |
| 0x05 (0x25)                               | PINE     | –                                                    | –      | –      | –      | –           | PINE2  | PINE1  | PINE0  | 83            |
| 0x04 <sup>(1)</sup> (0x24) <sup>(1)</sup> | OSCCAL   | –                                                    | CAL6   | CAL5   | CAL4   | CAL3        | CAL2   | CAL1   | CAL0   | 39            |
|                                           | OCDR     | On-chip Debug Register                               |        |        |        |             |        |        |        | 202           |
| 0x03 (0x23)                               | UDR1     | USART1 I/O Data Register                             |        |        |        |             |        |        |        | 186           |
| 0x02 (0x22)                               | UCSR1A   | RXC1                                                 | TXC1   | UDRE1  | FE1    | DOR1        | UPE1   | U2X1   | MPCM1  | 186           |



| 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 | 187  |
| 0x00 (0x20) | UBRR1L | USART1 Baud Rate Register Low Byte |        |        |       |       |        |       |       | 190  |

- Notes:
1. When the OCDEN Fuse is unprogrammed, the OSCCAL Register is always accessed on this address. Refer to the debug-ger 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       |
| SBIW                                     | 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

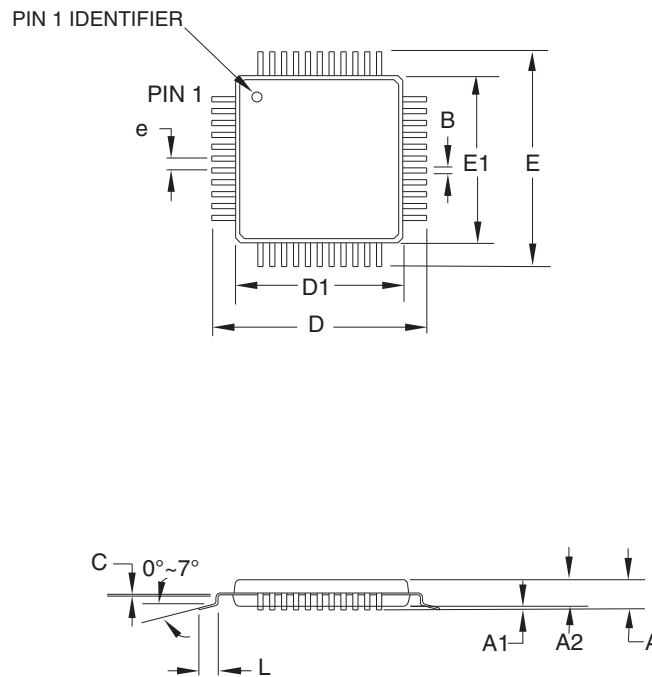
| Speed (MHz)       | Power Supply | Ordering Code <sup>(2)</sup> | Package <sup>(1)</sup> | Operation Range               |
|-------------------|--------------|------------------------------|------------------------|-------------------------------|
| 8 <sup>(3)</sup>  | 1.8 - 5.5V   | ATmega162V-8AU               | 44A                    | Industrial<br>(-40°C to 85°C) |
|                   |              | ATmega162V-8PU               | 40P6                   |                               |
|                   |              | ATmega162V-8MU               | 44M1                   |                               |
| 16 <sup>(4)</sup> | 2.7 - 5.5V   | ATmega162-16AU               | 44A                    | Industrial<br>(-40°C to 85°C) |
|                   |              | ATmega162-16PU               | 40P6                   |                               |
|                   |              | ATmega162-16MU               | 44M1                   |                               |

- Notes:
1. This device can also be supplied in wafer form. Please contact your local Atmel sales office for detailed ordering information and minimum quantities.
  2. Pb-free packaging, complies to the European Directive for Restriction of Hazardous Substances (RoHS directive). Also Halide free and fully Green.
  3. See [Figure 113 on page 266](#).
  4. See [Figure 114 on page 266](#).

| 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 (QFN/MLF) |

# Packaging Information

44A



**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      | 11.75    | 12.00 | 12.25 |        |
| D1     | 9.90     | 10.00 | 10.10 | Note 2 |
| E      | 11.75    | 12.00 | 12.25 |        |
| E1     | 9.90     | 10.00 | 10.10 | 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 ACB.
2. Dimensions D1 and E1 do not include mold protrusion. Allowable protrusion is 0.25mm per side. Dimensions D1 and E1 are maximum plastic body size dimensions including mold mismatch.
3. Lead coplanarity is 0.10mm maximum.

2010-10-20



2325 Orchard Parkway  
San Jose, CA 95131

**TITLE**

**44A**, 44-lead, 10 x 10mm body size, 1.0mm body thickness,  
0.8 mm lead pitch, thin profile plastic quad flat package (TQFP)

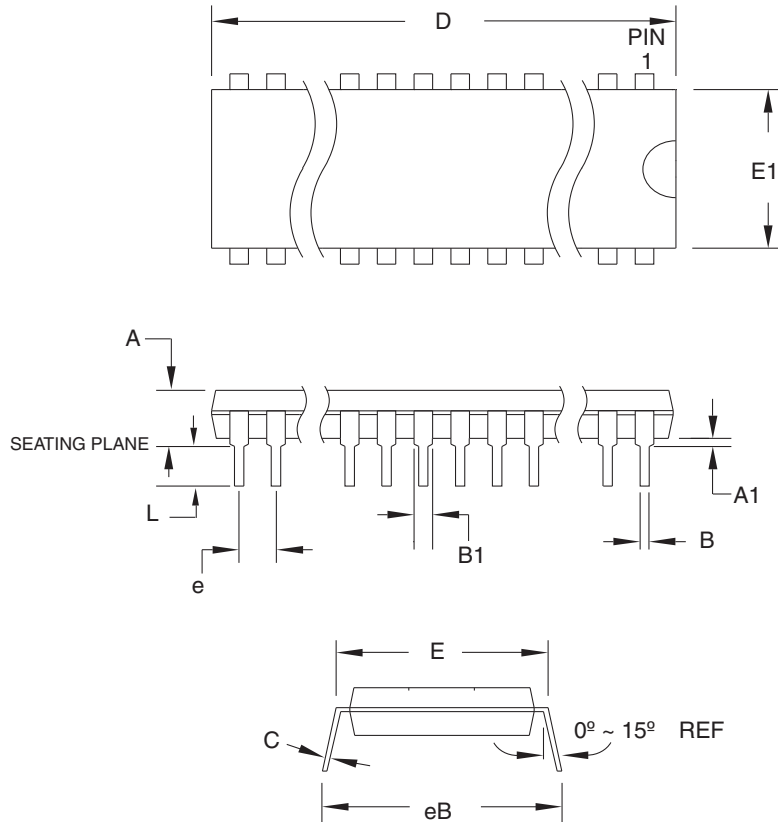
**DRAWING NO.**

44A

**REV.**

C

40P6



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

| SYMBOL | MIN       | NOM | MAX    | NOTE   |
|--------|-----------|-----|--------|--------|
| A      | —         | —   | 4.826  |        |
| A1     | 0.381     | —   | —      |        |
| D      | 52.070    | —   | 52.578 | Note 2 |
| E      | 15.240    | —   | 15.875 |        |
| E1     | 13.462    | —   | 13.970 | Note 2 |
| B      | 0.356     | —   | 0.559  |        |
| B1     | 1.041     | —   | 1.651  |        |
| L      | 3.048     | —   | 3.556  |        |
| C      | 0.203     | —   | 0.381  |        |
| eB     | 15.494    | —   | 17.526 |        |
| e      | 2.540 TYP |     |        |        |

Notes:

1. This package conforms to JEDEC reference MS-011, Variation AC.
2. Dimensions D and E1 do not include mold Flash or Protrusion.  
Mold Flash or Protrusion shall not exceed 0.25mm (0.010").

09/28/01



2325 Orchard Parkway  
San Jose, CA 95131

**TITLE**

**40P6**, 40-lead (0.600"/15.24mm Wide) Plastic Dual  
Inline Package (PDIP)

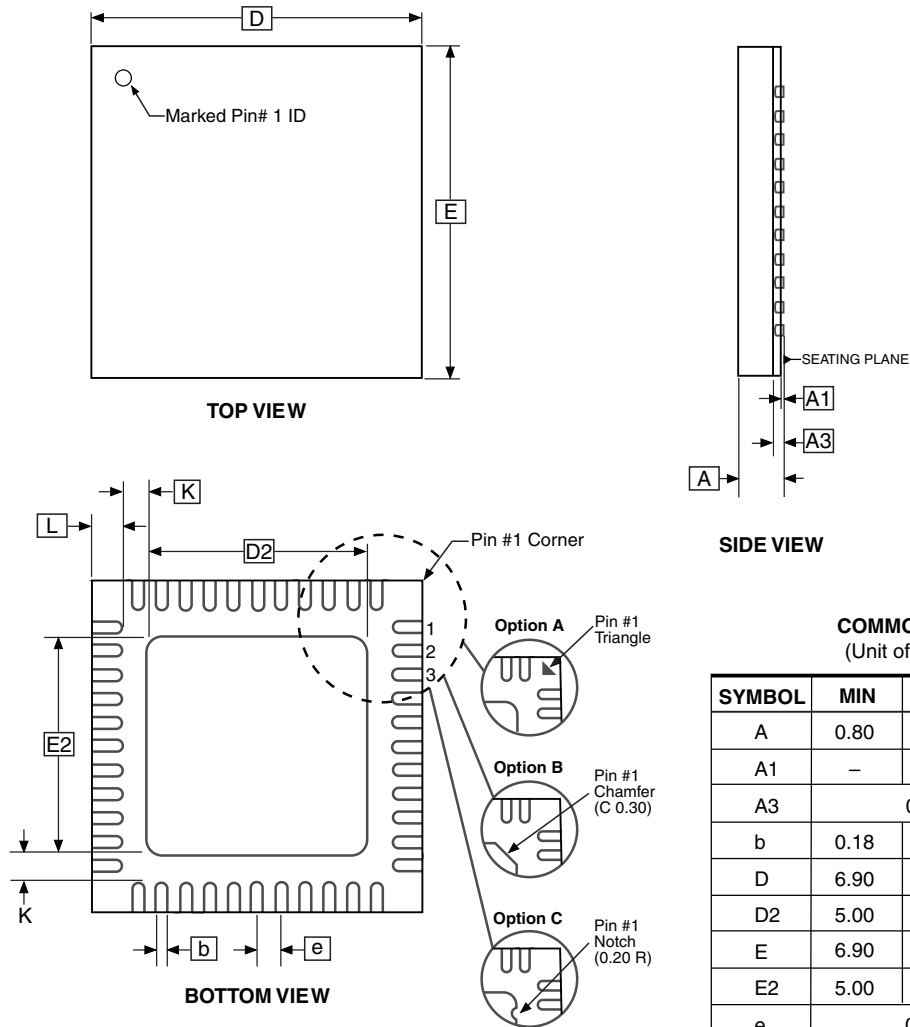
**DRAWING NO.**

40P6

**REV.**

B

## 44M1



Note: JEDEC Standard MO-220, Fig. 1 (SAW Singulation) VKKD-3.

9/26/08



**Package Drawing Contact:**  
packagedrawings@atmel.com

**TITLE**  
**44M1**, 44-pad, 7 x 7 x 1.0mm body, lead pitch 0.50mm, 5.20mm exposed pad, thermally enhanced plastic very thin quad flat no lead package (VQFN)

**GPC**  
ZWS

**DRAWING NO.**  
44M1

**REV.**  
H



## Errata

The revision letter in this section refers to the revision of the ATmega162 device.

### ATmega162, all rev.

There are no errata for this revision of ATmega162. However, a proposal for solving problems regarding the JTAG instruction IDCODE is presented below.

- **IDCODE masks data from TDI input**
- **Reading EEPROM by using ST or STS to set EERE bit triggers unexpected interrupt request**
- **Interrupts may be lost when writing the timer register in asynchronous timer**

#### 1. IDCODE masks data from TDI input

The public but optional JTAG instruction IDCODE is not implemented correctly according to IEEE1149.1; a logic one is scanned into the shift register instead of the TDI input while shifting the Device ID Register. Hence, captured data from the preceding devices in the boundary scan chain are lost and replaced by all-ones, and data to succeeding devices are replaced by all-ones during Update-DR.

If ATmega162 is the only device in the scan chain, the problem is not visible.

##### Problem Fix / Workaround

Select the Device ID Register of the ATmega162 (Either by issuing the IDCODE instruction or by entering the Test-Logic-Reset state of the TAP controller) to read out the contents of its Device ID Register and possibly data from succeeding devices of the scan chain. Note that data to succeeding devices cannot be entered during this scan, but data to preceding devices can. Issue the BYPASS instruction to the ATmega162 to select its Bypass Register while reading the Device ID Registers of preceding devices of the boundary scan chain. Never read data from succeeding devices in the boundary scan chain or upload data to the succeeding devices while the Device ID Register is selected for the ATmega162. Note that the IDCODE instruction is the default instruction selected by the Test-Logic-Reset state of the TAP-controller.

##### Alternative Problem Fix / Workaround

If the Device IDs of all devices in the boundary scan chain must be captured simultaneously (for instance if blind interrogation is used), the boundary scan chain can be connected in such way that the ATmega162 is the first device in the chain. Update-DR will still not work for the succeeding devices in the boundary scan chain as long as IDCODE is present in the JTAG Instruction Register, but the Device ID registered cannot be uploaded in any case.

#### 2. Reading EEPROM by using ST or STS to set EERE bit triggers unexpected interrupt request.

Reading EEPROM by using the ST or STS command to set the EERE bit in the EECR register triggers an unexpected EEPROM interrupt request.

##### Problem Fix / Workaround

Always use OUT or SBI to set EERE in EECR.

#### 3. Interrupts may be lost when writing the timer register in asynchronous timer

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

##### Problem Fix / Workaround

Always check that the asynchronous Timer/Counter register neither have the value 0xFF nor 0x00 before writing to the asynchronous Timer Control Register (TCCRx), asynchronous Timer Counter Register (TCNTx), or asynchronous Output Compare Register (OCRx).

## Datasheet Revision History

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.

- |                                                          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|----------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Changes from Rev. 2513K-08/07 to Rev. 2513L-03/13</b> | <ol style="list-style-type: none"> <li>1. Updated <a href="#">“Ordering Information” on page 14</a>:<br/>Removed -AI, -PI and -MI ordering codes. Only Pb-free package options are available.</li> </ol>                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| <b>Changes from Rev. 2513J-08/07 to Rev. 2513K-07/09</b> | <ol style="list-style-type: none"> <li>1. Updated <a href="#">“Errata” on page 18</a>.</li> <li>2. Updated the last page with Atmel’s new addresses.</li> </ol>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| <b>Changes from Rev. 2513I-04/07 to Rev. 2513J-08/07</b> | <ol style="list-style-type: none"> <li>1. Updated <a href="#">“Features” on page 1</a>.</li> <li>2. Added <a href="#">“Data Retention” on page 7</a>.</li> <li>3. Updated <a href="#">“Errata” on page 18</a>.</li> <li>4. Updated <a href="#">“Version” on page 205</a>.</li> <li>5. Updated <a href="#">“C Code Example<sup>(1)</sup>” on page 172</a>.</li> <li>6. Updated <a href="#">Figure 18 on page 35</a>.</li> <li>7. Updated <a href="#">“Clock Distribution” on page 35</a>.</li> <li>8. Updated <a href="#">“SPI Serial Programming Algorithm” on page 246</a>.</li> <li>9. Updated <a href="#">“Slave Mode” on page 162</a>.</li> </ol> |
| <b>Changes from Rev. 2513H-04/06 to Rev. 2513I-04/07</b> | <ol style="list-style-type: none"> <li>1. Updated <a href="#">“Using all 64KB Locations of External Memory” on page 34</a>.</li> <li>2. Updated <a href="#">“Bit 6 – ACBG: Analog Comparator Bandgap Select” on page 195</a>.</li> <li>3. Updated V<sub>OH</sub> conditions in <a href="#">“DC Characteristics” on page 264</a>.</li> </ol>                                                                                                                                                                                                                                                                                                           |
| <b>Changes from Rev. 2513G-03/05 to Rev. 2513H-04/06</b> | <ol style="list-style-type: none"> <li>1. Added <a href="#">“Resources” on page 7</a>.</li> <li>2. Updated <a href="#">“Calibrated Internal RC Oscillator” on page 38</a>.</li> <li>3. Updated note for <a href="#">Table 19 on page 50</a>.</li> <li>4. Updated <a href="#">“Serial Peripheral Interface – SPI” on page 157</a>.</li> </ol>                                                                                                                                                                                                                                                                                                          |
| <b>Changes from Rev. 2513F-09/03 to Rev. 2513G-03/05</b> | <ol style="list-style-type: none"> <li>1. MLF-package alternative changed to “Quad Flat No-Lead/Micro Lead Frame Package QFN/MLF”.</li> <li>2. Updated <a href="#">“Electrical Characteristics” on page 264</a></li> <li>3. Updated <a href="#">“Ordering Information” on page 14</a></li> </ol>                                                                                                                                                                                                                                                                                                                                                      |

**Changes from Rev. 2513D-04/03 to Rev. 2513E-09/03**

1. Removed “Preliminary” from the datasheet.
2. Added note on [Figure 1 on page 2](#).
3. Renamed and updated “On-chip Debug System” to “[JTAG Interface and On-chip Debug System](#)” on page 46.
4. Updated [Table 18 on page 48](#) and [Table 19 on page 50](#).
5. Updated “[Test Access Port – TAP](#)” on page 197 regarding JTAGEN.
6. Updated description for the JTD bit on [page 207](#).
7. Added note on JTAGEN in [Table 99 on page 233](#).
8. Updated Absolute Maximum Ratings\* and DC Characteristics in “[Electrical Characteristics](#)” on page 264.
9. Added a proposal for solving problems regarding the JTAG instruction IDCODE in “[Errata](#)” on page 18.

**Changes from Rev. 2513C-09/02 to Rev. 2513D-04/03**

1. Updated the “[Ordering Information](#)” on page 14 and “[Packaging Information](#)” on page 15.
2. Updated “[Features](#)” on page 1.
3. Added characterization plots under “[ATmega162 Typical Characteristics](#)” on page 275.
4. Added Chip Erase as a first step under “[Programming the Flash](#)” on page 260 and “[Programming the EEPROM](#)” on page 262.
5. Changed CAL7, the highest bit in the OSCCAL Register, to a reserved bit on [page 39](#) and in “[Register Summary](#)” on page 8.
6. Changed CPCE to CLKPCE on [page 41](#).
7. Corrected code examples on [page 55](#).
8. Corrected OCn waveforms in [Figure 52 on page 120](#).
9. Various minor Timer1 corrections.
10. Added note under “[Filling the Temporary Buffer \(Page Loading\)](#)” on page 224 about writing to the EEPROM during an SPM Page Load.
11. Added section “[EEPROM Write During Power-down Sleep Mode](#)” on page 24.
12. Added information about PWM symmetry for Timer0 on [page 98](#) and Timer2 on [page 147](#).
13. Updated [Table 18 on page 48](#), [Table 20 on page 50](#), [Table 36 on page 77](#), [Table 83 on page 205](#), [Table 109 on page 247](#), [Table 112 on page 267](#), and [Table 113 on page 268](#).

14. Added Figures for “[Absolute Maximum Frequency as a function of VCC, ATmega162](#)” on page 266.
15. Updated [Figure 29 on page 64](#), [Figure 32 on page 68](#), and [Figure 88 on page 210](#).
16. Removed Table 114, “External RC Oscillator, Typical Frequencies<sup>(1)</sup>,” on page 265.
17. Updated “[Electrical Characteristics](#)” on page 264.

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

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

**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 50, and “[Ordering Information](#)” on page 14.



Enabling Unlimited Possibilities®

**Atmel Corporation**

1600 Technology Drive  
San Jose, CA 95110  
USA

**Tel:** (+1) (408) 441-0311

**Fax:** (+1) (408) 487-2600

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

**Atmel Asia Limited**

Unit 01-5 & 16, 19F  
BEA Tower, Millennium City 5  
418 Kwun Tong Roa

Kwun Tong, Kowloon

HONG KONG

**Tel:** (+852) 2245-6100

**Fax:** (+852) 2722-1369

**Atmel Munich GmbH**

Business Campus  
Parking 4  
D-85748 Garching b. Munich  
GERMANY

**Tel:** (+49) 89-31970-0

**Fax:** (+49) 89-3194621

**Atmel Japan G.K.**

16F Shin-Osaki Kangyo Bldg  
1-6-4 Osaki, Shinagawa-ku  
Tokyo 141-0032  
JAPAN

**Tel:** (+81) (3) 6417-0300

**Fax:** (+81) (3) 6417-0370

© 2013 Atmel Corporation. All rights reserved. / Rev.: 2513LS-AVR-03/2013

Atmel®, Atmel logo and combinations thereof, Enabling Unlimited Possibilities®, AVR®, and others are registered trademarks or trademarks of Atmel Corporation or its subsidiaries. Other terms and product names may be trademarks of others.

Disclaimer: The information in this document is provided in connection with Atmel products. No license, express or implied, by estoppel or otherwise, to any intellectual property right is granted by this document or in connection with the sale of Atmel products. EXCEPT AS SET FORTH IN THE ATMEL TERMS AND CONDITIONS OF SALES LOCATED ON THE ATMEL WEBSITE, ATMEL ASSUMES NO LIABILITY WHATSOEVER AND DISCLAIMS ANY EXPRESS, IMPLIED OR STATUTORY WARRANTY RELATING TO ITS PRODUCTS INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. IN NO EVENT SHALL ATMEL BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, PUNITIVE, SPECIAL OR INCIDENTAL DAMAGES (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS AND PROFITS, BUSINESS INTERRUPTION, OR LOSS OF INFORMATION) ARISING OUT OF THE USE OR INABILITY TO USE THIS DOCUMENT, EVEN IF ATMEL HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Atmel makes no representations or warranties with respect to the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and products descriptions at any time without notice. Atmel does not make any commitment to update the information contained herein. Unless specifically provided otherwise, Atmel products are not suitable for, and shall not be used in, automotive applications. Atmel products are not intended, authorized, or warranted for use as components in applications intended to support or sustain life.