## Features

- High-performance, Low-power 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
- Nonvolatile Program and Data Memories
  - 16K Bytes of In-System Self-Programmable Flash
    Endurance: 10,000 Write/Erase Cycles
  - Optional Boot Code Section with Independent Lock Bits
    In-System Programming by On-chip Boot Program
    True Read-While-Write Operation
  - 512 Bytes EEPROM
    Endurance: 100,000 Write/Erase Cycles
  - 1K Byte Internal SRAM
  - 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
  - 2.7 - 5.5V for ATmega16L
  - 4.5 - 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

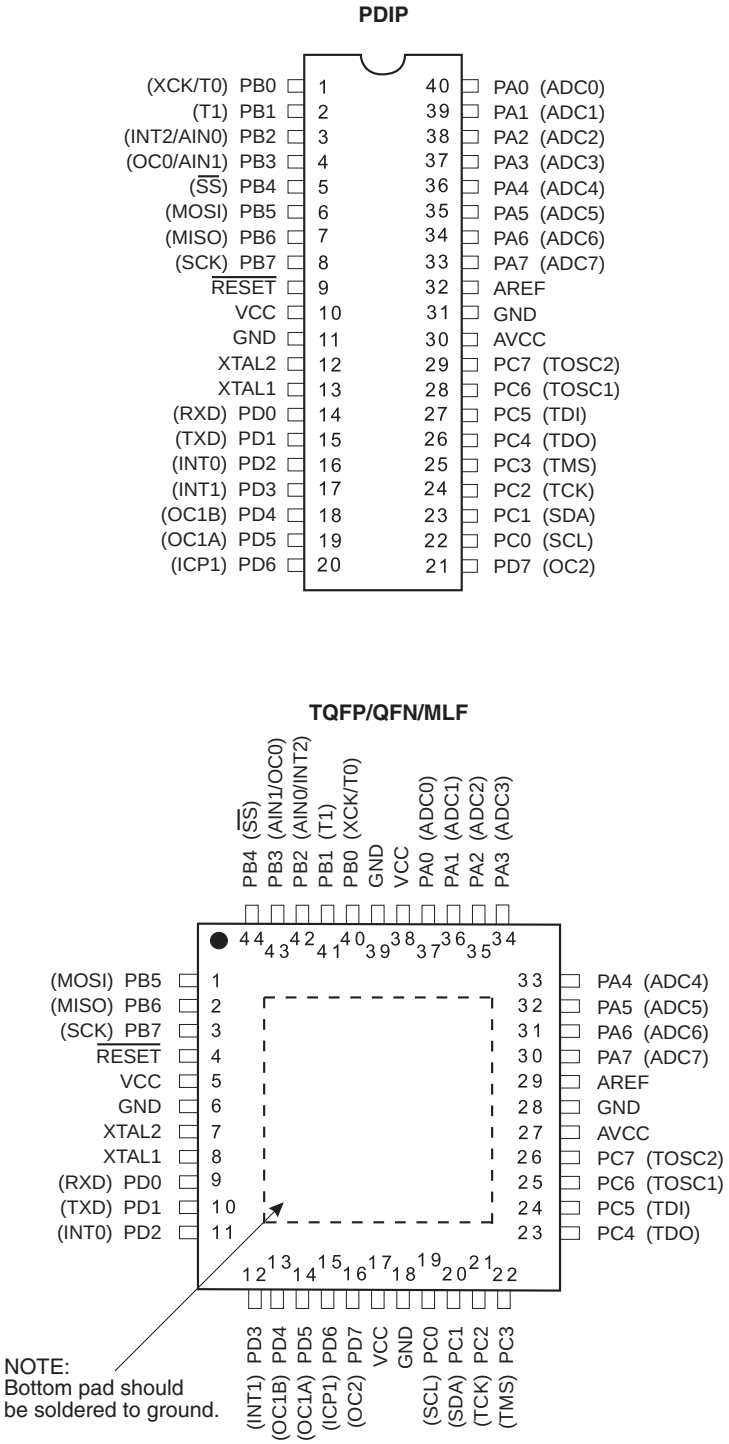
# 8-bit AVR® Microcontroller with 16K Bytes In-System Programmable Flash

# ATmega16
# ATmega16L

## Summary

## Pin Configurations

**Figure 1.** Pinout ATmega16




## Disclaimer

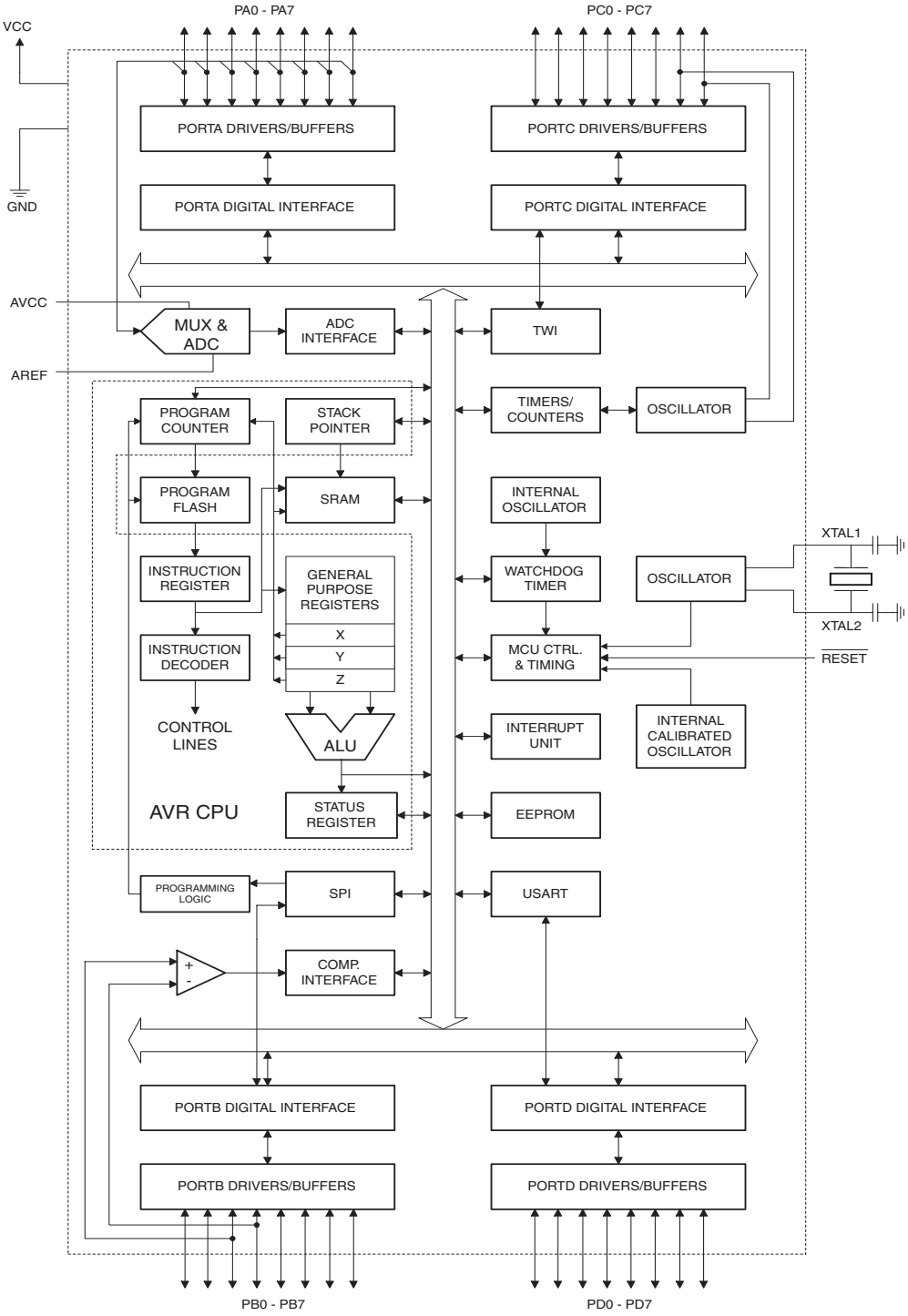
Typical values contained in this datasheet are based on simulations and characterization of other AVR microcontrollers manufactured on the same process technology. Min and Max values will be available after the device is characterized.

## 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: 16K bytes of In-System Programmable Flash Program memory with Read-While-Write capabilities, 512 bytes EEPROM, 1K byte 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 56.

**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 59.

**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 61.

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

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 36. 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.

## 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 | 7 |
| $3E ($5E) | SPH | – | – | – | – | – | SP10 | SP9 | SP8 | 10 |
| $3D ($5D) | SPL | SP7 | SP6 | SP5 | SP4 | SP3 | SP2 | SP1 | SP0 | 10 |
| $3C ($5C) | OCR0 | Timer/Counter0 Output Compare Register | | | | | | | | 83 |
| $3B ($5B) | GICR | INT1 | INT0 | INT2 | – | – | – | IVSEL | IVCE | 46, 67 |
| $3A ($5A) | GIFR | INTF1 | INTF0 | INTF2 | – | – | – | – | – | 68 |
| $39 ($59) | TIMSK | OCIE2 | TOIE2 | TICIE1 | OCIE1A | OCIE1B | TOIE1 | OCIE0 | TOIE0 | 83, 114, 132 |
| $38 ($58) | TIFR | OCF2 | TOV2 | ICF1 | OCF1A | OCF1B | TOV1 | OCF0 | TOV0 | 84, 115, 132 |
| $37 ($57) | SPMCR | SPMIE | RWWSB | – | RWWSRE | BLBSET | PGWRT | PGERS | SPMEN | 250 |
| $36 ($56) | TWCR | TWINT | TWEA | TWSTA | TWSTO | TWWC | TWEN | – | TWIE | 178 |
| $35 ($55) | MCUCR | SM2 | SE | SM1 | SM0 | ISC11 | ISC10 | ISC01 | ISC00 | 30, 66 |
| $34 ($54) | MCUCSR | JTD | ISC2 | – | JTRF | WDRF | BORF | EXTRF | PORF | 39, 67, 229 |
| $33 ($53) | TCCR0 | FOC0 | WGM00 | COM01 | COM00 | WGM01 | CS02 | CS01 | CS00 | 81 |
| $32 ($52) | TCNT0 | Timer/Counter0 (8 Bits) | | | | | | | | 83 |
| $31[1] ($51)[1] | OSCCAL | Oscillator Calibration Register | | | | | | | | 28 |
| | OCDR | On-Chip Debug Register | | | | | | | | 225 |
| $30 ($50) | SFIOR | ADTS2 | ADTS1 | ADTS0 | – | ACME | PUD | PSR2 | PSR10 | 55,86,133,199,219 |
| $2F ($4F) | TCCR1A | COM1A1 | COM1A0 | COM1B1 | COM1B0 | FOC1A | FOC1B | WGM11 | WGM10 | 109 |
| $2E ($4E) | TCCR1B | ICNC1 | ICES1 | – | WGM13 | WGM12 | CS12 | CS11 | CS10 | 112 |
| $2D ($4D) | TCNT1H | Timer/Counter1 – Counter Register High Byte | | | | | | | | 113 |
| $2C ($4C) | TCNT1L | Timer/Counter1 – Counter Register Low Byte | | | | | | | | 113 |
| $2B ($4B) | OCR1AH | Timer/Counter1 – Output Compare Register A High Byte | | | | | | | | 113 |
| $2A ($4A) | OCR1AL | Timer/Counter1 – Output Compare Register A Low Byte | | | | | | | | 113 |
| $29 ($49) | OCR1BH | Timer/Counter1 – Output Compare Register B High Byte | | | | | | | | 113 |
| $28 ($48) | OCR1BL | Timer/Counter1 – Output Compare Register B Low Byte | | | | | | | | 113 |
| $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 | 127 |
| $24 ($44) | TCNT2 | Timer/Counter2 (8 Bits) | | | | | | | | 129 |
| $23 ($43) | OCR2 | Timer/Counter2 Output Compare Register | | | | | | | | 129 |
| $22 ($42) | ASSR | – | – | – | – | AS2 | TCN2UB | OCR2UB | TCR2UB | 130 |
| $21 ($41) | WDTCR | – | – | – | WDTOE | WDE | WDP2 | WDP1 | WDP0 | 41 |
| $20[2] ($40)[2] | UBRRH | URSEL | – | – | – | UBRR[11:8] | | | | 165 |
| | UCSRC | URSEL | UMSEL | UPM1 | UPM0 | USBS | UCSZ1 | UCSZ0 | UCPOL | 164 |
| $1F ($3F) | EEARH | – | – | – | – | – | – | – | EEAR8 | 17 |
| $1E ($3E) | EEARL | EEPROM Address Register Low Byte | | | | | | | | 17 |
| $1D ($3D) | EEDR | EEPROM Data Register | | | | | | | | 17 |
| $1C ($3C) | EECR | – | – | – | – | EERIE | EEMWE | EEWE | EERE | 17 |
| $1B ($3B) | PORTA | PORTA7 | PORTA6 | PORTA5 | PORTA4 | PORTA3 | PORTA2 | PORTA1 | PORTA0 | 64 |
| $1A ($3A) | DDRA | DDA7 | DDA6 | DDA5 | DDA4 | DDA3 | DDA2 | DDA1 | DDA0 | 64 |
| $19 ($39) | PINA | PINA7 | PINA6 | PINA5 | PINA4 | PINA3 | PINA2 | PINA1 | PINA0 | 64 |
| $18 ($38) | PORTB | PORTB7 | PORTB6 | PORTB5 | PORTB4 | PORTB3 | PORTB2 | PORTB1 | PORTB0 | 64 |
| $17 ($37) | DDRB | DDB7 | DDB6 | DDB5 | DDB4 | DDB3 | DDB2 | DDB1 | DDB0 | 64 |
| $16 ($36) | PINB | PINB7 | PINB6 | PINB5 | PINB4 | PINB3 | PINB2 | PINB1 | PINB0 | 64 |
| $15 ($35) | PORTC | PORTC7 | PORTC6 | PORTC5 | PORTC4 | PORTC3 | PORTC2 | PORTC1 | PORTC0 | 65 |
| $14 ($34) | DDRC | DDC7 | DDC6 | DDC5 | DDC4 | DDC3 | DDC2 | DDC1 | DDC0 | 65 |
| $13 ($33) | PINC | PINC7 | PINC6 | PINC5 | PINC4 | PINC3 | PINC2 | PINC1 | PINC0 | 65 |
| $12 ($32) | PORTD | PORTD7 | PORTD6 | PORTD5 | PORTD4 | PORTD3 | PORTD2 | PORTD1 | PORTD0 | 65 |
| $11 ($31) | DDRD | DDD7 | DDD6 | DDD5 | DDD4 | DDD3 | DDD2 | DDD1 | DDD0 | 65 |
| $10 ($30) | PIND | PIND7 | PIND6 | PIND5 | PIND4 | PIND3 | PIND2 | PIND1 | PIND0 | 65 |
| $0F ($2F) | SPDR | SPI Data Register | | | | | | | | 140 |
| $0E ($2E) | SPSR | SPIF | WCOL | – | – | – | – | – | SPI2X | 140 |
| $0D ($2D) | SPCR | SPIE | SPE | DORD | MSTR | CPOL | CPHA | SPR1 | SPR0 | 138 |
| $0C ($2C) | UDR | USART I/O Data Register | | | | | | | | 161 |
| $0B ($2B) | UCSRA | RXC | TXC | UDRE | FE | DOR | PE | U2X | MPCM | 162 |
| $0A ($2A) | UCSRB | RXCIE | TXCIE | UDRIE | RXEN | TXEN | UCSZ2 | RXB8 | TXB8 | 163 |
| $09 ($29) | UBRRL | USART Baud Rate Register Low Byte | | | | | | | | 165 |
| $08 ($28) | ACSR | ACD | ACBG | ACO | ACI | ACIE | ACIC | ACIS1 | ACIS0 | 200 |
| $07 ($27) | ADMUX | REFS1 | REFS0 | ADLAR | MUX4 | MUX3 | MUX2 | MUX1 | MUX0 | 215 |
| $06 ($26) | ADCSRA | ADEN | ADSC | ADATE | ADIF | ADIE | ADPS2 | ADPS1 | ADPS0 | 217 |
| $05 ($25) | ADCH | ADC Data Register High Byte | | | | | | | | 218 |
| $04 ($24) | ADCL | ADC Data Register Low Byte | | | | | | | | 218 |
| $03 ($23) | TWDR | Two-wire Serial Interface Data Register | | | | | | | | 180 |
| $02 ($22) | TWAR | TWA6 | TWA5 | TWA4 | TWA3 | TWA2 | TWA1 | TWA0 | TWGCE | 180 |

| 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 | 179 |
| $00 ($20) | TWBR | Two-wire Serial Interface Bit Rate Register | | | | | | | | 178 |

Notes:
1. When the OCDEN Fuse is unprogrammed, the OSCCAL Register is always accessed on this address. Refer to the debugger specific documentation for details on how to use the OCDR Register.
2. Refer to the USART description for details on how to access UBRRH and UCSRC.
3. For compatibility with future devices, reserved bits should be written to zero if accessed. Reserved I/O memory addresses should never be written.
4. Some of the Status Flags are cleared by writing a logical one to them. Note that the CBI and SBI instructions will operate on all bits in the I/O Register, writing a one back into any flag read as set, thus clearing the flag. The CBI and SBI instructions work with registers $00 to $1F only.

# Instruction Set Summary

| Mnemonics | Operands | Description | Operation | Flags | #Clocks |
|---|---|---|---|---|---|
| **ARITHMETIC AND LOGIC INSTRUCTIONS** | | | | | |
| ADD | Rd, Rr | Add two Registers | Rd ← Rd + Rr | Z,C,N,V,H | 1 |
| ADC | Rd, Rr | Add with Carry two Registers | Rd ← Rd + Rr + C | Z,C,N,V,H | 1 |
| ADIW | Rdl,K | Add Immediate to Word | Rdh:Rdl ← Rdh:Rdl + K | Z,C,N,V,S | 2 |
| SUB | Rd, Rr | Subtract two Registers | Rd ← Rd - Rr | Z,C,N,V,H | 1 |
| SUBI | Rd, K | Subtract Constant from Register | Rd ← Rd - K | Z,C,N,V,H | 1 |
| SBC | Rd, Rr | Subtract with Carry two Registers | Rd ← Rd - Rr - C | Z,C,N,V,H | 1 |
| SBCI | Rd, K | Subtract with Carry Constant from Reg. | Rd ← Rd - K - C | Z,C,N,V,H | 1 |
| SBIW | Rdl,K | Subtract Immediate from Word | Rdh:Rdl ← Rdh:Rdl - K | Z,C,N,V,S | 2 |
| AND | Rd, Rr | Logical AND Registers | Rd ← Rd • Rr | Z,N,V | 1 |
| ANDI | Rd, K | Logical AND Register and Constant | Rd ← Rd • K | Z,N,V | 1 |
| OR | Rd, Rr | Logical OR Registers | Rd ← Rd v Rr | Z,N,V | 1 |
| ORI | Rd, K | Logical OR Register and Constant | Rd ← Rd v K | Z,N,V | 1 |
| EOR | Rd, Rr | Exclusive OR Registers | Rd ← Rd ⊕ Rr | Z,N,V | 1 |
| COM | Rd | One's Complement | Rd ← $FF − Rd | Z,C,N,V | 1 |
| NEG | Rd | Two's Complement | Rd ← $00 − Rd | Z,C,N,V,H | 1 |
| SBR | Rd,K | Set Bit(s) in Register | Rd ← Rd v K | Z,N,V | 1 |
| CBR | Rd,K | Clear Bit(s) in Register | Rd ← Rd • ($FF - K) | Z,N,V | 1 |
| INC | Rd | Increment | Rd ← Rd + 1 | Z,N,V | 1 |
| DEC | Rd | Decrement | Rd ← Rd − 1 | Z,N,V | 1 |
| TST | Rd | Test for Zero or Minus | Rd ← Rd • Rd | Z,N,V | 1 |
| CLR | Rd | Clear Register | Rd ← Rd ⊕ Rd | Z,N,V | 1 |
| SER | Rd | Set Register | Rd ← $FF | None | 1 |
| MUL | Rd, Rr | Multiply Unsigned | R1:R0 ← Rd x Rr | Z,C | 2 |
| MULS | Rd, Rr | Multiply Signed | R1:R0 ← Rd x Rr | Z,C | 2 |
| MULSU | Rd, Rr | Multiply Signed with Unsigned | R1:R0 ← Rd x Rr | Z,C | 2 |
| FMUL | Rd, Rr | Fractional Multiply Unsigned | R1:R0 ← (Rd x Rr) << 1 | Z,C | 2 |
| FMULS | Rd, Rr | Fractional Multiply Signed | R1:R0 ← (Rd x Rr) << 1 | Z,C | 2 |
| FMULSU | Rd, Rr | Fractional Multiply Signed with Unsigned | R1:R0 ← (Rd x Rr) << 1 | Z,C | 2 |
| **BRANCH INSTRUCTIONS** | | | | | |
| RJMP | k | Relative Jump | PC ← PC + k + 1 | None | 2 |
| IJMP | | Indirect Jump to (Z) | PC ← Z | None | 2 |
| JMP | k | Direct Jump | PC ← k | None | 3 |
| RCALL | k | Relative Subroutine Call | PC ← PC + k + 1 | None | 3 |
| ICALL | | Indirect Call to (Z) | PC ← Z | None | 3 |
| CALL | k | Direct Subroutine Call | PC ← k | None | 4 |
| RET | | Subroutine Return | PC ← STACK | None | 4 |
| RETI | | Interrupt Return | PC ← STACK | I | 4 |
| CPSE | Rd,Rr | Compare, Skip if Equal | if (Rd = Rr) PC ← 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 ← PC + 2 or 3 | None | 1 / 2 / 3 |
| SBRS | Rr, b | Skip if Bit in Register is Set | if (Rr(b)=1) PC ← PC + 2 or 3 | None | 1 / 2 / 3 |
| SBIC | P, b | Skip if Bit in I/O Register Cleared | if (P(b)=0) PC ← 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 ← PC + 2 or 3 | None | 1 / 2 / 3 |
| BRBS | s, k | Branch if Status Flag Set | if (SREG(s) = 1) then PC←PC+k + 1 | None | 1 / 2 |
| BRBC | s, k | Branch if Status Flag Cleared | if (SREG(s) = 0) then PC←PC+k + 1 | None | 1 / 2 |
| BREQ | k | Branch if Equal | if (Z = 1) then PC ← PC + k + 1 | None | 1 / 2 |
| BRNE | k | Branch if Not Equal | if (Z = 0) then PC ← PC + k + 1 | None | 1 / 2 |
| BRCS | k | Branch if Carry Set | if (C = 1) then PC ← PC + k + 1 | None | 1 / 2 |
| BRCC | k | Branch if Carry Cleared | if (C = 0) then PC ← PC + k + 1 | None | 1 / 2 |
| BRSH | k | Branch if Same or Higher | if (C = 0) then PC ← PC + k + 1 | None | 1 / 2 |
| BRLO | k | Branch if Lower | if (C = 1) then PC ← PC + k + 1 | None | 1 / 2 |
| BRMI | k | Branch if Minus | if (N = 1) then PC ← PC + k + 1 | None | 1 / 2 |
| BRPL | k | Branch if Plus | if (N = 0) then PC ← PC + k + 1 | None | 1 / 2 |
| BRGE | k | Branch if Greater or Equal, Signed | if (N ⊕ V= 0) then PC ← PC + k + 1 | None | 1 / 2 |
| BRLT | k | Branch if Less Than Zero, Signed | if (N ⊕ V= 1) then PC ← PC + k + 1 | None | 1 / 2 |
| BRHS | k | Branch if Half Carry Flag Set | if (H = 1) then PC ← PC + k + 1 | None | 1 / 2 |
| BRHC | k | Branch if Half Carry Flag Cleared | if (H = 0) then PC ← PC + k + 1 | None | 1 / 2 |
| BRTS | k | Branch if T Flag Set | if (T = 1) then PC ← PC + k + 1 | None | 1 / 2 |
| BRTC | k | Branch if T Flag Cleared | if (T = 0) then PC ← PC + k + 1 | None | 1 / 2 |
| BRVS | k | Branch if Overflow Flag is Set | if (V = 1) then PC ← PC + k + 1 | None | 1 / 2 |
| BRVC | k | Branch if Overflow Flag is Cleared | if (V = 0) then PC ← PC + k + 1 | None | 1 / 2 |

| 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 |
| **DATA TRANSFER INSTRUCTIONS** | | | | | |
| 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 |
| **BIT AND BIT-TEST INSTRUCTIONS** | | | | | |
| 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

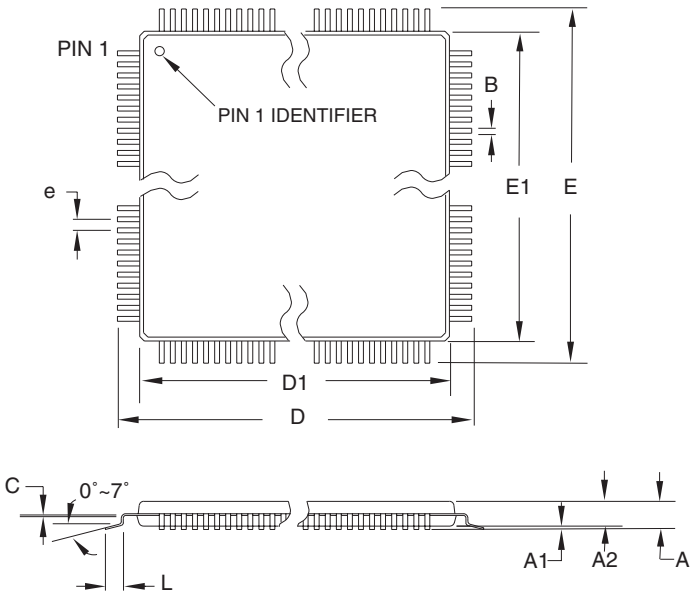
| Speed (MHz) | Power Supply | Ordering Code | Package | Operation Range |
|---|---|---|---|---|
| 8 | 2.7 - 5.5V | ATmega16L-8AC<br>ATmega16L-8PC<br>ATmega16L-8MC | 44A<br>40P6<br>44M1 | Commercial<br>(0°C to 70°C) |
| | | ATmega16L-8AI<br>ATmega16L-8AU(1)<br>ATmega16L-8PI<br>ATmega16L-8PU(1)<br>ATmega16L-8MI<br>ATmega16L-8MU(1) | 44A<br>44A<br>40P6<br>40P6<br>44M1<br>44M1 | Industrial<br>(-40°C to 85°C) |
| 16 | 4.5 - 5.5V | ATmega16-16AC<br>ATmega16-16PC<br>ATmega16-16MC | 44A<br>40P6<br>44M1 | Commercial<br>(0°C to 70°C) |
| | | ATmega16-16AI<br>ATmega16-16AU(1)<br>ATmega16-16PI<br>ATmega16-16PU(1)<br>ATmega16-16MI<br>ATmega16-16MU(1) | 44A<br>44A<br>40P6<br>40P6<br>44M1<br>44M1 | Industrial<br>(-40°C to 85°C) |

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

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

# Packaging Information

## 44A

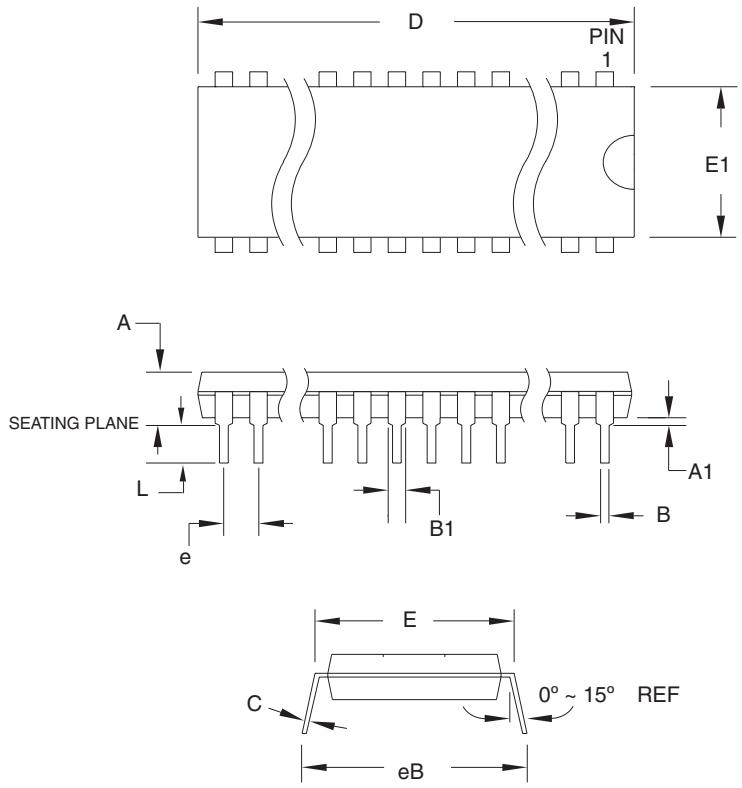



**COMMON DIMENSIONS**
(Unit of Measure = mm)

| 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 0.25 mm per side. Dimensions D1 and E1 are maximum plastic body size dimensions including mold mismatch.
3. Lead coplanarity is 0.10 mm maximum.

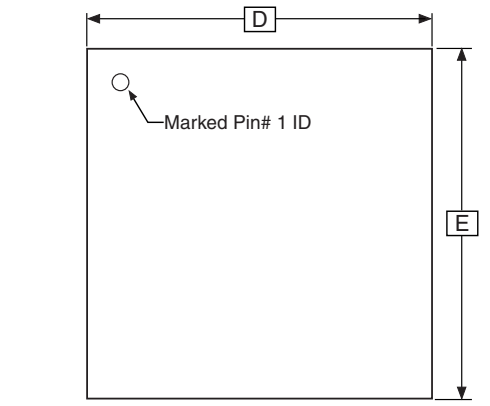
## 40P6



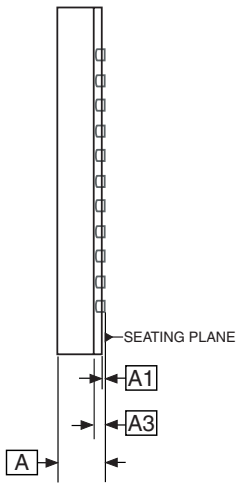

**COMMON DIMENSIONS**
(Unit of Measure = mm)

| 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 | | | |

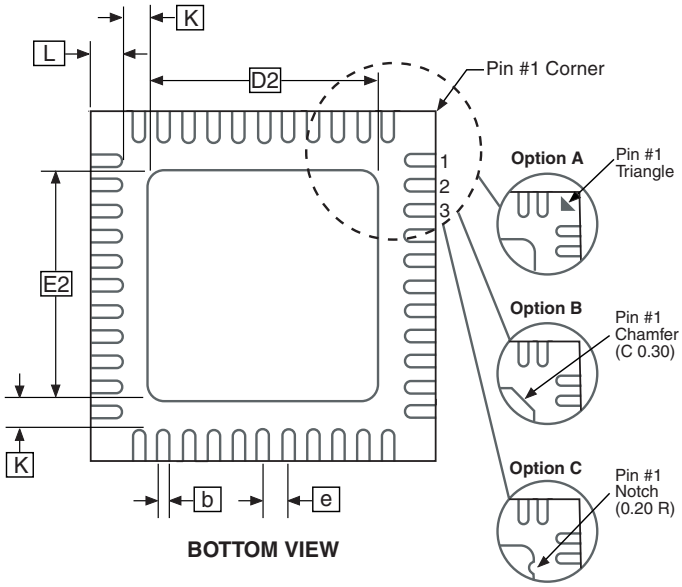
Notes: 1. This package conforms to JEDEC reference MS-011, Variation AC.
2. Dimensions D and E1 do not include mold Flash or Protrusion. Mold Flash or Protrusion shall not exceed 0.25 mm (0.010").




TOP VIEW




SIDE VIEW




BOTTOM VIEW

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

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

| SYMBOL | MIN | NOM | MAX | NOTE |
|---|---|---|---|---|
| A | 0.80 | 0.90 | 1.00 | |
| A1 | – | 0.02 | 0.05 | |
| A3 | | 0.25 REF | | |
| b | 0.18 | 0.23 | 0.30 | |
| D | 6.90 | 7.00 | 7.10 | |
| D2 | 5.00 | 5.20 | 5.40 | |
| E | 6.90 | 7.00 | 7.10 | |
| E2 | 5.00 | 5.20 | 5.40 | |
| e | | 0.50 BSC | | |
| L | 0.59 | 0.64 | 0.69 | |
| K | 0.20 | 0.26 | 0.41 | |

## 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**

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 theAnalog Comparator before the first conversion.

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

   If one of the timer registers which is synchronized to the asynchronous timer2 clock is written in the cycle before a overflow interrupt occurs, the interrupt may be lost.

   **Problem Fix/Workaround**

   Always check that the Timer2 Timer/Counter register, TCNT2, does not have the value 0xFF before writing the Timer2 Control Register, TCCR2, or Output Compare Register, OCR2

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 fist device in the chain.

### 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**

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 theAnalog Comparator before the first conversion.

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

If one of the timer registers which is synchronized to the asynchronous timer2 clock is written in the cycle before a overflow interrupt occurs, the interrupt may be lost.

**Problem Fix/Workaround**

Always check that the Timer2 Timer/Counter register, TCNT2, does not have the value 0xFF before writing the Timer2 Control Register, TCCR2, or Output Compare Register, OCR2

**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 fist device in the chain.

## 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**

**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 theAnalog Comparator before the first conversion.

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

If one of the timer registers which is synchronized to the asynchronous timer2 clock is written in the cycle before a overflow interrupt occurs, the interrupt may be lost.

**Problem Fix/Workaround**

Always check that the Timer2 Timer/Counter register, TCNT2, does not have the value 0xFF before writing the Timer2 Control Register, TCCR2, or Output Compare Register, OCR2

**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 fist device in the chain.

## 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**

**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 theAnalog Comparator before the first conversion.

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

If one of the timer registers which is synchronized to the asynchronous timer2 clock is written in the cycle before a overflow interrupt occurs, the interrupt may be lost.

**Problem Fix/Workaround**

Always check that the Timer2 Timer/Counter register, TCNT2, does not have the value 0xFF before writing the Timer2 Control Register, TCCR2, or Output Compare Register, OCR2

**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 fist device in the chain.

## ATmega16(L) Rev. I

- **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**

**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 theAnalog Comparator before the first conversion.

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

   If one of the timer registers which is synchronized to the asynchronous timer2 clock is written in the cycle before a overflow interrupt occurs, the interrupt may be lost.

   **Problem Fix/Workaround**

   Always check that the Timer2 Timer/Counter register, TCNT2, does not have the value 0xFF before writing the Timer2 Control Register, TCCR2, or Output Compare Register, OCR2

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 fist device in the chain.

## 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**

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 theAnalog Comparator before the first conversion.

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

   If one of the timer registers which is synchronized to the asynchronous timer2 clock is written in the cycle before a overflow interrupt occurs, the interrupt may be lost.

   **Problem Fix/Workaround**

   Always check that the Timer2 Timer/Counter register, TCNT2, does not have the value 0xFF before writing the Timer2 Control Register, TCCR2, or Output Compare Register, OCR2

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 fist device in the chain.

## 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. 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 112, Table 47 on page 113, Table 50 on page 129 and Table 52 on page 130.**
4. **Updated C code example in "USART Initialization" on page 150.**
5. **Updated "Errata" on page 343.**

### Rev. 2466M-04/06

1. **Updated typos.**
2. **Updated "Serial Peripheral Interface – SPI" on page 136.**
3. **Updated Table 86 on page 222, Table 116 on page 279 ,Table 121 on page 298 and Table 122 on page 300.**

### Rev. 2466L-06/05

1. **Updated note in "Bit Rate Generator Unit" on page 179.**
2. **Updated values for $V_{INT}$ in "ADC Characteristics" on page 300.**
3. **Updated "Serial Programming Instruction set" on page 279.**
4. **Updated USART init C-code example in "USART" on page 145.**

### Rev. 2466K-04/05

1. **Updated "Ordering Information" on page 11.**
2. **MLF-package alternative changed to "Quad Flat No-Lead/Micro Lead Frame Package QFN/MLF".**
3. **Updated "Electrical Characteristics" on page 294.**

### Rev. 2466J-10/04

1. **Updated "Ordering Information" on page 11.**

### 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 211, Table 116 on page 279, and Table 119 on page 296.**
3. **Updated "Pinout ATmega16" on page 2.**
4. **Updated features in "Analog to Digital Converter" on page 205.**
5. **Updated "Version" on page 230.**
6. **Updated "Calibration Byte" on page 264.**

7. Added “Page Size” on page 265.

**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 218 and Table 115 on page 279.
7. Updated “Test Access Port – TAP” on page 223 regarding JTAGEN.
8. Updated description for the JTD bit on page 232.
9. Added note 2 to Figure 126 on page 255.
10. Added a note regarding JTAGEN fuse to Table 105 on page 263.
11. Updated Absolute Maximum Ratings* and DC Characteristics in “Electrical Characteristics” on page 294.
12. Updated “ATmega16 Typical Characteristics” on page 302.
13. Fixed typo for 16 MHz QFN/MLF package in “Ordering Information” on page 11.
14. Added a proposal for solving problems regarding the JTAG instruction IDCODE in “Errata” on page 15.

**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 291 and “Programming the EEPROM” on page 292.
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 256 about writing to the EEPROM during an SPM Page Load.**

8. **Removed ADHSM completely.**

9. **Added Table 73, "TWI Bit Rate Prescaler," on page 183 to describe the TWPS bits in the "TWI Status Register – TWSR" on page 182.**

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 296.**

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 208.**

16. **Added updated "Packaging Information" on page 12.**

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

1. **Updated "DC Characteristics" on page 294.**

**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 112, Table 46 on page 112, Table 59 on page 144, Table 67 on page 168, Table 90 on page 237, Table 102 on page 261, "DC Characteristics" on page 294, Table 119 on page 296, Table 121 on page 298, and Table 122 on page 300.**

3. **Updated "Errata" on page 15.**

**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 30.

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 294, Table 119 on page 296, Table 121 on page 298, and Table 122 on page 300.

5. **Updated TWI Chapter.**

Added the note at the end of the “Bit Rate Generator Unit” on page 179.

6. **Corrected description of ADSC bit in “ADC Control and Status Register A – ADCSRA” on page 220.**

7. **Improved description on how to do a polarity check of the ADC doff results in “ADC Conversion Result” on page 217.**

8. **Added JTAG version number for rev. H in Table 87 on page 230.**

9. **Added not regarding OCDEN Fuse below Table 105 on page 263.**

10. **Updated Programming Figures:**

    Figure 127 on page 265 and Figure 136 on page 277 are updated to also reflect that AVCC must be connected during Programming mode. Figure 131 on page 273 added to illustrate how to program the fuses.

11. **Added a note regarding usage of the “PROG_PAGELOAD ($6)” on page 283 and “PROG_PAGEREAD ($7)” on page 283.**

12. **Removed alternative algortihm for leaving JTAG Programming mode.**

    See “Leaving Programming Mode” on page 291.

13. **Added Calibrated RC Oscillator characterization curves in section “ATmega16 Typical Characteristics” on page 302.**

14. **Corrected ordering code for QFN/MLF package (16MHz) in “Ordering Information” on page 11.**

15. **Corrected Table 90, “Scan Signals for the Oscillators(1)(2)(3),” on page 237.**