

Features

- High Performance, Low Power AVR® 8-Bit Microcontroller
- Advanced RISC Architecture
 - 130 Powerful Instructions – Most Single Clock Cycle Execution
 - 32 x 8 General Purpose Working Registers
 - Fully Static Operation
 - Up to 16 MIPS Throughput at 16 MHz
 - On-Chip 2-cycle Multiplier
- Non-volatile Program and Data Memories
 - 16K bytes of In-System Self-Programmable Flash
Endurance: 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 Prescaler and Compare Mode
 - 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
 - Programmable Serial USART
 - Master/Slave SPI Serial Interface
 - Universal Serial Interface with Start Condition Detector
 - Programmable Watchdog Timer with Separate On-chip Oscillator
 - On-chip Analog Comparator
 - Interrupt and Wake-up on Pin Change
- Special Microcontroller Features
 - Power-on Reset and Programmable Brown-out Detection
 - Internal Calibrated Oscillator
 - External and Internal Interrupt Sources
 - Five Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, and Standby
- I/O and Packages
 - 53 Programmable I/O Lines 2573GS
 - 64-lead TQFP and 64-pad QFN/MLF
- Speed Grade:
 - ATmega165V: 0 - 4 MHz @ 1.8 - 5.5V, 0 - 8 MHz @ 2.7 - 5.5V
 - ATmega165: 0 - 8 MHz @ 2.7 - 5.5V, 0 - 16 MHz @ 4.5 - 5.5V
- Temperature range:
 - -40°C to 85°C Industrial
- Ultra-Low Power Consumption
 - Active Mode:
 - 1 MHz, 1.8V: 350µA
 - 32 kHz, 1.8V: 20µA (including Oscillator)
 - Power-down Mode:
 - 0.1µA at 1.8V



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

ATmega165V
ATmega165

Preliminary Summary

Notice:
Not recommended in new
designs.

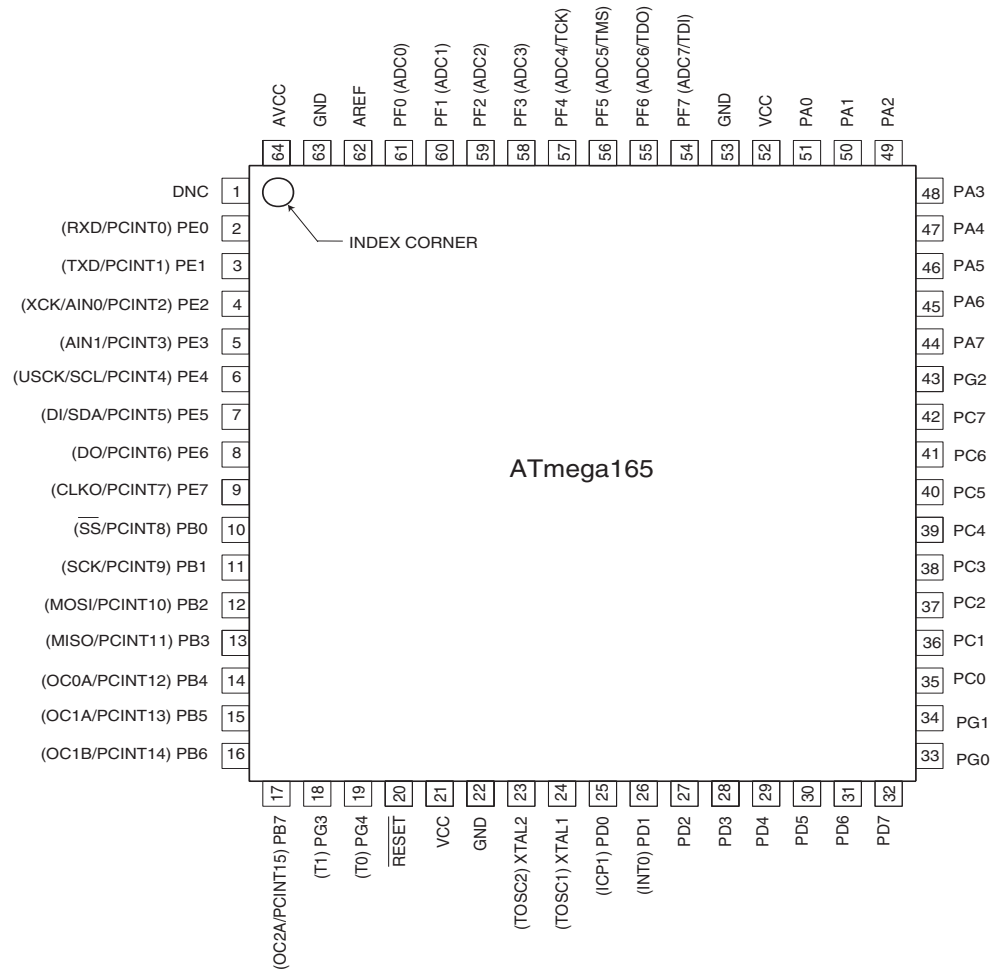
2573GS-AVR-07/09



Note: This is a summary document. A complete document is available on our Web site at www.atmel.com.

Pin Configurations

Figure 1. Pinout ATmega165



Note: The large center pad underneath the QFN/MLF packages is made of metal and internally connected to GND. It should be soldered or glued to the board to ensure good mechanical stability. If the center pad is left unconnected, the package might loosen from the board.

Disclaimer

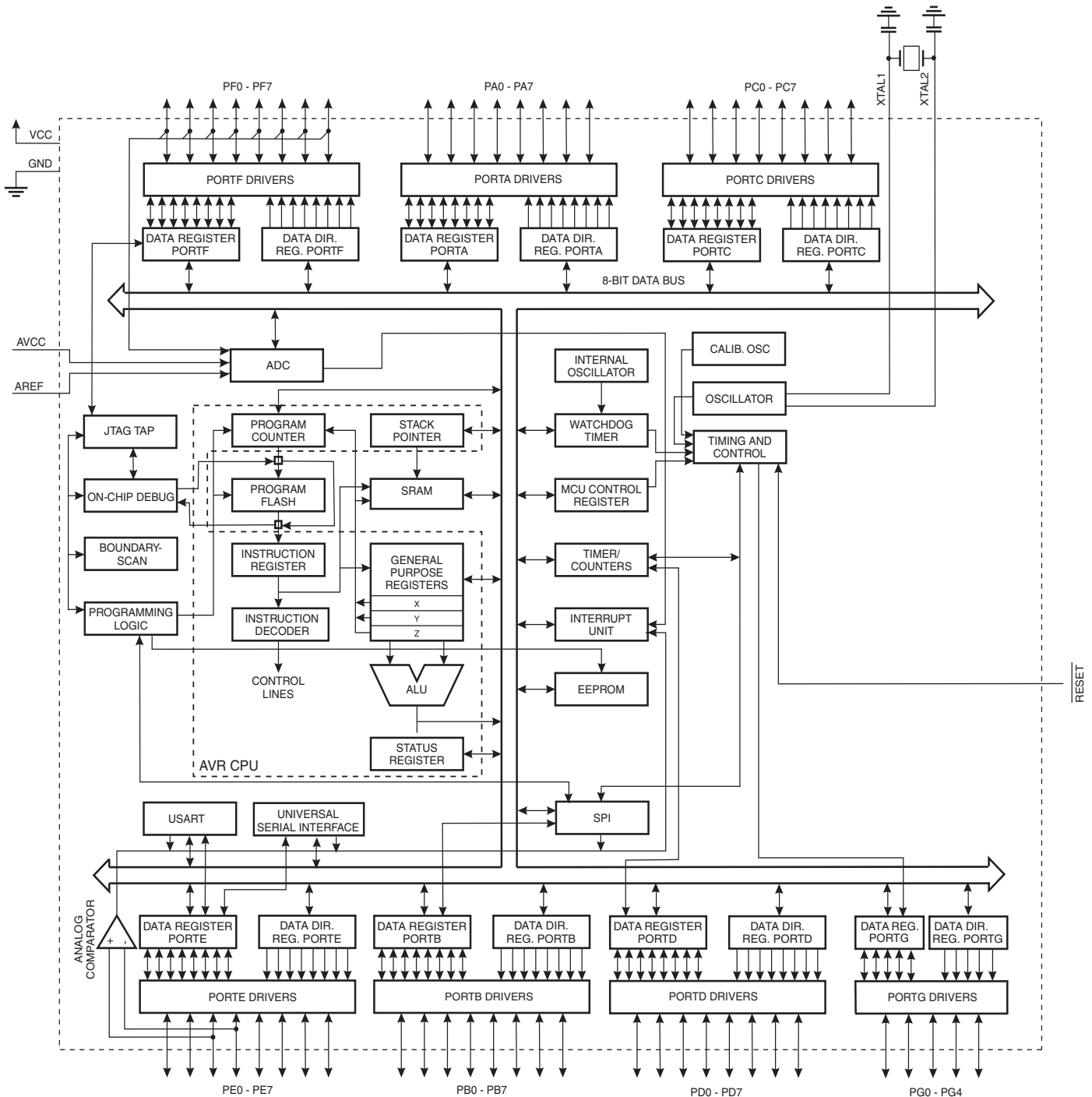
Typical values contained in this datasheet are based on simulations and characterization of other AVR microcontrollers manufactured on the same process technology. Min and Max values will be available after the device is characterized.

Overview

The ATmega165 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 ATmega165 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 ATmega165 provides the following features: 16K bytes of In-System Programmable Flash with Read-While-Write capabilities, 512 bytes EEPROM, 1K byte SRAM, 53 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, Universal Serial Interface with Start Condition Detector, an 8-channel, 10-bit ADC, a programmable Watchdog Timer with internal Oscillator, an SPI serial port, and five software selectable power saving modes. The Idle mode stops the CPU while allowing the SRAM, Timer/Counters, SPI port, and interrupt system to continue functioning. The Power-down mode saves the register contents but freezes the Oscillator, disabling all other chip functions until the next interrupt or hardware reset. In Power-save mode, the asynchronous timer continues to run, allowing the user to maintain a timer base while the rest of the device is sleeping. The ADC Noise Reduction mode stops the CPU and all I/O modules except asynchronous timer and ADC, to minimize switching noise during ADC conversions. In Standby mode, the crystal/resonator Oscillator is running while the rest of the device is sleeping. This allows very fast start-up combined with low-power consumption.

The device is manufactured using Atmel's high density non-volatile memory technology. The On-chip ISP Flash allows the program memory to be reprogrammed In-System through an SPI serial interface, by a conventional non-volatile memory programmer, or by an On-chip Boot program running on the AVR core. The Boot program can use any interface to download the application program in the Application Flash memory. Software in the Boot Flash section will continue to run while the Application Flash section is updated, providing true Read-While-Write operation. By combining an 8-bit RISC CPU with In-System Self-Programmable Flash on a monolithic chip, the Atmel ATmega165 is a powerful microcontroller that provides a highly flexible and cost effective solution to many embedded control applications.

The ATmega165 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) | <p>Port A is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port A output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port A pins that are externally pulled low will source current if the 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.</p> |
| Port B (PB7..PB0) | <p>Port B is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port B output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port B pins that are externally pulled low will source current if the pull-up resistors are activated. The Port B pins are tri-stated when a reset condition becomes active, even if the clock is not running.</p> <p>Port B has better driving capabilities than the other ports.</p> <p>Port B also serves the functions of various special features of the ATmega165 as listed on page 62.</p> |
| Port C (PC7..PC0) | <p>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.</p> |
| Port D (PD7..PD0) | <p>Port D is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port D output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port D pins that are externally pulled low will source current if the pull-up resistors are activated. The Port D pins are tri-stated when a reset condition becomes active, even if the clock is not running.</p> <p>Port D also serves the functions of various special features of the ATmega165 as listed on page 65.</p> |
| Port E (PE7..PE0) | <p>Port E is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port E output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port E pins that are externally pulled low will source current if the pull-up resistors are activated. The Port E pins are tri-stated when a reset condition becomes active, even if the clock is not running.</p> <p>Port E also serves the functions of various special features of the ATmega165 as listed on page 66.</p> |
| Port F (PF7..PF0) | <p>Port F serves as the analog inputs to the A/D Converter.</p> <p>Port F 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 F output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port F pins that are externally pulled low will source current if the pull-up resistors are activated. The Port F 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 resis-</p> |

tors on pins PF7(TDI), PF5(TMS), and PF4(TCK) will be activated even if a reset occurs.

Port F also serves the functions of the JTAG interface.

Port G (PG4..PG0)

Port G is a 5-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port G output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port G pins that are externally pulled low will source current if the pull-up resistors are activated. The Port G pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port G also serves the functions of various special features of the ATmega165 as listed on page 66.

$\overline{\text{RESET}}$

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 16 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 F 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

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

Register Summary

| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Page |
|---------|----------|-------------------------------|-------|-------|-------|-------|-------|-------|-------|------|
| (0xFF) | Reserved | – | – | – | – | – | – | – | – | |
| (0xFE) | Reserved | – | – | – | – | – | – | – | – | |
| (0xFD) | Reserved | – | – | – | – | – | – | – | – | |
| (0xFC) | Reserved | – | – | – | – | – | – | – | – | |
| (0xFB) | Reserved | – | – | – | – | – | – | – | – | |
| (0xFA) | Reserved | – | – | – | – | – | – | – | – | |
| (0xF9) | Reserved | – | – | – | – | – | – | – | – | |
| (0xF8) | Reserved | – | – | – | – | – | – | – | – | |
| (0xF7) | Reserved | – | – | – | – | – | – | – | – | |
| (0xF6) | Reserved | – | – | – | – | – | – | – | – | |
| (0xF5) | Reserved | – | – | – | – | – | – | – | – | |
| (0xF4) | Reserved | – | – | – | – | – | – | – | – | |
| (0xF3) | Reserved | – | – | – | – | – | – | – | – | |
| (0xF2) | Reserved | – | – | – | – | – | – | – | – | |
| (0xF1) | Reserved | – | – | – | – | – | – | – | – | |
| (0xF0) | Reserved | – | – | – | – | – | – | – | – | |
| (0xEF) | Reserved | – | – | – | – | – | – | – | – | |
| (0xEE) | Reserved | – | – | – | – | – | – | – | – | |
| (0xED) | Reserved | – | – | – | – | – | – | – | – | |
| (0xEC) | Reserved | – | – | – | – | – | – | – | – | |
| (0xEB) | Reserved | – | – | – | – | – | – | – | – | |
| (0xEA) | Reserved | – | – | – | – | – | – | – | – | |
| (0xE9) | Reserved | – | – | – | – | – | – | – | – | |
| (0xE8) | Reserved | – | – | – | – | – | – | – | – | |
| (0xE7) | Reserved | – | – | – | – | – | – | – | – | |
| (0xE6) | Reserved | – | – | – | – | – | – | – | – | |
| (0xE5) | Reserved | – | – | – | – | – | – | – | – | |
| (0xE4) | Reserved | – | – | – | – | – | – | – | – | |
| (0xE3) | Reserved | – | – | – | – | – | – | – | – | |
| (0xE2) | Reserved | – | – | – | – | – | – | – | – | |
| (0xE1) | Reserved | – | – | – | – | – | – | – | – | |
| (0xE0) | Reserved | – | – | – | – | – | – | – | – | |
| (0xDF) | Reserved | – | – | – | – | – | – | – | – | |
| (0xDE) | Reserved | – | – | – | – | – | – | – | – | |
| (0xDD) | Reserved | – | – | – | – | – | – | – | – | |
| (0xDC) | Reserved | – | – | – | – | – | – | – | – | |
| (0xDB) | Reserved | – | – | – | – | – | – | – | – | |
| (0xDA) | Reserved | – | – | – | – | – | – | – | – | |
| (0xD9) | Reserved | – | – | – | – | – | – | – | – | |
| (0xD8) | Reserved | – | – | – | – | – | – | – | – | |
| (0xD7) | Reserved | – | – | – | – | – | – | – | – | |
| (0xD6) | Reserved | – | – | – | – | – | – | – | – | |
| (0xD5) | Reserved | – | – | – | – | – | – | – | – | |
| (0xD4) | Reserved | – | – | – | – | – | – | – | – | |
| (0xD3) | Reserved | – | – | – | – | – | – | – | – | |
| (0xD2) | Reserved | – | – | – | – | – | – | – | – | |
| (0xD1) | Reserved | – | – | – | – | – | – | – | – | |
| (0xD0) | Reserved | – | – | – | – | – | – | – | – | |
| (0xCF) | Reserved | – | – | – | – | – | – | – | – | |
| (0xCE) | Reserved | – | – | – | – | – | – | – | – | |
| (0xCD) | Reserved | – | – | – | – | – | – | – | – | |
| (0xCC) | Reserved | – | – | – | – | – | – | – | – | |
| (0xCB) | Reserved | – | – | – | – | – | – | – | – | |
| (0xCA) | Reserved | – | – | – | – | – | – | – | – | |
| (0xC9) | Reserved | – | – | – | – | – | – | – | – | |
| (0xC8) | Reserved | – | – | – | – | – | – | – | – | |
| (0xC7) | Reserved | – | – | – | – | – | – | – | – | |
| (0xC6) | UDR | USART I/O Data Register | | | | | | | | 166 |
| (0xC5) | UBRRH | USART Baud Rate Register High | | | | | | | | 170 |
| (0xC4) | UBRRL | USART Baud Rate Register Low | | | | | | | | 170 |
| (0xC3) | Reserved | – | – | – | – | – | – | – | – | |
| (0xC2) | UCSRC | – | UMSEL | UPM1 | UPM0 | USBS | UCSZ1 | UCSZ0 | UCPOL | 166 |
| (0xC1) | UCSRB | RXCIE | TXCIE | UDRIE | RXEN | TXEN | UCSZ2 | RXB8 | TXB8 | 166 |
| (0xC0) | UCSRA | RXC | TXC | UDRE | FE | DOR | UPE | U2X | MPCM | 166 |

| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Page |
|---------|----------|--|--------|--------|--------|---------|---------|---------|---------|------|
| (0xBF) | Reserved | – | – | – | – | – | – | – | – | |
| (0xBE) | Reserved | – | – | – | – | – | – | – | – | |
| (0xBD) | Reserved | – | – | – | – | – | – | – | – | |
| (0xBC) | Reserved | – | – | – | – | – | – | – | – | |
| (0xBB) | Reserved | – | – | – | – | – | – | – | – | |
| (0xBA) | USIDR | USI Data Register | | | | | | | | 181 |
| (0xB9) | USISR | USISIF | USIOIF | USIPF | USIDC | USICNT3 | USICNT2 | USICNT1 | USICNT0 | 182 |
| (0xB8) | USICR | USISIE | USIOIE | USIWM1 | USIWM0 | USICS1 | USICS0 | USICLK | USITC | 183 |
| (0xB7) | Reserved | – | – | – | – | – | – | – | – | |
| (0xB6) | ASSR | – | – | – | EXCLK | AS2 | TCN2UB | OCR2UB | TCR2UB | 134 |
| (0xB5) | Reserved | – | – | – | – | – | – | – | – | |
| (0xB4) | Reserved | – | – | – | – | – | – | – | – | |
| (0xB3) | OCR2A | Timer/Counter2 Output Compare Register A | | | | | | | | 133 |
| (0xB2) | TCNT2 | Timer/Counter2 (8-bit) | | | | | | | | 133 |
| (0xB1) | Reserved | – | – | – | – | – | – | – | – | |
| (0xB0) | TCCR2A | FOC2A | WGM20 | COM2A1 | COM2A0 | WGM21 | CS22 | CS21 | CS20 | 131 |
| (0xAF) | Reserved | – | – | – | – | – | – | – | – | |
| (0xAE) | Reserved | – | – | – | – | – | – | – | – | |
| (0xAD) | Reserved | – | – | – | – | – | – | – | – | |
| (0xAC) | Reserved | – | – | – | – | – | – | – | – | |
| (0xAB) | Reserved | – | – | – | – | – | – | – | – | |
| (0xAA) | Reserved | – | – | – | – | – | – | – | – | |
| (0xA9) | Reserved | – | – | – | – | – | – | – | – | |
| (0xA8) | Reserved | – | – | – | – | – | – | – | – | |
| (0xA7) | Reserved | – | – | – | – | – | – | – | – | |
| (0xA6) | Reserved | – | – | – | – | – | – | – | – | |
| (0xA5) | Reserved | – | – | – | – | – | – | – | – | |
| (0xA4) | Reserved | – | – | – | – | – | – | – | – | |
| (0xA3) | Reserved | – | – | – | – | – | – | – | – | |
| (0xA2) | Reserved | – | – | – | – | – | – | – | – | |
| (0xA1) | Reserved | – | – | – | – | – | – | – | – | |
| (0xA0) | Reserved | – | – | – | – | – | – | – | – | |
| (0x9F) | Reserved | – | – | – | – | – | – | – | – | |
| (0x9E) | Reserved | – | – | – | – | – | – | – | – | |
| (0x9D) | Reserved | – | – | – | – | – | – | – | – | |
| (0x9C) | Reserved | – | – | – | – | – | – | – | – | |
| (0x9B) | Reserved | – | – | – | – | – | – | – | – | |
| (0x9A) | Reserved | – | – | – | – | – | – | – | – | |
| (0x99) | Reserved | – | – | – | – | – | – | – | – | |
| (0x98) | Reserved | – | – | – | – | – | – | – | – | |
| (0x97) | Reserved | – | – | – | – | – | – | – | – | |
| (0x96) | Reserved | – | – | – | – | – | – | – | – | |
| (0x95) | Reserved | – | – | – | – | – | – | – | – | |
| (0x94) | Reserved | – | – | – | – | – | – | – | – | |
| (0x93) | Reserved | – | – | – | – | – | – | – | – | |
| (0x92) | Reserved | – | – | – | – | – | – | – | – | |
| (0x91) | Reserved | – | – | – | – | – | – | – | – | |
| (0x90) | Reserved | – | – | – | – | – | – | – | – | |
| (0x8F) | Reserved | – | – | – | – | – | – | – | – | |
| (0x8E) | Reserved | – | – | – | – | – | – | – | – | |
| (0x8D) | Reserved | – | – | – | – | – | – | – | – | |
| (0x8C) | Reserved | – | – | – | – | – | – | – | – | |
| (0x8B) | OCR1BH | Timer/Counter1 - Output Compare Register B High Byte | | | | | | | | 117 |
| (0x8A) | OCR1BL | Timer/Counter1 - Output Compare Register B Low Byte | | | | | | | | 117 |
| (0x89) | OCR1AH | Timer/Counter1 - Output Compare Register A High Byte | | | | | | | | 117 |
| (0x88) | OCR1AL | Timer/Counter1 - Output Compare Register A Low Byte | | | | | | | | 117 |
| (0x87) | ICR1H | Timer/Counter1 - Input Capture Register High Byte | | | | | | | | 118 |
| (0x86) | ICR1L | Timer/Counter1 - Input Capture Register Low Byte | | | | | | | | 118 |
| (0x85) | TCNT1H | Timer/Counter1 - Counter Register High Byte | | | | | | | | 117 |
| (0x84) | TCNT1L | Timer/Counter1 - Counter Register Low Byte | | | | | | | | 117 |
| (0x83) | Reserved | – | – | – | – | – | – | – | – | |
| (0x82) | TCCR1C | FOC1A | FOC1B | – | – | – | – | – | – | 116 |
| (0x81) | TCCR1B | ICNC1 | ICES1 | – | WGM13 | WGM12 | CS12 | CS11 | CS10 | 115 |
| (0x80) | TCCR1A | COM1A1 | COM1A0 | COM1B1 | COM1B0 | – | – | WGM11 | WGM10 | 113 |
| (0x7F) | DIDR1 | – | – | – | – | – | – | AIN1D | AIN0D | 188 |
| (0x7E) | DIDR0 | ADC7D | ADC6D | ADC5D | ADC4D | ADC3D | ADC2D | ADC1D | ADC0D | 205 |



| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Page |
|-------------|----------|--|---------|---------|---------|---------|---------|----------|--------|----------|
| (0x7D) | Reserved | – | – | – | – | – | – | – | – | |
| (0x7C) | ADMUX | REFS1 | REFS0 | ADLAR | MUX4 | MUX3 | MUX2 | MUX1 | MUX0 | 201 |
| (0x7B) | ADCSRB | – | ACME | – | – | – | ADTS2 | ADTS1 | ADTS0 | 186, 205 |
| (0x7A) | ADCSRA | ADEN | ADSC | ADATE | ADIF | ADIE | ADPS2 | ADPS1 | ADPS0 | 203 |
| (0x79) | ADCH | ADC Data Register High byte | | | | | | | | 204 |
| (0x78) | ADCL | ADC Data Register Low byte | | | | | | | | 204 |
| (0x77) | Reserved | – | – | – | – | – | – | – | – | |
| (0x76) | Reserved | – | – | – | – | – | – | – | – | |
| (0x75) | Reserved | – | – | – | – | – | – | – | – | |
| (0x74) | Reserved | – | – | – | – | – | – | – | – | |
| (0x73) | Reserved | – | – | – | – | – | – | – | – | |
| (0x72) | Reserved | – | – | – | – | – | – | – | – | |
| (0x71) | Reserved | – | – | – | – | – | – | – | – | |
| (0x70) | TIMSK2 | – | – | – | – | – | – | OCIE2A | TOIE2 | 136 |
| (0x6F) | TIMSK1 | – | – | ICIE1 | – | – | OCIE1B | OCIE1A | TOIE1 | 118 |
| (0x6E) | TIMSK0 | – | – | – | – | – | – | OCIE0A | TOIE0 | 88 |
| (0x6D) | Reserved | – | – | – | – | – | – | – | – | |
| (0x6C) | PCMSK1 | PCINT15 | PCINT14 | PCINT13 | PCINT12 | PCINT11 | PCINT10 | PCINT9 | PCINT8 | 54 |
| (0x6B) | PCMSK0 | PCINT7 | PCINT6 | PCINT5 | PCINT4 | PCINT3 | PCINT2 | PCINT1 | PCINT0 | 54 |
| (0x6A) | Reserved | – | – | – | – | – | – | – | – | |
| (0x69) | EICRA | – | – | – | – | – | – | ISC01 | ISC00 | 52 |
| (0x68) | Reserved | – | – | – | – | – | – | – | – | |
| (0x67) | Reserved | – | – | – | – | – | – | – | – | |
| (0x66) | OSCCAL | Oscillator Calibration Register | | | | | | | | 28 |
| (0x65) | Reserved | – | – | – | – | – | – | – | – | |
| (0x64) | PRR | – | – | – | – | PRTIM1 | PRSPI | PRUSART0 | PRADC | 34 |
| (0x63) | Reserved | – | – | – | – | – | – | – | – | |
| (0x62) | Reserved | – | – | – | – | – | – | – | – | |
| (0x61) | CLKPR | CLKPCE | – | – | – | CLKPS3 | CLKPS2 | CLKPS1 | CLKPS0 | 29 |
| (0x60) | WDTCSR | – | – | – | WDCE | WDE | WDP2 | WDP1 | WDP0 | 43 |
| 0x3F (0x5F) | SREG | I | T | H | S | V | N | Z | C | 9 |
| 0x3E (0x5E) | SPH | – | – | – | – | – | SP10 | SP9 | SP8 | 11 |
| 0x3D (0x5D) | SPL | SP7 | SP6 | SP5 | SP4 | SP3 | SP2 | SP1 | SP0 | 11 |
| 0x3C (0x5C) | Reserved | – | – | – | – | – | – | – | – | |
| 0x3B (0x5B) | Reserved | – | – | – | – | – | – | – | – | |
| 0x3A (0x5A) | Reserved | – | – | – | – | – | – | – | – | |
| 0x39 (0x59) | Reserved | – | – | – | – | – | – | – | – | |
| 0x38 (0x58) | Reserved | – | – | – | – | – | – | – | – | |
| 0x37 (0x57) | SPMCSR | SPMIE | RWWWSB | – | RWWWSRE | BLBSET | PGWRT | PGERS | SPMEN | 237 |
| 0x36 (0x56) | Reserved | – | – | – | – | – | – | – | – | |
| 0x35 (0x55) | MCUCR | JTD | – | – | PUD | – | – | IVSEL | IVCE | 215 |
| 0x34 (0x54) | MCUSR | – | – | – | JTRF | WDRF | BORF | EXTRF | PORF | 216 |
| 0x33 (0x53) | SMCR | – | – | – | – | SM2 | SM1 | SM0 | SE | 32 |
| 0x32 (0x52) | Reserved | – | – | – | – | – | – | – | – | |
| 0x31 (0x51) | OCDR | IDRD/ OCD | OCDR6 | OCDR5 | OCDR4 | OCDR3 | OCDR2 | OCDR1 | OCDR0 | 211 |
| 0x30 (0x50) | ACSR | ACD | ACBG | ACO | ACI | ACIE | ACIC | ACIS1 | ACIS0 | 186 |
| 0x2F (0x4F) | Reserved | – | – | – | – | – | – | – | – | |
| 0x2E (0x4E) | SPDR | SPI Data Register | | | | | | | | 146 |
| 0x2D (0x4D) | SPSR | SPIF | WCOL | – | – | – | – | – | SPI2X | 146 |
| 0x2C (0x4C) | SPCR | SPIE | SPE | DORD | MSTR | CPOL | CPHA | SPR1 | SPR0 | 144 |
| 0x2B (0x4B) | GPOR2 | General Purpose I/O Register 2 | | | | | | | | 22 |
| 0x2A (0x4A) | GPOR1 | General Purpose I/O Register 1 | | | | | | | | 22 |
| 0x29 (0x49) | Reserved | – | – | – | – | – | – | – | – | |
| 0x28 (0x48) | Reserved | – | – | – | – | – | – | – | – | |
| 0x27 (0x47) | OCR0A | Timer/Counter0 Output Compare Register A | | | | | | | | 88 |
| 0x26 (0x46) | TCNT0 | Timer/Counter0 (8 Bit) | | | | | | | | 87 |
| 0x25 (0x45) | Reserved | – | – | – | – | – | – | – | – | |
| 0x24 (0x44) | TCCR0A | FOC0A | WGM00 | COM0A1 | COM0A0 | WGM01 | CS02 | CS01 | CS00 | 85 |
| 0x23 (0x43) | GTCCR | TSM | – | – | – | – | – | PSR2 | PSR10 | 90 |
| 0x22 (0x42) | EEARH | – | – | – | – | – | – | – | EEAR8 | 18 |
| 0x21 (0x41) | EEARL | EEPROM Address Register Low Byte | | | | | | | | 18 |
| 0x20 (0x40) | EEDR | EEPROM Data Register | | | | | | | | 18 |
| 0x1F (0x3F) | EECR | – | – | – | – | EERIE | EEMWE | EERE | EERE | 18 |
| 0x1E (0x3E) | GPOR0 | General Purpose I/O Register 0 | | | | | | | | 22 |
| 0x1D (0x3D) | EIMSK | PCIE1 | PCIE0 | – | – | – | – | – | INT0 | 53 |
| 0x1C (0x3C) | EIFR | PCIF1 | PCIF0 | – | – | – | – | – | INTF0 | 53 |

| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Page |
|-------------|----------|--------|--------|--------|--------|--------|--------|--------|--------|------|
| 0x1B (0x3B) | Reserved | – | – | – | – | – | – | – | – | |
| 0x1A (0x3A) | Reserved | – | – | – | – | – | – | – | – | |
| 0x19 (0x39) | Reserved | – | – | – | – | – | – | – | – | |
| 0x18 (0x38) | Reserved | – | – | – | – | – | – | – | – | |
| 0x17 (0x37) | TIFR2 | – | – | – | – | – | – | OCF2A | TOV2 | 137 |
| 0x16 (0x36) | TIFR1 | – | – | ICF1 | – | – | OCF1B | OCF1A | TOV1 | 119 |
| 0x15 (0x35) | TIFR0 | – | – | – | – | – | – | OCF0A | TOV0 | 88 |
| 0x14 (0x34) | PORTG | – | – | – | PORTG4 | PORTG3 | PORTG2 | PORTG1 | PORTG0 | 74 |
| 0x13 (0x33) | DDRG | – | – | – | DDG4 | DDG3 | DDG2 | DDG1 | DDG0 | 74 |
| 0x12 (0x32) | PING | – | – | – | PING4 | PING3 | PING2 | PING1 | PING0 | 74 |
| 0x11 (0x31) | PORTF | PORTF7 | PORTF6 | PORTF5 | PORTF4 | PORTF3 | PORTF2 | PORTF1 | PORTF0 | 73 |
| 0x10 (0x30) | DDRF | DDF7 | DDF6 | DDF5 | DDF4 | DDF3 | DDF2 | DDF1 | DDF0 | 73 |
| 0x0F (0x2F) | PINF | PINF7 | PINF6 | PINF5 | PINF4 | PINF3 | PINF2 | PINF1 | PINF0 | 74 |
| 0x0E (0x2E) | PORTE | PORTE7 | PORTE6 | PORTE5 | PORTE4 | PORTE3 | PORTE2 | PORTE1 | PORTE0 | 73 |
| 0x0D (0x2D) | DDRE | DDE7 | DDE6 | DDE5 | DDE4 | DDE3 | DDE2 | DDE1 | DDE0 | 73 |
| 0x0C (0x2C) | PINE | PINE7 | PINE6 | PINE5 | PINE4 | PINE3 | PINE2 | PINE1 | PINE0 | 73 |
| 0x0B (0x2B) | PORTD | PORTD7 | PORTD6 | PORTD5 | PORTD4 | PORTD3 | PORTD2 | PORTD1 | PORTD0 | 73 |
| 0x0A (0x2A) | DDRD | DDD7 | DDD6 | DDD5 | DDD4 | DDD3 | DDD2 | DDD1 | DDD0 | 73 |
| 0x09 (0x29) | PIND | PIND7 | PIND6 | PIND5 | PIND4 | PIND3 | PIND2 | PIND1 | PIND0 | 73 |
| 0x08 (0x28) | PORTC | PORTC7 | PORTC6 | PORTC5 | PORTC4 | PORTC3 | PORTC2 | PORTC1 | PORTC0 | 72 |
| 0x07 (0x27) | DDRC | DDC7 | DDC6 | DDC5 | DDC4 | DDC3 | DDC2 | DDC1 | DDC0 | 72 |
| 0x06 (0x26) | PINC | PINC7 | PINC6 | PINC5 | PINC4 | PINC3 | PINC2 | PINC1 | PINC0 | 73 |
| 0x05 (0x25) | PORTB | PORTB7 | PORTB6 | PORTB5 | PORTB4 | PORTB3 | PORTB2 | PORTB1 | PORTB0 | 72 |
| 0x04 (0x24) | DDRB | DDB7 | DDB6 | DDB5 | DDB4 | DDB3 | DDB2 | DDB1 | DDB0 | 72 |
| 0x03 (0x23) | PINB | PINB7 | PINB6 | PINB5 | PINB4 | PINB3 | PINB2 | PINB1 | PINB0 | 72 |
| 0x02 (0x22) | PORTA | PORTA7 | PORTA6 | PORTA5 | PORTA4 | PORTA3 | PORTA2 | PORTA1 | PORTA0 | 72 |
| 0x01 (0x21) | DDRA | DDA7 | DDA6 | DDA5 | DDA4 | DDA3 | DDA2 | DDA1 | DDA0 | 72 |
| 0x00 (0x20) | PINA | PINA7 | PINA6 | PINA5 | PINA4 | PINA3 | PINA2 | PINA1 | PINA0 | 72 |

- Note:
1. For compatibility with future devices, reserved bits should be written to zero if accessed. Reserved I/O memory addresses should never be written.
 2. I/O Registers within the address range 0x00 - 0x1F are directly bit-accessible using the SBI and CBI instructions. In these registers, the value of single bits can be checked by using the SBIS and SBIC instructions.
 3. Some of the Status Flags are cleared by writing a logical one to them. Note that, unlike most other AVRs, the CBI and SBI instructions will only operate on the specified bit, and can therefore be used on registers containing such Status Flags. The CBI and SBI instructions work with registers 0x00 to 0x1F only.
 4. When using the I/O specific commands IN and OUT, the I/O addresses 0x00 - 0x3F must be used. When addressing I/O Registers as data space using LD and ST instructions, 0x20 must be added to these addresses. The ATmega165 is a complex microcontroller with more peripheral units than can be supported within the 64 location reserved in Opcode for the IN and OUT instructions. For the Extended I/O space from 0x60 - 0xFF in SRAM, only the ST/STS/STD and LD/LDS/LDD instructions can be used.

Instruction Set Summary

| Mnemonics | Operands | Description | Operation | Flags | #Clocks |
|--|----------|--|---|---------------|---------|
| ARITHMETIC AND LOGIC INSTRUCTIONS | | | | | |
| ADD | Rd, Rr | Add two Registers | $Rd \leftarrow Rd + Rr$ | Z,C,N,V,H | 1 |
| ADC | Rd, Rr | Add with Carry two Registers | $Rd \leftarrow Rd + Rr + C$ | Z,C,N,V,H | 1 |
| ADIW | Rd,K | Add Immediate to Word | $Rdh:Rdl \leftarrow Rdh:Rdl + K$ | Z,C,N,V,S | 2 |
| SUB | Rd, Rr | Subtract two Registers | $Rd \leftarrow Rd - Rr$ | Z,C,N,V,H | 1 |
| SUBI | Rd, K | Subtract Constant from Register | $Rd \leftarrow Rd - K$ | Z,C,N,V,H | 1 |
| SBC | Rd, Rr | Subtract with Carry two Registers | $Rd \leftarrow Rd - Rr - C$ | Z,C,N,V,H | 1 |
| SBCI | Rd, K | Subtract with Carry Constant from Reg. | $Rd \leftarrow Rd - K - C$ | Z,C,N,V,H | 1 |
| SBIW | Rd,K | Subtract Immediate from Word | $Rdh:Rdl \leftarrow Rdh:Rdl - K$ | Z,C,N,V,S | 2 |
| AND | Rd, Rr | Logical AND Registers | $Rd \leftarrow Rd \bullet Rr$ | Z,N,V | 1 |
| ANDI | Rd, K | Logical AND Register and Constant | $Rd \leftarrow Rd \bullet K$ | Z,N,V | 1 |
| OR | Rd, Rr | Logical OR Registers | $Rd \leftarrow Rd \vee Rr$ | Z,N,V | 1 |
| ORI | Rd, K | Logical OR Register and Constant | $Rd \leftarrow Rd \vee K$ | Z,N,V | 1 |
| EOR | Rd, Rr | Exclusive OR Registers | $Rd \leftarrow Rd \oplus Rr$ | Z,N,V | 1 |
| COM | Rd | One's Complement | $Rd \leftarrow 0xFF - Rd$ | Z,C,N,V | 1 |
| NEG | Rd | Two's Complement | $Rd \leftarrow 0x00 - Rd$ | Z,C,N,V,H | 1 |
| SBR | Rd,K | Set Bit(s) in Register | $Rd \leftarrow Rd \vee K$ | Z,N,V | 1 |
| CBR | Rd,K | Clear Bit(s) in Register | $Rd \leftarrow Rd \bullet (0xFF - K)$ | Z,N,V | 1 |
| INC | Rd | Increment | $Rd \leftarrow Rd + 1$ | Z,N,V | 1 |
| DEC | Rd | Decrement | $Rd \leftarrow Rd - 1$ | Z,N,V | 1 |
| TST | Rd | Test for Zero or Minus | $Rd \leftarrow Rd \bullet Rd$ | Z,N,V | 1 |
| CLR | Rd | Clear Register | $Rd \leftarrow Rd \oplus Rd$ | Z,N,V | 1 |
| SER | Rd | Set Register | $Rd \leftarrow 0xFF$ | None | 1 |
| MUL | Rd, Rr | Multiply Unsigned | $R1:R0 \leftarrow Rd \times Rr$ | Z,C | 2 |
| MULS | Rd, Rr | Multiply Signed | $R1:R0 \leftarrow Rd \times Rr$ | Z,C | 2 |
| MULSU | Rd, Rr | Multiply Signed with Unsigned | $R1:R0 \leftarrow Rd \times Rr$ | Z,C | 2 |
| FMUL | Rd, Rr | Fractional Multiply Unsigned | $R1:R0 \leftarrow (Rd \times Rr) \lll 1$ | Z,C | 2 |
| FMULS | Rd, Rr | Fractional Multiply Signed | $R1:R0 \leftarrow (Rd \times Rr) \lll 1$ | Z,C | 2 |
| FMULSU | Rd, Rr | Fractional Multiply Signed with Unsigned | $R1:R0 \leftarrow (Rd \times Rr) \lll 1$ | Z,C | 2 |
| BRANCH INSTRUCTIONS | | | | | |
| RJMP | k | Relative Jump | $PC \leftarrow PC + k + 1$ | None | 2 |
| IJMP | | Indirect Jump to (Z) | $PC \leftarrow Z$ | None | 2 |
| JMP | k | Direct Jump | $PC \leftarrow k$ | None | 3 |
| RCALL | k | Relative Subroutine Call | $PC \leftarrow PC + k + 1$ | None | 3 |
| ICALL | | Indirect Call to (Z) | $PC \leftarrow Z$ | None | 3 |
| CALL | k | Direct Subroutine Call | $PC \leftarrow k$ | None | 4 |
| RET | | Subroutine Return | $PC \leftarrow STACK$ | None | 4 |
| RETI | | Interrupt Return | $PC \leftarrow STACK$ | I | 4 |
| CPSE | Rd,Rr | Compare, Skip if Equal | if $(Rd = Rr)$ $PC \leftarrow PC + 2$ or 3 | None | 1/2/3 |
| CP | Rd,Rr | Compare | $Rd - Rr$ | Z, N, V, C, H | 1 |
| CPC | Rd,Rr | Compare with Carry | $Rd - Rr - C$ | Z, N, V, C, H | 1 |
| CPI | Rd,K | Compare Register with Immediate | $Rd - K$ | Z, N, V, C, H | 1 |
| SBRC | Rr, b | Skip if Bit in Register Cleared | if $(Rr(b)=0)$ $PC \leftarrow PC + 2$ or 3 | None | 1/2/3 |
| SBRS | Rr, b | Skip if Bit in Register is Set | if $(Rr(b)=1)$ $PC \leftarrow PC + 2$ or 3 | None | 1/2/3 |
| SBIC | P, b | Skip if Bit in I/O Register Cleared | if $(P(b)=0)$ $PC \leftarrow PC + 2$ or 3 | None | 1/2/3 |
| SBIS | P, b | Skip if Bit in I/O Register is Set | if $(P(b)=1)$ $PC \leftarrow PC + 2$ or 3 | None | 1/2/3 |
| BRBS | s, k | Branch if Status Flag Set | if $(SREG(s) = 1)$ then $PC \leftarrow PC + k + 1$ | None | 1/2 |
| BRBC | s, k | Branch if Status Flag Cleared | if $(SREG(s) = 0)$ then $PC \leftarrow PC + k + 1$ | None | 1/2 |
| BREQ | k | Branch if Equal | if $(Z = 1)$ then $PC \leftarrow PC + k + 1$ | None | 1/2 |
| BRNE | k | Branch if Not Equal | if $(Z = 0)$ then $PC \leftarrow PC + k + 1$ | None | 1/2 |
| BRCS | k | Branch if Carry Set | if $(C = 1)$ then $PC \leftarrow PC + k + 1$ | None | 1/2 |
| BRCC | k | Branch if Carry Cleared | if $(C = 0)$ then $PC \leftarrow PC + k + 1$ | None | 1/2 |
| BRSH | k | Branch if Same or Higher | if $(C = 0)$ then $PC \leftarrow PC + k + 1$ | None | 1/2 |
| BRLO | k | Branch if Lower | if $(C = 1)$ then $PC \leftarrow PC + k + 1$ | None | 1/2 |
| BRMI | k | Branch if Minus | if $(N = 1)$ then $PC \leftarrow PC + k + 1$ | None | 1/2 |
| BRPL | k | Branch if Plus | if $(N = 0)$ then $PC \leftarrow PC + k + 1$ | None | 1/2 |
| BRGE | k | Branch if Greater or Equal, Signed | if $(N \oplus V = 0)$ then $PC \leftarrow PC + k + 1$ | None | 1/2 |
| BRLT | k | Branch if Less Than Zero, Signed | if $(N \oplus V = 1)$ then $PC \leftarrow PC + k + 1$ | None | 1/2 |
| BRHS | k | Branch if Half Carry Flag Set | if $(H = 1)$ then $PC \leftarrow PC + k + 1$ | None | 1/2 |
| BRHC | k | Branch if Half Carry Flag Cleared | if $(H = 0)$ then $PC \leftarrow PC + k + 1$ | None | 1/2 |
| BRTS | k | Branch if T Flag Set | if $(T = 1)$ then $PC \leftarrow PC + k + 1$ | None | 1/2 |
| BRTC | k | Branch if T Flag Cleared | if $(T = 0)$ then $PC \leftarrow PC + k + 1$ | None | 1/2 |
| BRVS | k | Branch if Overflow Flag is Set | if $(V = 1)$ then $PC \leftarrow PC + k + 1$ | None | 1/2 |
| BRVC | k | Branch if Overflow Flag is Cleared | if $(V = 0)$ then $PC \leftarrow PC + k + 1$ | None | 1/2 |

| Mnemonics | Operands | Description | Operation | Flags | #Clocks |
|--------------------------------------|----------|----------------------------------|--|---------|---------|
| BRIE | k | Branch if Interrupt Enabled | if (I = 1) then PC ← PC + k + 1 | None | 1/2 |
| BRID | k | Branch if Interrupt Disabled | if (I = 0) then PC ← PC + k + 1 | None | 1/2 |
| 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 |
| CLH | | Clear Half Carry Flag in SREG | H ← 0 | H | 1 |
| 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 |

| Mnemonics | Operands | Description | Operation | Flags | #Clocks |
|---------------------------------|----------|-------------------------|--|-------|---------|
| POP | Rd | Pop Register from Stack | Rd ← STACK | None | 2 |
| 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) ⁽³⁾ | Power Supply | Ordering Code | Package ⁽¹⁾ | Operation Range |
|----------------------------|--------------|--|----------------------------|-------------------------------|
| 8 | 1.8 - 5.5V | ATmega165V-8AI ATmega165V-8AU ⁽²⁾ ATmega165V-8MI ATmega165V-8MU ⁽²⁾ | 64A 64A 64M1 64M1 | Industrial (-40°C to 85°C) |
| 16 | 2.7 - 5.5V | ATmega165-16AI ATmega165-16AU ⁽²⁾ ATmega165-16MI ATmega165-16MU ⁽²⁾ | 64A 64A 64M1 64M1 | Industrial (-40°C to 85°C) |

- Notes:
1. This device can also be supplied in wafer form. Please contact your local Atmel sales office for detailed ordering information and minimum quantities.
 2. Pb-free packaging alternative, complies to the European Directive for Restriction of Hazardous Substances (RoHS directive). Also Halide free and fully Green.
 3. For Speed Vs. V_{CC} See Figure 128 on page 282 and Figure 129 on page 282.

| Package Type | |
|--------------|---|
| 64A | 64-Lead, Thin (1.0 mm) Plastic Gull Wing Quad Flat Package (TQFP) |
| 64M1 | 64-pad, 9 x 9 x 1.0 mm body, lead pitch 0.50 mm, Quad Flat No-Lead/Micro Lead Frame Package (QFN/MLF) |

Packaging Information

64A

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 | 15.75 | 16.00 | 16.25 | |
| D1 | 13.90 | 14.00 | 14.10 | Note 2 |
| E | 15.75 | 16.00 | 16.25 | |
| E1 | 13.90 | 14.00 | 14.10 | Note 2 |
| B | 0.30 | - | 0.45 | |
| C | 0.09 | - | 0.20 | |
| L | 0.45 | - | 0.75 | |
| e | 0.80 TYP | | | |

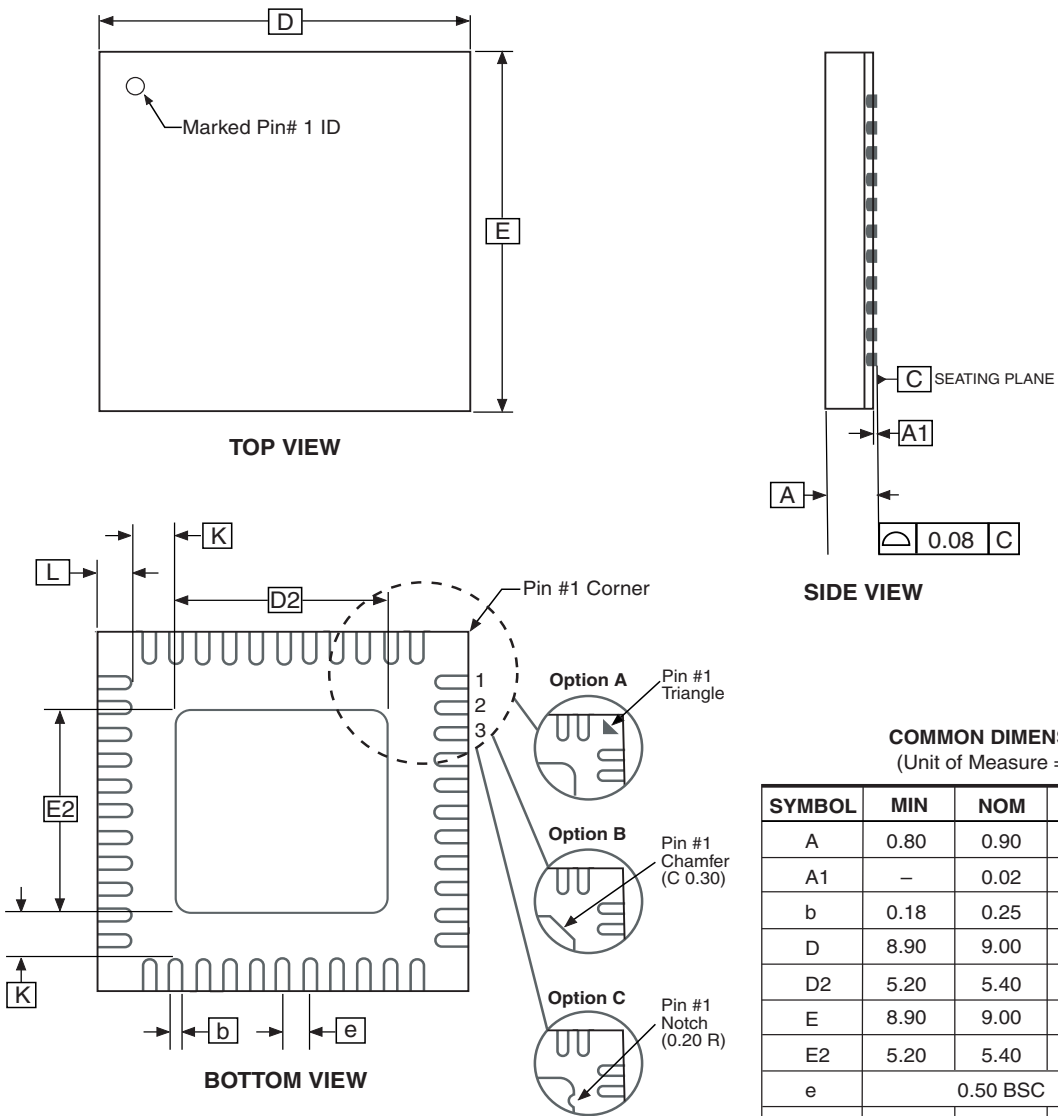
Notes:

1. This package conforms to JEDEC reference MS-026, Variation AEB.
2. Dimensions D1 and E1 do not include mold protrusion. Allowable protrusion is 0.25 mm per side. Dimensions D1 and E1 are maximum plastic body size dimensions including mold mismatch.
3. Lead coplanarity is 0.10 mm maximum.

10/5/2001

| | | | | |
|--|--|--|---------------------------|------------------|
| | 2325 Orchard Parkway San Jose, CA 95131 | TITLE 64A, 64-lead, 14 x 14 mm Body Size, 1.0 mm Body Thickness, 0.8 mm Lead Pitch, Thin Profile Plastic Quad Flat Package (TQFP) | DRAWING NO. 64A | REV. B |
| | | | | |

64M1



COMMON DIMENSIONS
(Unit of Measure = mm)

| SYMBOL | MIN | NOM | MAX | NOTE |
|--------|----------|------|------|------|
| A | 0.80 | 0.90 | 1.00 | |
| A1 | — | 0.02 | 0.05 | |
| b | 0.18 | 0.25 | 0.30 | |
| D | 8.90 | 9.00 | 9.10 | |
| D2 | 5.20 | 5.40 | 5.60 | |
| E | 8.90 | 9.00 | 9.10 | |
| E2 | 5.20 | 5.40 | 5.60 | |
| e | 0.50 BSC | | | |
| L | 0.35 | 0.40 | 0.45 | |
| K | 1.25 | 1.40 | 1.55 | |

Note: 1. JEDEC Standard MO-220, (SAW Singulation) Fig. 1, VMMD.
2. Dimension and tolerance conform to ASMEY14.5M-1994.

5/25/06



2325 Orchard Parkway
San Jose, CA 95131

TITLE

64M1, 64-pad, 9 x 9 x 1.0 mm Body, Lead Pitch 0.50 mm,
5.40 mm Exposed Pad, Micro Lead Frame Package (MLF)

DRAWING NO.

64M1

REV.

G



Errata

ATmega165 Rev A

- Interrupts may be lost when writing the timer registers in the asynchronous timer

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

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

Problem Fix/Workaround

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

Datasheet Revision History

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

Changes from Rev. 2573F-08/06 to Rev. 2573G-07/09

1. Updated “Errata” on page 329.
2. Updated the last page with Atmel’s new addresses.

Changes from Rev. 2573E-07/06 to Rev. 2573F-08/06

1. Updated “Device Identification Register” on page 213.
2. Updated “Signature Bytes” on page 249.
3. Added “Device and JTAG ID” on page 249.

Changes from Rev. 2573D-03/06 to Rev. 2573E-07/06

1. Updated “Fast PWM Mode” on page 105.
2. Updated Features in “USI – Universal Serial Interface” on page 175.
3. Updated Table 42 on page 86, Table 44 on page 86, Table 49 on page 113, Table 50 on page 114, Table 51 on page 115, Table 54 on page 131 and Table 56 on page 132.
4. Added “Errata” on page 329.

Changes from Rev. 2573C-03/06 to Rev. 2573D-03/06

1. Updated number of General Purpose I/O pins from 53 to 54.
2. Updated “Serial Peripheral Interface – SPI” on page 139.

Changes from Rev. 2573B-03/05 to Rev. 2573C-02/06

1. Added Not recommended in new designs.
2. Updated “BODLEVEL Fuse Coding(1)” on page 40.

Changes from Rev. 2573A-06/04 to Rev. 2573B-03/05

1. MLF-package alternative changed to “Quad Flat No-Lead/Micro Lead Frame Package QFN/MLF”.
2. Updated Table 16 on page 38, Table 49 on page 113, Table 50 on page 114, Table 86 on page 212 and Table 115 on page 263.
3. Added “Pin Change Interrupt Timing” on page 51.
4. Updated C Code Example in “USART Initialization” on page 152
5. Moved “Table 106 on page 248” and “Table 107 on page 248” to “Page Size” on page 248.
6. Updated “Register Summary” on page 7
7. Updated Figure 115 on page 255.
8. Updated “Ordering Information” on page 14



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

Technical Support
avr@atmel.com

Sales Contact
www.atmel.com/contacts

Literature Requests
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.

© 2009 Atmel Corporation. All rights reserved. Atmel®, Atmel logo and combinations thereof, 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 ATMEGA165-16AI 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