

## Features

- High-performance, Low-power Atmel® AVR® 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
- High Endurance Non-volatile Memory segments
  - 16 Kbytes of In-System Self-programmable Flash program memory
  - 512 Bytes EEPROM
  - 1 Kbyte Internal SRAM
  - Write/Erase Cycles: 10,000 Flash/100,000 EEPROM
  - Data retention: 20 years at 85°C/100 years at 25°C<sup>(1)</sup>
  - Optional Boot Code Section with Independent Lock Bits  
In-System Programming by On-chip Boot Program  
True Read-While-Write Operation
  - 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
  - One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
  - Real Time Counter with Separate Oscillator
  - Four PWM Channels
  - 8-channel, 10-bit ADC
    - 8 Single-ended Channels
    - 7 Differential Channels in TQFP Package Only
    - 2 Differential Channels with Programmable Gain at 1x, 10x, or 200x
  - Byte-oriented Two-wire Serial Interface
  - Programmable Serial USART
  - 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
  - Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby and Extended Standby
- I/O and Packages
  - 32 Programmable I/O Lines
  - 40-pin PDIP, 44-lead TQFP, and 44-pad QFN/MLF
- Operating Voltages
  - 2.7V - 5.5V for ATmega16L
  - 4.5V - 5.5V for ATmega16
- Speed Grades
  - 0 - 8 MHz for ATmega16L
  - 0 - 16 MHz for ATmega16
- Power Consumption @ 1 MHz, 3V, and 25°C for ATmega16L
  - Active: 1.1 mA
  - Idle Mode: 0.35 mA
  - Power-down Mode: < 1 µA



## 8-bit AVR® Microcontroller with 16K Bytes In-System Programmable Flash

ATmega16  
ATmega16L

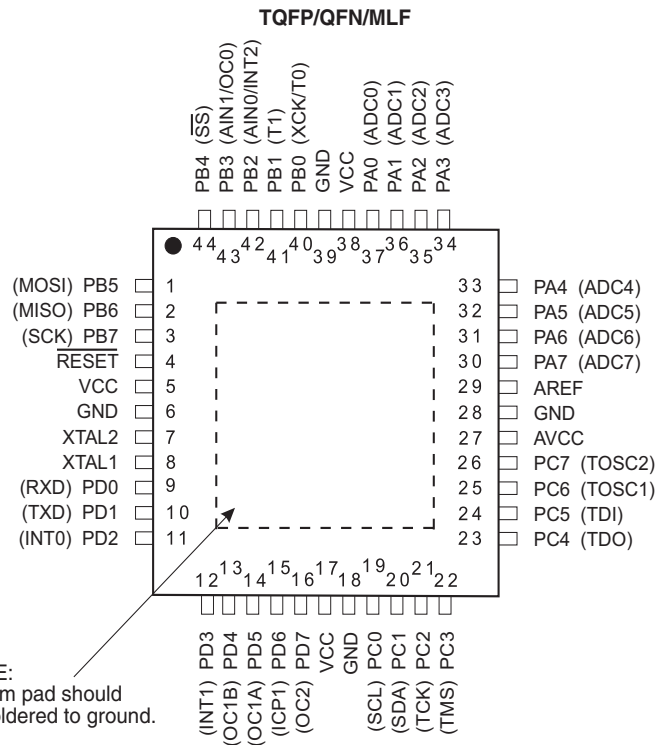
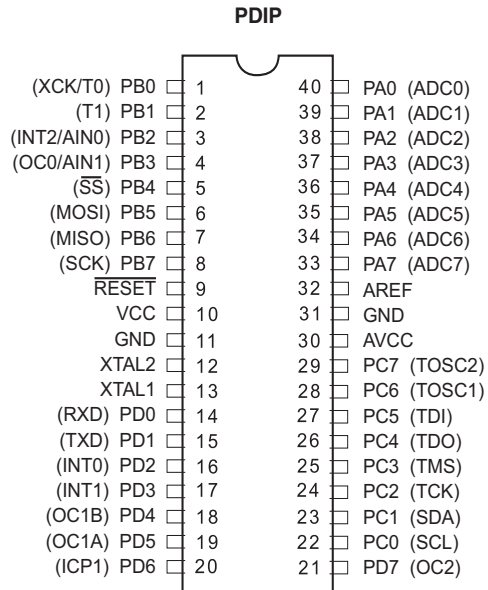
## Summary

Rev. 2466TS-AVR-07/10



# Pin Configurations

Figure 1. Pinout ATmega16



## Disclaimer

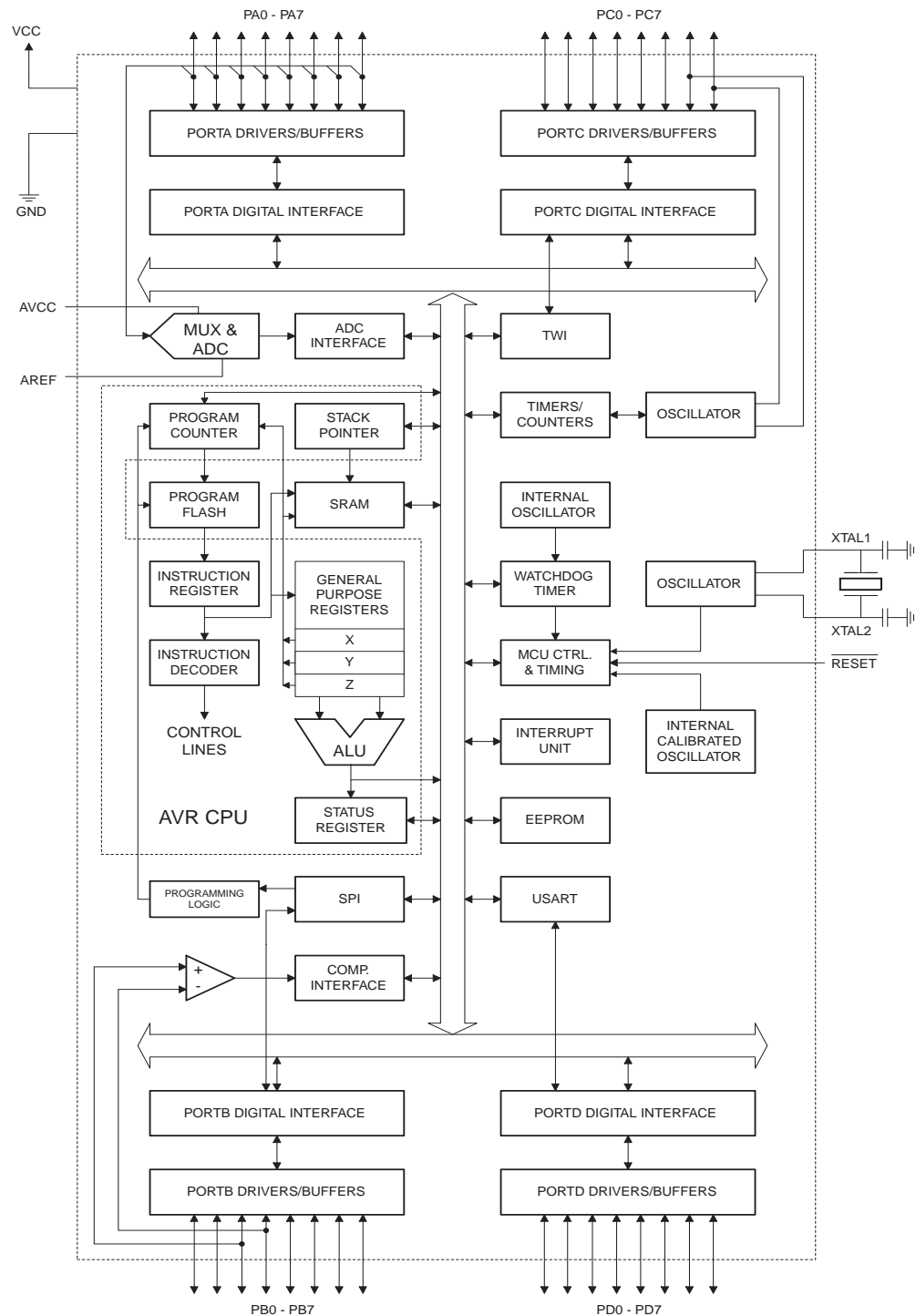
Typical values contained in this datasheet are based on simulations and characterization of other AVR microcontrollers manufactured on the same process technology. Min and Max values will be available after the device is characterized.

## Overview

The ATmega16 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 ATmega16 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 ATmega16 provides the following features: 16 Kbytes of In-System Programmable Flash Program memory with Read-While-Write capabilities, 512 bytes EEPROM, 1 Kbyte SRAM, 32 general purpose I/O lines, 32 general purpose working registers, a JTAG interface for Boundary-scan, On-chip Debugging support and programming, three flexible Timer/Counters with compare modes, Internal and External Interrupts, a serial programmable USART, a byte oriented Two-wire Serial Interface, an 8-channel, 10-bit ADC with optional differential input stage with programmable gain (TQFP package only), a programmable Watchdog Timer with Internal Oscillator, an SPI serial port, and six software selectable power saving modes. The Idle mode stops the CPU while allowing the USART, Two-wire interface, A/D Converter, 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 External Interrupt or Hardware Reset. In Power-save mode, the Asynchronous Timer continues to run, allowing the user to maintain a timer base while the rest of the device is sleeping. The ADC Noise Reduction mode stops the CPU and all I/O modules except Asynchronous Timer and ADC, to minimize switching noise during ADC conversions. In Standby mode, the crystal/resonator Oscillator is running while the rest of the device is sleeping. This allows very fast start-up combined with low-power consumption. In Extended Standby mode, both the main Oscillator and the Asynchronous Timer continue to run.

The device is manufactured using Atmel's high density nonvolatile memory technology. The On-chip ISP Flash allows the program memory to be reprogrammed in-system through an SPI serial interface, by a conventional nonvolatile 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 ATmega16 is a powerful microcontroller that provides a highly-flexible and cost-effective solution to many embedded control applications.

The ATmega16 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.

## Pin Descriptions

**VCC** Digital supply voltage.

**GND** Ground.

**Port A (PA7..PA0)** Port A serves as the analog inputs to the A/D Converter.

Port A also serves as an 8-bit bi-directional I/O port, if the A/D Converter is not used. Port pins can provide 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 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 ATmega16 as listed on [page 58](#).

## 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 PC5(TDI), PC3(TMS) and PC2(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 ATmega16 as listed on [page 61](#).

## 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 ATmega16 as listed on [page 63](#).

## $\overline{\text{RESET}}$

Reset Input. A low level on this pin for longer than the minimum pulse length will generate a reset, even if the clock is not running. The minimum pulse length is given in [Table 15 on page 38](#). 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.

## AVCC

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

## AREF

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



## 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                  |
|---|--------|--|--------|--------|---------|------------|--------|--------|--------|-----------------------|
| \$3F (\$5F)                               | SREG   | I  | T      | H      | S       | V          | N      | Z      | C      | 9                     |
| \$3E (\$5E)                               | SPH    | –  | –      | –      | –       | –          | SP10   | SP9    | SP8    | 12                    |
| \$3D (\$5D)                               | SPL    | SP7  | SP6    | SP5    | SP4     | SP3        | SP2    | SP1    | SP0    | 12                    |
| \$3C (\$5C)                               | OCR0   | Timer/Counter0 Output Compare Register               |        |        |         |            |        |        |        | 85                    |
| \$3B (\$5B)                               | GICR   | INT1   | INT0   | INT2   | –       | –          | –      | IVSEL  | IVCE   | 48, 69                |
| \$3A (\$5A)                               | GIFR   | INTF1  | INTF0  | INTF2  | –       | –          | –      | –      | –      | 70                    |
| \$39 (\$59)                               | TIMSK  | OCIE2  | TOIE2  | TICIE1 | OCIE1A  | OCIE1B     | TOIE1  | OCIE0  | TOIE0  | 85, 115, 133          |
| \$38 (\$58)                               | TIFR   | OCF2   | TOV2   | ICF1   | OCF1A   | OCF1B      | TOV1   | OCF0   | TOV0   | 86, 115, 133          |
| \$37 (\$57)                               | SPMCR  | SPMIE  | RWWWSB | –      | RWWWSRE | BLBSET     | PGWRT  | PGERS  | SPMEN  | 250                   |
| \$36 (\$56)                               | TWCR   | TWINT  | TWEA   | TWSTA  | TWSTO   | TWWC       | TWEN   | –      | TWIE   | 180                   |
| \$35 (\$55)                               | MCUCR  | SM2  | SE     | SM1    | SM0     | ISC11      | ISC10  | ISC01  | ISC00  | 32, 68                |
| \$34 (\$54)                               | MCUCSR | JTD  | ISC2   | –      | JTRF    | WDRF       | BORF   | EXTRF  | PORF   | 41, 69, 231           |
| \$33 (\$53)                               | TCCR0  | FOC0   | WGM00  | COM01  | COM00   | WGM01      | CS02   | CS01   | CS00   | 83                    |
| \$32 (\$52)                               | TCNT0  | Timer/Counter0 (8 Bits)                              |        |        |         |            |        |        |        | 85                    |
| \$31 <sup>(1)</sup> (\$51) <sup>(1)</sup> | OSCCAL | Oscillator Calibration Register                      |        |        |         |            |        |        |        | 30                    |
|   | OCDR   | On-Chip Debug Register                               |        |        |         |            |        |        |        | 227                   |
| \$30 (\$50)                               | SFIOR  | ADTS2  | ADTS1  | ADTS0  | –       | ACME       | PUD    | PSR2   | PSR10  | 57, 88, 134, 201, 221 |
| \$2F (\$4F)                               | TCCR1A | COM1A1   | COM1A0 | COM1B1 | COM1B0  | FOC1A      | FOC1B  | WGM11  | WGM10  | 110                   |
| \$2E (\$4E)                               | TCCR1B | ICNC1  | ICES1  | –      | WGM13   | WGM12      | CS12   | CS11   | CS10   | 113                   |
| \$2D (\$4D)                               | TCNT1H | Timer/Counter1 – Counter Register High Byte          |        |        |         |            |        |        |        | 114                   |
| \$2C (\$4C)                               | TCNT1L | Timer/Counter1 – Counter Register Low Byte           |        |        |         |            |        |        |        | 114                   |
| \$2B (\$4B)                               | OCR1AH | Timer/Counter1 – Output Compare Register A High Byte |        |        |         |            |        |        |        | 114                   |
| \$2A (\$4A)                               | OCR1AL | Timer/Counter1 – Output Compare Register A Low Byte  |        |        |         |            |        |        |        | 114                   |
| \$29 (\$49)                               | OCR1BH | Timer/Counter1 – Output Compare Register B High Byte |        |        |         |            |        |        |        | 114                   |
| \$28 (\$48)                               | OCR1BL | Timer/Counter1 – Output Compare Register B Low Byte  |        |        |         |            |        |        |        | 114                   |
| \$27 (\$47)                               | ICR1H  | Timer/Counter1 – Input Capture Register High Byte    |        |        |         |            |        |        |        | 114                   |
| \$26 (\$46)                               | ICR1L  | Timer/Counter1 – Input Capture Register Low Byte     |        |        |         |            |        |        |        | 114                   |
| \$25 (\$45)                               | TCCR2  | FOC2   | WGM20  | COM21  | COM20   | WGM21      | CS22   | CS21   | CS20   | 128                   |
| \$24 (\$44)                               | TCNT2  | Timer/Counter2 (8 Bits)                              |        |        |         |            |        |        |        | 130                   |
| \$23 (\$43)                               | OCR2   | Timer/Counter2 Output Compare Register               |        |        |         |            |        |        |        | 130                   |
| \$22 (\$42)                               | ASSR   | –  | –      | –      | –       | AS2        | TCN2UB | OCR2UB | TCR2UB | 131                   |
| \$21 (\$41)                               | WDTCSR | –  | –      | –      | WDTOE   | WDE        | WDP2   | WDP1   | WDP0   | 43                    |
| \$20 <sup>(2)</sup> (\$40) <sup>(2)</sup> | UBRRH  | URSEL  | –      | –      | –       | UBRR[11:8] |        |        |        | 167                   |
|   | UCSRC  | URSEL  | UMSEL  | UPM1   | UPM0    | USBS       | UCSZ1  | UCSZ0  | UCPOL  | 166                   |
| \$1F (\$3F)                               | EEARH  | –  | –      | –      | –       | –          | –      | –      | EEAR8  | 19                    |
| \$1E (\$3E)                               | EEARL  | EEPROM Address Register Low Byte                     |        |        |         |            |        |        |        | 19                    |
| \$1D (\$3D)                               | EEDR   | EEPROM Data Register                                 |        |        |         |            |        |        |        | 19                    |
| \$1C (\$3C)                               | EEDCR  | –  | –      | –      | –       | EERIE      | EEMWE  | EERE   | EERE   | 19                    |
| \$1B (\$3B)                               | PORTA  | PORTA7   | PORTA6 | PORTA5 | PORTA4  | PORTA3     | PORTA2 | PORTA1 | PORTA0 | 66                    |
| \$1A (\$3A)                               | DDRA   | DDA7   | DDA6   | DDA5   | DDA4    | DDA3       | DDA2   | DDA1   | DDA0   | 66                    |
| \$19 (\$39)                               | PINA   | PINA7  | PINA6  | PINA5  | PINA4   | PINA3      | PINA2  | PINA1  | PINA0  | 66                    |
| \$18 (\$38)                               | PORTB  | PORTB7   | PORTB6 | PORTB5 | PORTB4  | PORTB3     | PORTB2 | PORTB1 | PORTB0 | 66                    |
| \$17 (\$37)                               | DDRB   | DDB7   | DDB6   | DDB5   | DDB4    | DDB3       | DDB2   | DDB1   | DDB0   | 66                    |
| \$16 (\$36)                               | PINB   | PINB7  | PINB6  | PINB5  | PINB4   | PINB3      | PINB2  | PINB1  | PINB0  | 66                    |
| \$15 (\$35)                               | PORTC  | PORTC7   | PORTC6 | PORTC5 | PORTC4  | PORTC3     | PORTC2 | PORTC1 | PORTC0 | 67                    |
| \$14 (\$34)                               | DDRC   | DDC7   | DDC6   | DDC5   | DDC4    | DDC3       | DDC2   | DDC1   | DDC0   | 67                    |
| \$13 (\$33)                               | PINC   | PINC7  | PINC6  | PINC5  | PINC4   | PINC3      | PINC2  | PINC1  | PINC0  | 67                    |
| \$12 (\$32)                               | PORTD  | PORTD7   | PORTD6 | PORTD5 | PORTD4  | PORTD3     | PORTD2 | PORTD1 | PORTD0 | 67                    |
| \$11 (\$31)                               | DDRD   | DDD7   | DDD6   | DDD5   | DDD4    | DDD3       | DDD2   | DDD1   | DDD0   | 67                    |
| \$10 (\$30)                               | PIND   | PIND7  | PIND6  | PIND5  | PIND4   | PIND3      | PIND2  | PIND1  | PIND0  | 67                    |
| \$0F (\$2F)                               | SPDR   | SPI Data Register                                    |        |        |         |            |        |        |        | 142                   |
| \$0E (\$2E)                               | SPSR   | SPIF   | WCOL   | –      | –       | –          | –      | –      | SPI2X  | 142                   |
| \$0D (\$2D)                               | SPCR   | SPIE   | SPE    | DORD   | MSTR    | CPOL       | CPHA   | SPR1   | SPR0   | 140                   |
| \$0C (\$2C)                               | UDR    | USART I/O Data Register                              |        |        |         |            |        |        |        | 163                   |
| \$0B (\$2B)                               | UCSRA  | RXC  | TXC    | UDRE   | FE      | DOR        | PE     | U2X    | MPCM   | 164                   |
| \$0A (\$2A)                               | UCSRB  | RXCIE  | TXCIE  | UDRIE  | RXEN    | TXEN       | UCSZ2  | RXB8   | TXB8   | 165                   |
| \$09 (\$29)                               | UBRRL  | USART Baud Rate Register Low Byte                    |        |        |         |            |        |        |        | 167                   |
| \$08 (\$28)                               | ACSR   | ACD  | ACBG   | ACO    | ACI     | ACIE       | ACIC   | ACIS1  | ACIS0  | 202                   |
| \$07 (\$27)                               | ADMUX  | REFS1  | REFS0  | ADLAR  | MUX4    | MUX3       | MUX2   | MUX1   | MUX0   | 217                   |
| \$06 (\$26)                               | ADCSRA | ADEN   | ADSC   | ADATE  | ADIF    | ADIE       | ADPS2  | ADPS1  | ADPS0  | 219                   |
| \$05 (\$25)                               | ADCH   | ADC Data Register High Byte                          |        |        |         |            |        |        |        | 220                   |
| \$04 (\$24)                               | ADCL   | ADC Data Register Low Byte                           |        |        |         |            |        |        |        | 220                   |
| \$03 (\$23)                               | TWDR   | Two-wire Serial Interface Data Register              |        |        |         |            |        |        |        | 182                   |
| \$02 (\$22)                               | TWAR   | TWA6   | TWA5   | TWA4   | TWA3    | TWA2       | TWA1   | TWA0   | TWGCE  | 182                   |



| Address     | Name | Bit 7                                       | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Page |
|-------------|------|---|-------|-------|-------|-------|-------|-------|-------|------|
| \$01 (\$21) | TWSR | TWS7  | TWS6  | TWS5  | TWS4  | TWS3  | –     | TWPS1 | TWPS0 | 181  |
| \$00 (\$20) | TWBR | Two-wire Serial Interface Bit Rate Register |       |       |       |       |       |       |       | 180  |

- 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 \$00 to \$1F 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                                     | RdI,K    | Add Immediate to Word                    | $Rdh:Rdl \leftarrow Rdh:Rdl + K$               | Z,C,N,V,S  | 2         |
| SUB                                      | Rd, Rr   | Subtract two Registers                   | $Rd \leftarrow Rd - Rr$                        | Z,C,N,V,H  | 1         |
| SUBI                                     | Rd, K    | Subtract Constant from Register          | $Rd \leftarrow Rd - K$                         | Z,C,N,V,H  | 1         |
| SBC                                      | Rd, Rr   | Subtract with Carry two Registers        | $Rd \leftarrow Rd - Rr - C$                    | Z,C,N,V,H  | 1         |
| SBCI                                     | Rd, K    | Subtract with Carry Constant from Reg.   | $Rd \leftarrow Rd - K - C$                     | Z,C,N,V,H  | 1         |
| SBIW                                     | RdI,K    | Subtract Immediate from Word             | $Rdh:Rdl \leftarrow Rdh:Rdl - K$               | Z,C,N,V,S  | 2         |
| AND                                      | Rd, Rr   | Logical AND Registers                    | $Rd \leftarrow Rd \bullet Rr$                  | Z,N,V      | 1         |
| ANDI                                     | Rd, K    | Logical AND Register and Constant        | $Rd \leftarrow Rd \bullet K$                   | Z,N,V      | 1         |
| OR                                       | Rd, Rr   | Logical OR Registers                     | $Rd \leftarrow Rd \vee Rr$                     | Z,N,V      | 1         |
| ORI                                      | Rd, K    | Logical OR Register and Constant         | $Rd \leftarrow Rd \vee K$                      | Z,N,V      | 1         |
| EOR                                      | Rd, Rr   | Exclusive OR Registers                   | $Rd \leftarrow Rd \oplus Rr$                   | Z,N,V      | 1         |
| COM                                      | Rd       | One's Complement                         | $Rd \leftarrow \$FF - Rd$                      | Z,C,N,V    | 1         |
| NEG                                      | Rd       | Two's Complement                         | $Rd \leftarrow \$00 - Rd$                      | 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                 | $Rd \leftarrow Rd \bullet (\$FF - K)$          | 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                             | $Rd \leftarrow \$FF$                           | None       | 1         |
| MUL                                      | Rd, Rr   | Multiply Unsigned                        | $R1:R0 \leftarrow Rd \times Rr$                | Z,C        | 2         |
| MULS                                     | Rd, Rr   | Multiply Signed                          | $R1:R0 \leftarrow Rd \times Rr$                | Z,C        | 2         |
| MULSU                                    | Rd, Rr   | Multiply Signed with Unsigned            | $R1:R0 \leftarrow Rd \times Rr$                | Z,C        | 2         |
| FMUL                                     | Rd, Rr   | Fractional Multiply Unsigned             | $R1:R0 \leftarrow (Rd \times Rr) \lll 1$       | Z,C        | 2         |
| FMULS                                    | Rd, Rr   | Fractional Multiply Signed               | $R1:R0 \leftarrow (Rd \times Rr) \lll 1$       | Z,C        | 2         |
| FMULSU                                   | Rd, Rr   | Fractional Multiply Signed with Unsigned | $R1:R0 \leftarrow (Rd \times Rr) \lll 1$       | Z,C        | 2         |
| <b>BRANCH INSTRUCTIONS</b>               |          |  |  |            |           |
| RJMP                                     | k        | Relative Jump                            | $PC \leftarrow PC + k + 1$                     | None       | 2         |
| IJMP                                     |          | Indirect Jump to (Z)                     | $PC \leftarrow Z$                              | None       | 2         |
| JMP                                      | 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 $(SREG(s) = 1) PC \leftarrow PC + k + 1$    | None       | 1 / 2     |
| BRBC                                     | s, k     | Branch if Status Flag Cleared            | if $(SREG(s) = 0) PC \leftarrow PC + k + 1$    | None       | 1 / 2     |
| BREQ                                     | k        | Branch if Equal                          | if $(Z = 1) PC \leftarrow PC + k + 1$          | None       | 1 / 2     |
| BRNE                                     | k        | Branch if Not Equal                      | if $(Z = 0) PC \leftarrow PC + k + 1$          | None       | 1 / 2     |
| BRCS                                     | k        | Branch if Carry Set                      | if $(C = 1) PC \leftarrow PC + k + 1$          | None       | 1 / 2     |
| BRCC                                     | k        | Branch if Carry Cleared                  | if $(C = 0) PC \leftarrow PC + k + 1$          | None       | 1 / 2     |
| BRSH                                     | k        | Branch if Same or Higher                 | if $(C = 0) PC \leftarrow PC + k + 1$          | None       | 1 / 2     |
| BRLO                                     | k        | Branch if Lower                          | if $(C = 1) PC \leftarrow PC + k + 1$          | None       | 1 / 2     |
| BRMI                                     | k        | Branch if Minus                          | if $(N = 1) PC \leftarrow PC + k + 1$          | None       | 1 / 2     |
| BRPL                                     | k        | Branch if Plus                           | if $(N = 0) PC \leftarrow PC + k + 1$          | None       | 1 / 2     |
| BRGE                                     | k        | Branch if Greater or Equal, Signed       | if $(N \oplus V = 0) PC \leftarrow PC + k + 1$ | None       | 1 / 2     |
| BRLT                                     | k        | Branch if Less Than Zero, Signed         | if $(N \oplus V = 1) PC \leftarrow PC + k + 1$ | None       | 1 / 2     |
| BRHS                                     | k        | Branch if Half Carry Flag Set            | if $(H = 1) PC \leftarrow PC + k + 1$          | None       | 1 / 2     |
| BRHC                                     | k        | Branch if Half Carry Flag Cleared        | if $(H = 0) PC \leftarrow PC + k + 1$          | None       | 1 / 2     |
| BRTS                                     | k        | Branch if T Flag Set                     | if $(T = 1) PC \leftarrow PC + k + 1$          | None       | 1 / 2     |
| BRTC                                     | k        | Branch if T Flag Cleared                 | if $(T = 0) PC \leftarrow PC + k + 1$          | None       | 1 / 2     |
| BRVS                                     | k        | Branch if Overflow Flag is Set           | if $(V = 1) PC \leftarrow PC + k + 1$          | None       | 1 / 2     |
| BRVC                                     | k        | Branch if Overflow Flag is Cleared       | if $(V = 0) PC \leftarrow PC + k + 1$          | 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 ← 0                                    | H     | 1       |
| MCU CONTROL INSTRUCTIONS |          |                               |  |       |         |
| 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

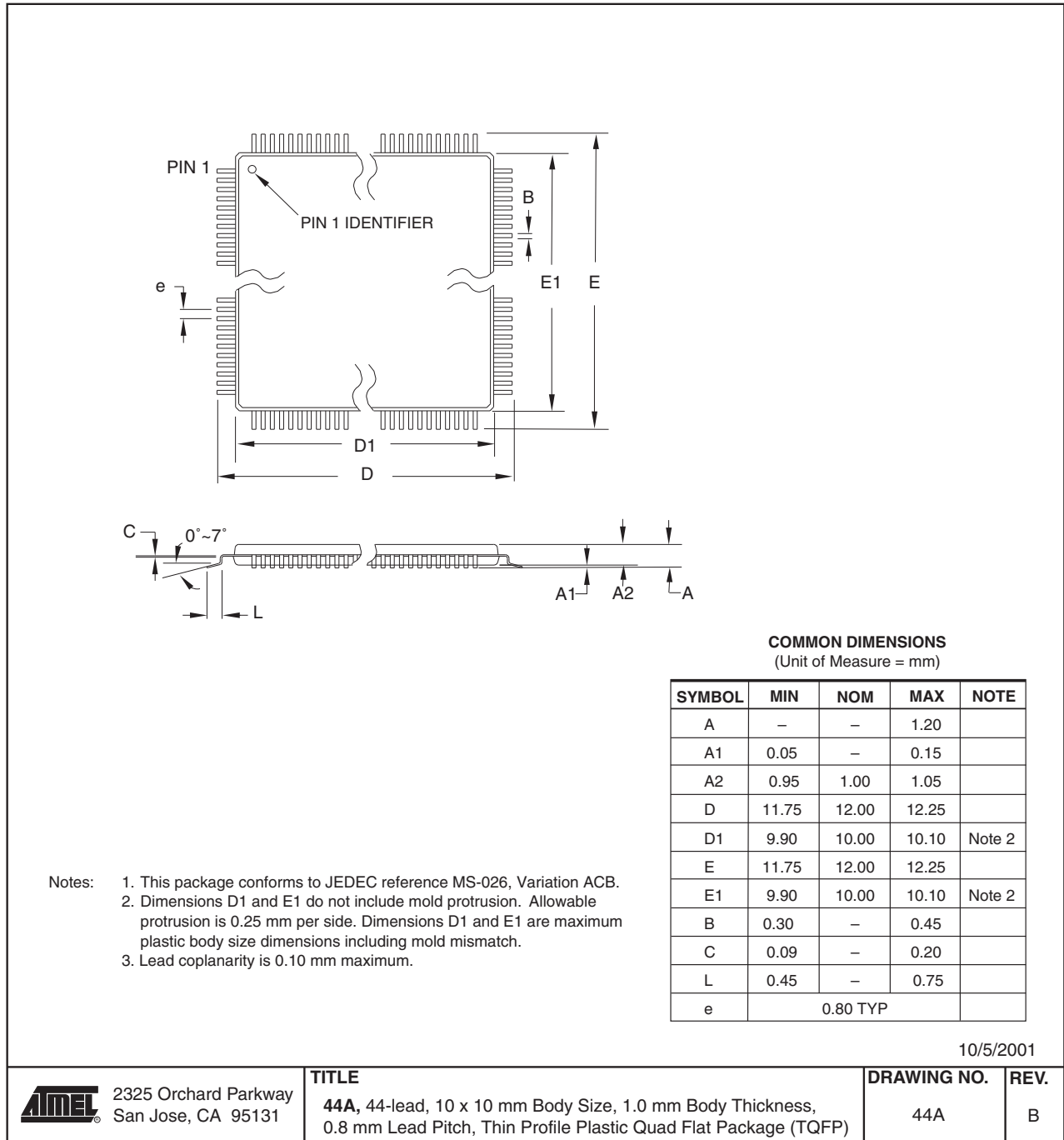
| Speed (MHz) | Power Supply | Ordering Code  | Package             | Operation Range               |
|-------------|--------------|--|---------------------|-------------------------------|
| 8           | 2.7V - 5.5V  | ATmega16L-8AU <sup>(1)</sup><br>ATmega16L-8PU <sup>(1)</sup><br>ATmega16L-8MU <sup>(1)</sup> | 44A<br>40P6<br>44M1 | Industrial<br>(-40°C to 85°C) |
| 16          | 4.5V - 5.5V  | ATmega16-16AU <sup>(1)</sup><br>ATmega16-16PU <sup>(1)</sup><br>ATmega16-16MU <sup>(1)</sup> | 44A<br>40P6<br>44M1 | Industrial<br>(-40°C to 85°C) |

Note: 1. Pb-free packaging complies to the European Directive for Restriction of Hazardous Substances (RoHS directive). Also Halide free and fully Green.

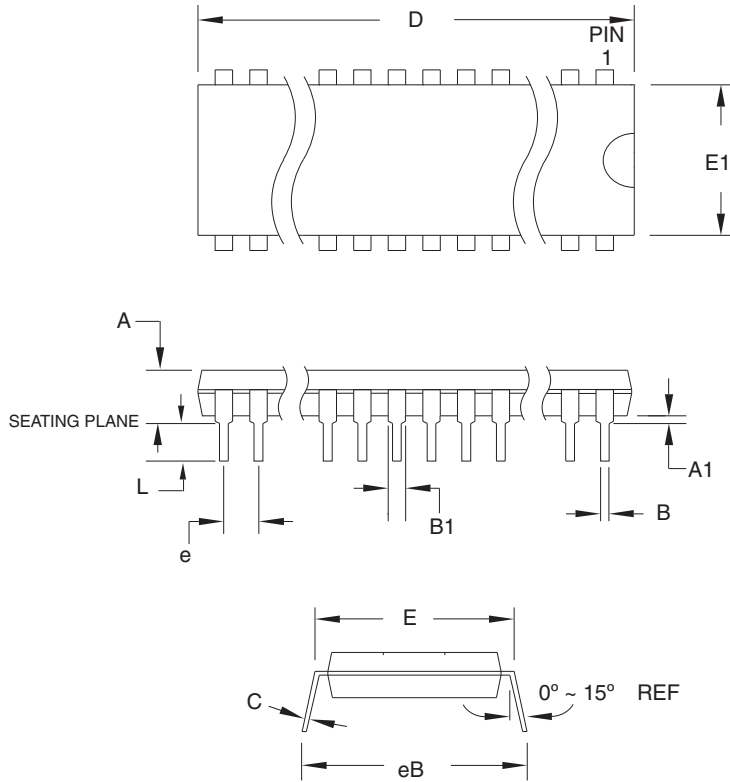
| Package Type |   |
|--------------|---|
| <b>44A</b>   | 44-lead, Thin (1.0 mm) Plastic Gull Wing Quad Flat Package (TQFP)                                     |
| <b>40P6</b>  | 40-pin, 0.600" Wide, Plastic Dual Inline Package (PDIP)   |
| <b>44M1</b>  | 44-pad, 7 × 7 × 1.0 mm body, lead pitch 0.50 mm, Quad Flat No-Lead/Micro Lead Frame Package (QFN/MLF) |

## Packaging Information

44A



# 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 0.25 mm (0.010").

09/28/01



2325 Orchard Parkway  
San Jose, CA 95131

**TITLE**

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

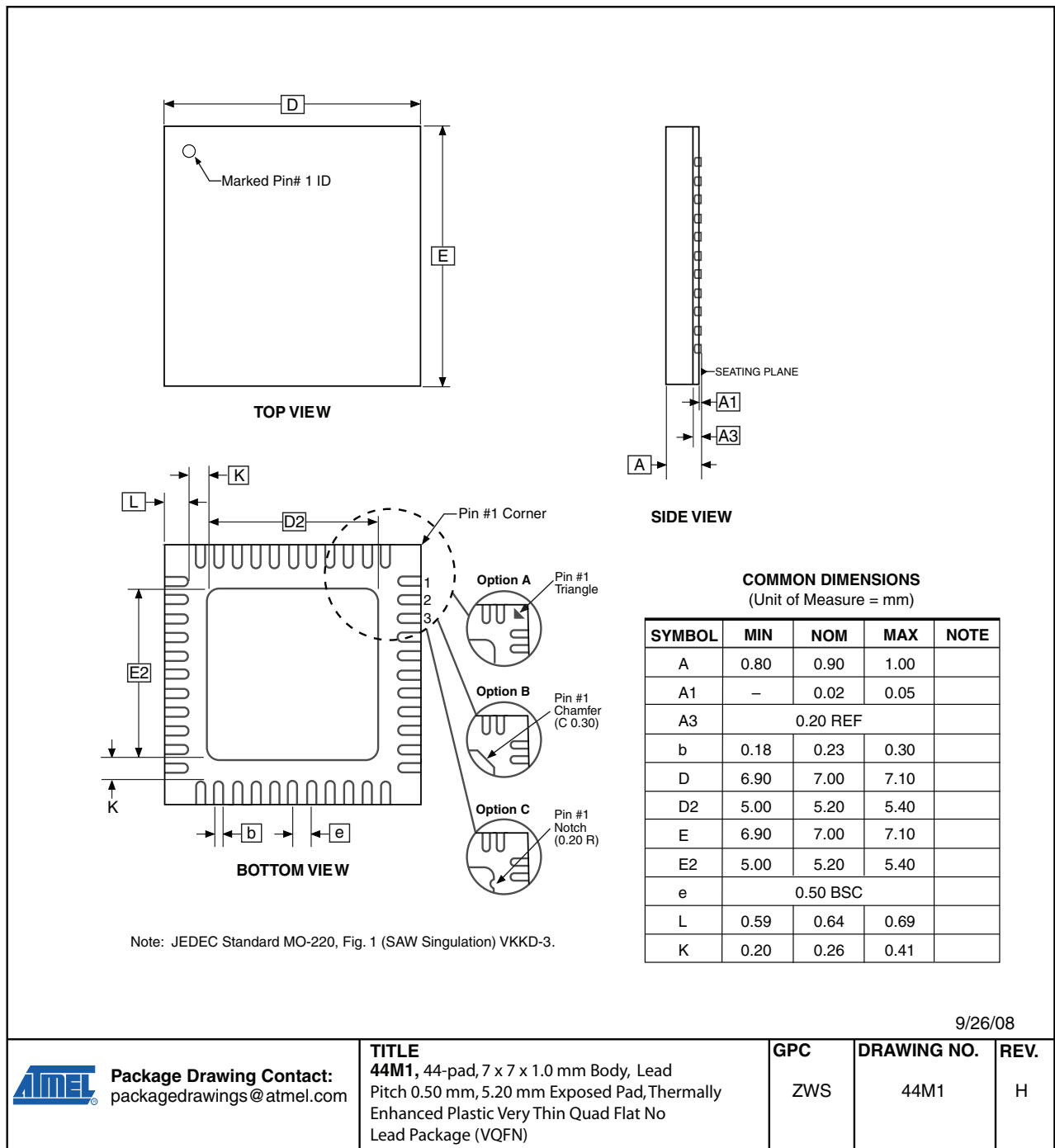
**DRAWING NO.**

40P6

**REV.**

B

44M1



## Errata

The revision letter in this section refers to the revision of the ATmega16 device.

### ATmega16(L) Rev. M

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

#### 1. First Analog Comparator conversion may be delayed

If the device is powered by a slow rising  $V_{CC}$ , the first Analog Comparator conversion will take longer than expected on some devices.

##### Problem Fix/Workaround

When the device has been powered or reset, disable then enable the Analog Comparator before the first conversion.

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

The interrupt will be lost if a timer register that is synchronized to the asynchronous 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).

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

The JTAG instruction IDCODE is not working correctly. Data to succeeding devices are replaced by all-ones during Update-DR.

##### Problem Fix / Workaround

- If ATmega16 is the only device in the scan chain, the problem is not visible.
- Select the Device ID Register of the ATmega16 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. Issue the BYPASS instruction to the ATmega16 while reading the Device ID Registers of preceding devices of the boundary scan chain.
- If the Device IDs of all devices in the boundary scan chain must be captured simultaneously, the ATmega16 must be the first device in the chain.

#### 4. 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.

### ATmega16(L) Rev. L

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

#### 1. First Analog Comparator conversion may be delayed

If the device is powered by a slow rising  $V_{CC}$ , the first Analog Comparator conversion will take longer than expected on some devices.

## Problem Fix/Workaround

When the device has been powered or reset, disable then enable the Analog Comparator before the first conversion.

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

The interrupt will be lost if a timer register that is synchronized to the asynchronous 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).

## 3. IDCODE masks data from TDI input

The JTAG instruction IDCODE is not working correctly. Data to succeeding devices are replaced by all-ones during Update-DR.

### Problem Fix / Workaround

- If ATmega16 is the only device in the scan chain, the problem is not visible.
- Select the Device ID Register of the ATmega16 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. Issue the BYPASS instruction to the ATmega16 while reading the Device ID Registers of preceding devices of the boundary scan chain.
- If the Device IDs of all devices in the boundary scan chain must be captured simultaneously, the ATmega16 must be the first device in the chain.

## 4. 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.

## ATmega16(L) Rev. K

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

## 1. First Analog Comparator conversion may be delayed

If the device is powered by a slow rising  $V_{CC}$ , the first Analog Comparator conversion will take longer than expected on some devices.

### Problem Fix/Workaround

When the device has been powered or reset, disable then enable the Analog Comparator before the first conversion.

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

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

### Problem Fix / Workaround

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

### 3. IDCODE masks data from TDI input

The JTAG instruction IDCODE is not working correctly. Data to succeeding devices are replaced by all-ones during Update-DR.

#### Problem Fix / Workaround

- If ATmega16 is the only device in the scan chain, the problem is not visible.
- Select the Device ID Register of the ATmega16 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. Issue the BYPASS instruction to the ATmega16 while reading the Device ID Registers of preceding devices of the boundary scan chain.
- If the Device IDs of all devices in the boundary scan chain must be captured simultaneously, the ATmega16 must be the first device in the chain.

### 4. 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.

## ATmega16(L) Rev. J

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

### 1. First Analog Comparator conversion may be delayed

If the device is powered by a slow rising  $V_{CC}$ , the first Analog Comparator conversion will take longer than expected on some devices.

#### Problem Fix/Workaround

When the device has been powered or reset, disable then enable the Analog Comparator before the first conversion.

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

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

#### Problem Fix / Workaround

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

### 3. IDCODE masks data from TDI input

The JTAG instruction IDCODE is not working correctly. Data to succeeding devices are replaced by all-ones during Update-DR.

## Problem Fix / Workaround

- If ATmega16 is the only device in the scan chain, the problem is not visible.
- Select the Device ID Register of the ATmega16 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. Issue the BYPASS instruction to the ATmega16 while reading the Device ID Registers of preceding devices of the boundary scan chain.
- If the Device IDs of all devices in the boundary scan chain must be captured simultaneously, the ATmega16 must be the first device in the chain.

## 4. 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.

## ATmega16(L) Rev.

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

### 1. First Analog Comparator conversion may be delayed

If the device is powered by a slow rising  $V_{CC}$ , the first Analog Comparator conversion will take longer than expected on some devices.

#### Problem Fix/Workaround

When the device has been powered or reset, disable then enable the Analog Comparator before the first conversion.

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

The interrupt will be lost if a timer register that is synchronized to the asynchronous 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 (TCR<sub>x</sub>), asynchronous Timer Counter Register (TCNT<sub>x</sub>), or asynchronous Output Compare Register (OCR<sub>x</sub>).

### 3. IDCODE masks data from TDI input

The JTAG instruction IDCODE is not working correctly. Data to succeeding devices are replaced by all-ones during Update-DR.

#### Problem Fix / Workaround

- If ATmega16 is the only device in the scan chain, the problem is not visible.
- Select the Device ID Register of the ATmega16 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. Issue the BYPASS instruction to the ATmega16 while reading the Device ID Registers of preceding devices of the boundary scan chain.
- If the Device IDs of all devices in the boundary scan chain must be captured simultaneously, the ATmega16 must be the first device in the chain.

**4. 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.

**ATmega16(L) Rev. H**

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

**1. First Analog Comparator conversion may be delayed**

If the device is powered by a slow rising  $V_{CC}$ , the first Analog Comparator conversion will take longer than expected on some devices.

**Problem Fix/Workaround**

When the device has been powered or reset, disable then enable the Analog Comparator before the first conversion.

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

The interrupt will be lost if a timer register that is synchronized to the asynchronous 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).

**3. IDCODE masks data from TDI input**

The JTAG instruction IDCODE is not working correctly. Data to succeeding devices are replaced by all-ones during Update-DR.

**Problem Fix / Workaround**

- If ATmega16 is the only device in the scan chain, the problem is not visible.
- Select the Device ID Register of the ATmega16 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. Issue the BYPASS instruction to the ATmega16 while reading the Device ID Registers of preceding devices of the boundary scan chain.
- If the Device IDs of all devices in the boundary scan chain must be captured simultaneously, the ATmega16 must be the first device in the chain.

**4. 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.

## 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.

- Rev. 2466T-07/10**
1. Corrected use of comma in formula  $R_p$  in [Table 120, “Two-wire Serial Bus Requirements,” on page 294.](#)
  2. Updated document according to Atmel’s Technical Terminology
  3. Note 6 and Note 7 under [Table 120, “Two-wire Serial Bus Requirements,” on page 294](#) have been removed.
- Rev. 2466S-05/09**
1. Updated [“Errata” on page 340.](#)
  2. Updated the last page with Atmel’s new addresses.
- Rev. 2466R-06/08**
1. Added [“Not recommended for new designs” note in Figure on page 1.](#)
- Rev. 2466Q-05/08**
1. Updated [“Fast PWM Mode” on page 77](#) in [“8-bit Timer/Counter0 with PWM” on page 71:](#)
    - Removed the last section describing how to achieve a frequency with 50% duty cycle waveform output in fast PWM mode.
  2. Removed note from Feature list in [“Analog to Digital Converter” on page 204.](#)
  3. Removed note from [Table 84 on page 218.](#)
  4. Updated [“Ordering Information” on page 336:](#)
    - Commercial ordering codes removed.
    - Non Pb-free package option removed.
- Rev. 2466P-08/07**
1. Updated [“Features” on page 1.](#)
  2. Added [“Data Retention” on page 6.](#)
  3. Updated [“Errata” on page 340.](#)
  4. Updated [“Slave Mode” on page 140.](#)
- Rev. 2466O-03/07**
1. Updated [“Calibrated Internal RC Oscillator” on page 29.](#)
  2. Updated C code example in [“USART Initialization” on page 149.](#)
  3. Updated [“ATmega16 Boundary-scan Order” on page 241.](#)
  4. Removed [“preliminary” from “ADC Characteristics” on page 297.](#)
  5. Updated from V to mV in [“I/O Pin Input Hysteresis vs.  \$V\_{CC}\$ ” on page 317.](#)
  6. Updated from V to mV in [“Reset Input Pin Hysteresis vs.  \$V\_{CC}\$ ” on page 318.](#)

- Rev. 2466N-10/06**
1. Updated “[Timer/Counter Oscillator](#)” on page 31.
  2. Updated “[Fast PWM Mode](#)” on page 102.
  3. Updated [Table 38 on page 83](#), [Table 40 on page 84](#), [Table 45 on page 111](#), [Table 47 on page 112](#), [Table 50 on page 128](#) and [Table 52 on page 129](#).
  4. Updated C code example in “[USART Initialization](#)” on page 149.
  5. Updated “[Errata](#)” on page 340.
- Rev. 2466M-04/06**
1. Updated typos.
  2. Updated “[Serial Peripheral Interface – SPI](#)” on page 135.
  3. Updated [Table 86 on page 221](#), [Table 116 on page 276](#) ,[Table 121 on page 295](#) and [Table 122 on page 297](#).
- Rev. 2466L-06/05**
1. Updated note in “[Bit Rate Generator Unit](#)” on page 178.
  2. Updated values for  $V_{INT}$  in “[ADC Characteristics](#)” on page 297.
  3. Updated “[Serial Programming Instruction set](#)” on page 276.
  4. Updated USART init C-code example in “[USART](#)” on page 144.
- Rev. 2466K-04/05**
1. Updated “[Ordering Information](#)” on page 336.
  2. MLF-package alternative changed to “[Quad Flat No-Lead/Micro Lead Frame Package QFN/MLF](#)”.
  3. Updated “[Electrical Characteristics](#)” on page 291.
- Rev. 2466J-10/04**
1. Updated “[Ordering Information](#)” on page 336.
- Rev. 2466I-10/04**
1. Removed references to analog ground.
  2. Updated [Table 7 on page 28](#), [Table 15 on page 38](#), [Table 16 on page 42](#), [Table 81 on page 209](#), [Table 116 on page 276](#), and [Table 119 on page 293](#).
  3. Updated “[Pinout ATmega16](#)” on page 2.
  4. Updated features in “[Analog to Digital Converter](#)” on page 204.
  5. Updated “[Version](#)” on page 229.
  6. Updated “[Calibration Byte](#)” on page 261.
  7. Added “[Page Size](#)” on page 262.
- Rev. 2466H-12/03**
1. Updated “[Calibrated Internal RC Oscillator](#)” on page 29.

## Rev. 2466G-10/03

1. Removed “Preliminary” from the datasheet.
2. Changed ICP to ICP1 in the datasheet.
3. Updated “[JTAG Interface and On-chip Debug System](#)” on page 36.
4. Updated assembly and C code examples in “[Watchdog Timer Control Register – WDTCR](#)” on page 43.
5. Updated [Figure 46](#) on page 103.
6. Updated [Table 15](#) on page 38, [Table 82](#) on page 217 and [Table 115](#) on page 276.
7. Updated “[Test Access Port – TAP](#)” on page 222 regarding JTAGEN.
8. Updated description for the JTD bit on [page 231](#).
9. Added note 2 to [Figure 126](#) on page 252.
10. Added a note regarding JTAGEN fuse to [Table 105](#) on page 260.
11. Updated Absolute Maximum Ratings\* and DC Characteristics in “[Electrical Characteristics](#)” on page 291.
12. Updated “[ATmega16 Typical Characteristics](#)” on page 299.
13. Fixed typo for 16 MHz QFN/MLF package in “[Ordering Information](#)” on page 336.
14. Added a proposal for solving problems regarding the JTAG instruction IDCODE in “[Errata](#)” on page 340.

## Rev. 2466F-02/03

1. Added note about masking out unused bits when reading the Program Counter in “[Stack Pointer](#)” on page 12.
2. Added Chip Erase as a first step in “[Programming the Flash](#)” on page 288 and “[Programming the EEPROM](#)” on page 289.
3. Added the section “[Unconnected pins](#)” on page 55.
4. Added tips on how to disable the OCD system in “[On-chip Debug System](#)” on page 34.
5. Removed reference to the “[Multi-purpose Oscillator](#)” application note and “[32 kHz Crystal Oscillator](#)” application note, which do not exist.
6. Added information about PWM symmetry for Timer0 and Timer2.
7. Added note in “[Filling the Temporary Buffer \(Page Loading\)](#)” on page 253 about writing to the EEPROM during an SPM Page Load.
8. Removed ADHSM completely.

9. Added [Table 73, “TWI Bit Rate Prescaler,”](#) on page 182 to describe the TWPS bits in the [“TWI Status Register – TWSR”](#) on page 181.
10. Added section [“Default Clock Source”](#) on page 25.
11. Added note about frequency variation when using an external clock. Note added in [“External Clock”](#) on page 31. An extra row and a note added in [Table 118](#) on page 293.
12. Various minor TWI corrections.
13. Added [“Power Consumption”](#) data in [“Features”](#) on page 1.
14. Added section [“EEPROM Write During Power-down Sleep Mode”](#) on page 22.
15. Added note about Differential Mode with Auto Triggering in [“Prescaling and Conversion Timing”](#) on page 207.
16. Added updated [“Packaging Information”](#) on page 337.

**Rev. 2466E-10/02**

1. Updated [“DC Characteristics”](#) on page 291.

**Rev. 2466D-09/02**

1. Changed all Flash write/erase cycles from 1,000 to 10,000.
2. Updated the following tables: [Table 4](#) on page 26, [Table 15](#) on page 38, [Table 42](#) on page 85, [Table 45](#) on page 111, [Table 46](#) on page 111, [Table 59](#) on page 143, [Table 67](#) on page 167, [Table 90](#) on page 235, [Table 102](#) on page 258, [“DC Characteristics”](#) on page 291, [Table 119](#) on page 293, [Table 121](#) on page 295, and [Table 122](#) on page 297.
3. Updated [“Errata”](#) on page 340.

**Rev. 2466C-03/02**

1. Updated typical EEPROM programming time, [Table 1](#) on page 20.
2. Updated typical start-up time in the following tables:  
[Table 3](#) on page 25, [Table 5](#) on page 27, [Table 6](#) on page 28, [Table 8](#) on page 29, [Table 9](#) on page 29, and [Table 10](#) on page 29.
3. Updated [Table 17](#) on page 43 with typical WDT Time-out.
4. **Added Some Preliminary Test Limits and Characterization Data.**  
Removed some of the TBD's in the following tables and pages:  
[Table 15](#) on page 38, [Table 16](#) on page 42, [Table 116](#) on page 272 (table removed in document review #D), [“Electrical Characteristics”](#) on page 291, [Table 119](#) on page 293, [Table 121](#) on page 295, and [Table 122](#) on page 297.
5. **Updated TWI Chapter.**  
Added the note at the end of the [“Bit Rate Generator Unit”](#) on page 178.
6. **Corrected description of ADSC bit in [“ADC Control and Status Register A – ADCSRA”](#) on page 219.**
7. **Improved description on how to do a polarity check of the ADC doff results in [“ADC Conversion Result”](#) on page 216.**

8. Added JTAG version number for rev. H in [Table 87 on page 229](#).
9. Added note regarding OCDEN Fuse below [Table 105 on page 260](#).
10. Updated Programming Figures:  
[Figure 127 on page 262](#) and [Figure 136 on page 273](#) are updated to also reflect that AVCC must be connected during Programming mode. [Figure 131 on page 269](#) added to illustrate how to program the fuses.
11. Added a note regarding usage of the “[PROG\\_PAGELOAD \(\\$6\)](#)” on [page 280](#) and “[PROG\\_PAGEREAD \(\\$7\)](#)” on [page 280](#).
12. Removed alternative algorithm for leaving JTAG Programming mode.  
See “Leaving Programming Mode” on [page 288](#).
13. Added Calibrated RC Oscillator characterization curves in section “[ATmega16 Typical Characteristics](#)” on [page 299](#).
14. Corrected ordering code for QFN/MLF package (16MHz) in “[Ordering Information](#)” on [page 336](#).
15. Corrected [Table 90](#), “[Scan Signals for the Oscillators<sup>\(1\)\(2\)\(3\)</sup>](#),” on [page 235](#).



## Headquarters

---

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

## International

---

**Atmel Asia**  
Unit 1-5 & 16, 19/F  
BEA Tower, Millennium City 5  
418 Kwun Tong Road  
Kwun Tong, Kowloon  
Hong Kong  
Tel: (852) 2245-6100  
Fax: (852) 2722-1369

**Atmel Europe**  
Le Krebs  
8, Rue Jean-Pierre Timbaud  
BP 309  
78054 Saint-Quentin-en-  
Yvelines Cedex  
France  
Tel: (33) 1-30-60-70-00  
Fax: (33) 1-30-60-71-11

**Atmel 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

## Product Contact

---

**Web Site**  
[www.atmel.com](http://www.atmel.com)

**Technical Support**  
[avr@atmel.com](mailto:avr@atmel.com)

**Sales Contact**  
[www.atmel.com/contacts](http://www.atmel.com/contacts)

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

---

**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 ATMEL'S TERMS AND CONDITIONS OF SALE LOCATED ON ATMEL'S WEB SITE, 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 OF 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 product 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's products are not intended, authorized, or warranted for use as components in applications intended to support or sustain life.

© 2010 Atmel Corporation. All rights reserved. Atmel®, Atmel logo and combinations thereof, AVR® and others are registered trademarks or trademarks of Atmel Corporation or its subsidiaries. Other terms and product names may be trademarks of others.

## Looking for pricing, stock, or lifecycle information?

Click below to explore more details on WIN SOURCE:

 [View ATMEGA16-16PU on WIN SOURCE](#)

 [Atmel Information](#)

## Optimize Your Supply Chain with WIN SOURCE Solutions

-  Global Sourcing Solution
-  Obsolete Management
-  Cost Control Management
-  Shortage Management
-  Alternative Solution
-  Excess Inventory Management