## Features

- High performance, low power Atmel® AVR® 8-bit microcontroller
- Advanced RISC architecture
  - 131 powerful instructions – most single clock cycle execution
  - 32 × 8 general purpose working registers
  - Fully static operation
  - Up to 20 MIPS throughput at 20MHz
  - On-chip 2-cycle multiplier
- High endurance non-volatile memory segments
  - 4/8/16 Kbytes of in-system self-programmable flash program memory
  - 256/512/512 bytes EEPROM
  - 512/1K/1Kbytes internal SRAM
  - Write/erase cyles: 10,000 flash/100,000 EEPROM
  - Data retention: 20 years at 85°C/100 years at 25°C(1)
  - Optional boot code section with independent lock bits
    - In-system programming by on-chip boot program
    - True read-while-write operation
  - Programming lock for software security
- QTouch® library support
  - Capacitive touch buttons, sliders and wheels
  - QTouch and QMatrix acquisition
  - Up to 64 sense channels
- 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
  - Six PWM channels
  - 8-channel 10-bit ADC in TQFP and QFN/MLF package
  - 6-channel 10-bit ADC in PDIP Package
  - Programmable serial USART
  - Master/slave SPI serial interface
  - Byte-oriented 2-wire serial interface (Philips $I^2C$ compatible)
  - Programmable watchdog timer with separate on-chip oscillator
  - On-chip analog comparator
  - Interrupt and wake-up on pin change
- Special microcontroller features
  - DebugWIRE on-chip debug system
  - 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
  - 23 programmable I/O lines
  - 28-pin PDIP, 32-lead TQFP, 28-pad QFN/MLF and 32-pad QFN/MLF
- Operating voltage:
  - 1.8V - 5.5V for Atmel ATmega48V/88V/168V
  - 2.7V - 5.5V for Atmel ATmega48/88/168
- Temperature range:
  - -40°C to 85°C
- Speed grade:
  - ATmega48V/88V/168V: 0 - 4MHz @ 1.8V - 5.5V, 0 - 10MHz @ 2.7V - 5.5V
  - ATmega48/88/168: 0 - 10MHz @ 2.7V - 5.5V, 0 - 20MHz @ 4.5V - 5.5V
- Low power consumption
  - Active mode:
    - 250µA at 1MHz, 1.8V
    - 15µA at 32kHz, 1.8V (including oscillator)
  - Power-down mode:
    - 0.1µA at 1.8V

Note: 1. See “Data retention” on page 8 for details.

# 8-bit Atmel Microcontroller with 4/8/16K Bytes In-System Programmable Flash

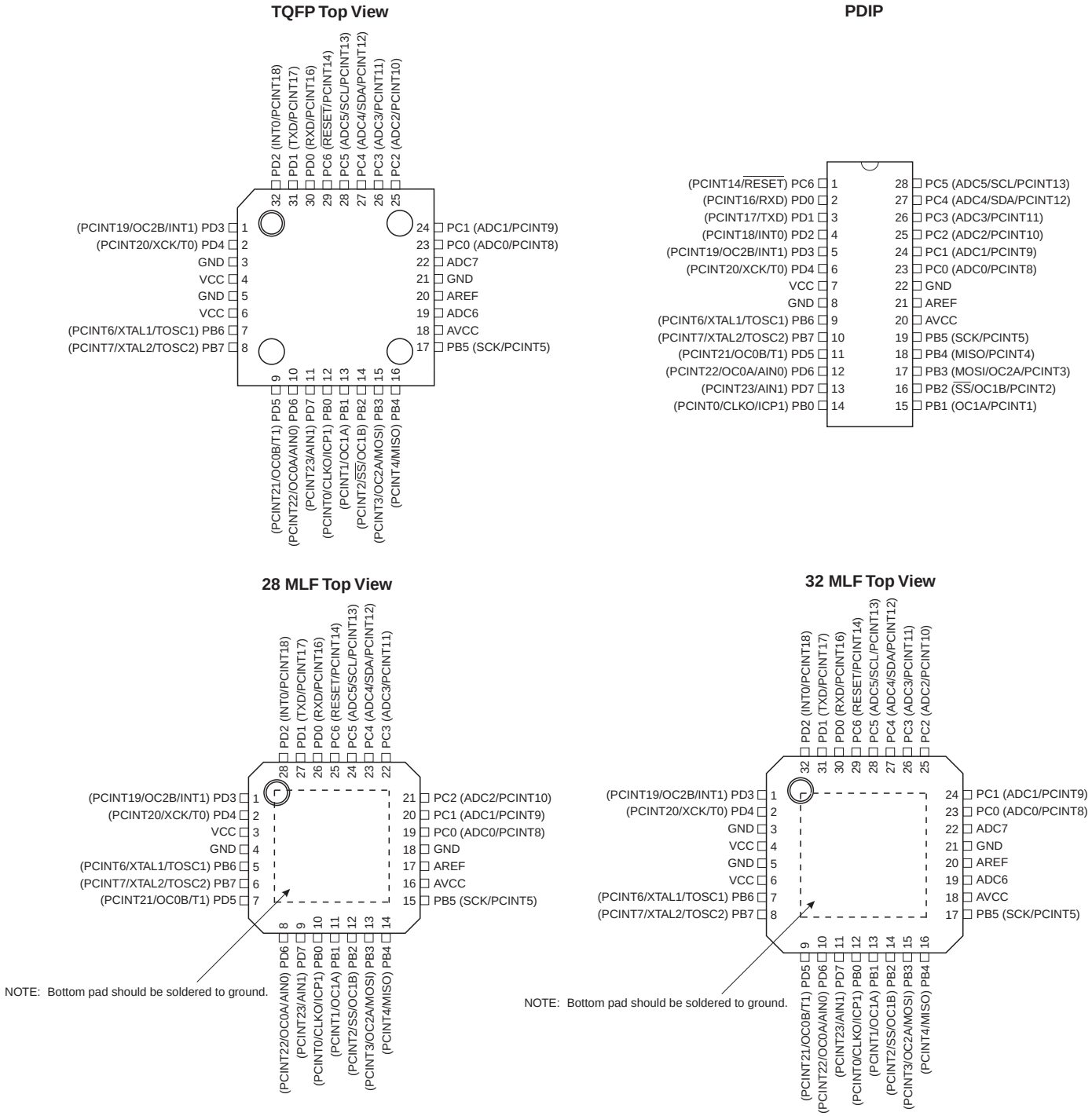
## ATmega48/V
## ATmega88/V
## ATmega168/V

## Summary

Rev. 2545US–AVR–11/2015

# 1. Pin configurations

**Figure 1-1. Pinout Atmel ATmega48/88/168.**



## 1.1 Pin descriptions

### 1.1.1 VCC

Digital supply voltage.

### 1.1.2 GND

Ground.

### 1.1.3 Port B (PB7:0) XTAL1/XTAL2/TOSC1/TOSC2

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.

Depending on the clock selection fuse settings, PB6 can be used as input to the inverting Oscillator amplifier and input to the internal clock operating circuit.

Depending on the clock selection fuse settings, PB7 can be used as output from the inverting Oscillator amplifier.

If the Internal Calibrated RC Oscillator is used as chip clock source, PB7..6 is used as TOSC2..1 input for the Asynchronous Timer/Counter2 if the AS2 bit in ASSR is set.

The various special features of Port B are elaborated in “Alternate functions of port B” on page 83 and “System clock and clock options” on page 27.

### 1.1.4 Port C (PC5:0)

Port C is a 7-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The PC5..0 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.

### 1.1.5 PC6/$\overline{\text{RESET}}$

If the RSTDISBL Fuse is programmed, PC6 is used as an I/O pin. Note that the electrical characteristics of PC6 differ from those of the other pins of Port C.

If the RSTDISBL Fuse is unprogrammed, PC6 is used as a 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 29-3 on page 314. Shorter pulses are not guaranteed to generate a Reset.

The various special features of Port C are elaborated in “Alternate functions of port C” on page 86.

### 1.1.6 Port D (PD7:0)

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.

The various special features of Port D are elaborated in “Alternate functions of port D” on page 89.

### 1.1.7 $AV_{CC}$

$AV_{CC}$ is the supply voltage pin for the A/D Converter, PC3:0, and ADC7:6. 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. Note that PC6..4 use digital supply voltage, $V_{CC}$.

### 1.1.8 AREF

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

### 1.1.9 ADC7:6 (TQFP and QFN/MLF package only)

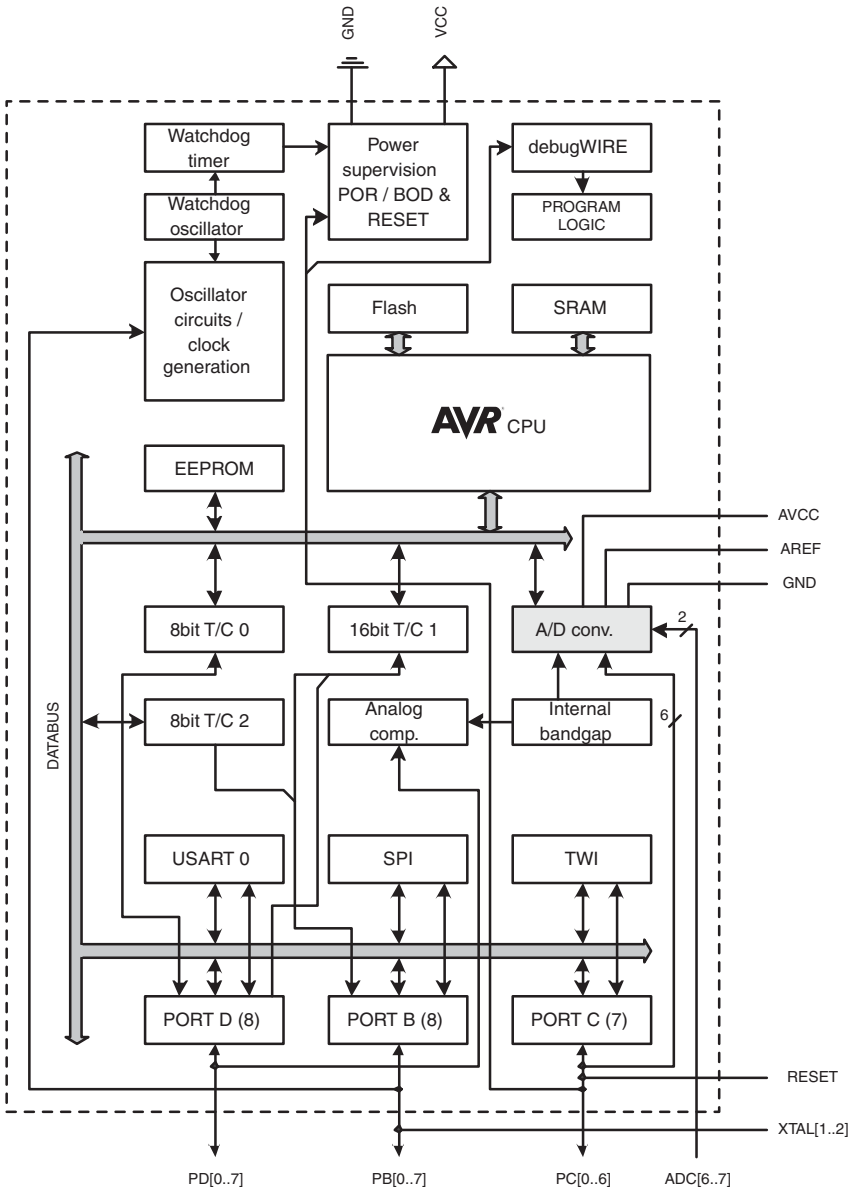
In the TQFP and QFN/MLF package, ADC7:6 serve as analog inputs to the A/D converter. These pins are powered from the analog supply and serve as 10-bit ADC channels.

# 2. Overview

The Atmel ATmega48/88/168 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 ATmega48/88/168 achieves throughputs approaching 1 MIPS per MHz allowing the system designer to optimize power consumption versus processing speed.

## 2.1 Block diagram

**Figure 2-1.** 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 Atmel ATmega48/88/168 provides the following features: 4K/8K/16K bytes of In-System Programmable Flash with Read-While-Write capabilities, 256/512/512 bytes EEPROM, 512/1K/1K bytes SRAM, 23 general purpose I/O lines, 32 general purpose working registers, three flexible Timer/Counters with compare modes, internal and external interrupts, a serial programmable USART, a byte-oriented 2-wire Serial Interface, an SPI serial port, a 6-channel 10-bit ADC (8 channels in TQFP and QFN/MLF packages), a programmable Watchdog Timer with internal Oscillator, and five software selectable power saving modes. The Idle mode stops the CPU while allowing the SRAM, Timer/Counters, USART, 2-wire Serial Interface, 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.

Atmel offers the QTouch Library for embedding capacitive touch buttons, sliders and wheels functionality into AVR microcontrollers. The patented charge-transfer signal acquisition offers robust sensing and includes fully debounced reporting of touch keys and includes Adjacent Key Suppression® (AKS®) technology for unambigiuous detection of key events. The easy-to-use QTouch Suite toolchain allows you to explore, develop and debug your own touch applications.

The device is manufactured using the Atmel 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 ATmega48/88/168 is a powerful microcontroller that provides a highly flexible and cost effective solution to many embedded control applications.

The ATmega48/88/168 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.

## 2.2 Comparison between Atmel ATmega48, Atmel ATmega88, and Atmel ATmega168

The ATmega48, ATmega88 and ATmega168 differ only in memory sizes, boot loader support, and interrupt vector sizes. Table 2-1 summarizes the different memory and interrupt vector sizes for the three devices.

**Table 2-1. Memory size summary.**

| Device | Flash | EEPROM | RAM | Interrupt vector size |
|---|---|---|---|---|
| ATmega48 | 4Kbytes | 256Bytes | 512Bytes | 1 instruction word/vector |
| ATmega88 | 8Kbytes | 512Bytes | 1Kbytes | 1 instruction word/vector |
| ATmega168 | 16Kbytes | 512Bytes | 1Kbytes | 2 instruction words/vector |

ATmega88 and ATmega168 support a real Read-While-Write Self-Programming mechanism. There is a separate Boot Loader Section, and the SPM instruction can only execute from there. In ATmega48, there is no Read-While-Write support and no separate Boot Loader Section. The SPM instruction can execute from the entire Flash.

## 3. Resources

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

## 4. Data retention

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

## 5. About code examples

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

For I/O Registers located in extended I/O map, “IN”, “OUT”, “SBIS”, “SBIC”, “CBI”, and “SBI” instructions must be replaced with instructions that allow access to extended I/O. Typically “LDS” and “STS” combined with “SBRS”, “SBRC”, “SBR”, and “CBR”.

## 6. Capacitive touch sensing

The Atmel QTouch Library provides a simple to use solution to realize touch sensitive interfaces on most Atmel AVR microcontrollers. The QTouch Library includes support for the QTouch and QMatrix acquisition methods.

Touch sensing can be added to any application by linking the appropriate Atmel QTouch Library for the AVR Microcontroller. This is done by using a simple set of APIs to define the touch channels and sensors, and then calling the touch sensing API’s to retrieve the channel information and determine the touch sensor states.

The QTouch Library is FREE and downloadable from the Atmel website at the following location: www.atmel.com/qtouchlibrary. For implementation details and other information, refer to the Atmel QTouch Library User Guide - also available for download from the Atmel website.

# 7. 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) | UDR0 | USART I/O data register | | | | | | | | 194 |
| (0xC5) | UBRR0H | | | | | USART baud rate register high | | | | 198 |
| (0xC4) | UBRR0L | USART baud rate register low | | | | | | | | 198 |
| (0xC3) | Reserved | – | – | – | – | – | – | – | – | |
| (0xC2) | UCSR0C | UMSEL01 | UMSEL00 | UPM01 | UPM00 | USBS0 | UCSZ01 /UDORD0 | UCSZ00 / UCPHA0 | UCPOL0 | 196/211 |
| (0xC1) | UCSR0B | RXCIE0 | TXCIE0 | UDRIE0 | RXEN0 | TXEN0 | UCSZ02 | RXB80 | TXB80 | 195 |
| (0xC0) | UCSR0A | RXC0 | TXC0 | UDRE0 | FE0 | DOR0 | UPE0 | U2X0 | MPCM0 | 194 |

| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Page |
|---|---|---|---|---|---|---|---|---|---|---|
| (0xBF) | Reserved | – | – | – | – | – | – | – | – | |
| (0xBE) | Reserved | – | – | – | – | – | – | – | – | |
| (0xBD) | TWAMR | TWAM6 | TWAM5 | TWAM4 | TWAM3 | TWAM2 | TWAM1 | TWAM0 | – | 244 |
| (0xBC) | TWCR | TWINT | TWEA | TWSTA | TWSTO | TWWC | TWEN | – | TWIE | 241 |
| (0xBB) | TWDR | 2-wire serial interface data register | | | | | | | | 243 |
| (0xBA) | TWAR | TWA6 | TWA5 | TWA4 | TWA3 | TWA2 | TWA1 | TWA0 | TWGCE | 244 |
| (0xB9) | TWSR | TWS7 | TWS6 | TWS5 | TWS4 | TWS3 | – | TWPS1 | TWPS0 | 243 |
| (0xB8) | TWBR | 2-wire serial interface bit rate register | | | | | | | | 241 |
| (0xB7) | Reserved | – | | – | – | – | – | – | – | |
| (0xB6) | ASSR | – | EXCLK | AS2 | TCN2UB | OCR2AUB | OCR2BUB | TCR2AUB | TCR2BUB | 163 |
| (0xB5) | Reserved | – | – | – | – | – | – | – | – | |
| (0xB4) | OCR2B | Timer/Counter2 output compare register B | | | | | | | | 162 |
| (0xB3) | OCR2A | Timer/Counter2 output compare register A | | | | | | | | 161 |
| (0xB2) | TCNT2 | Timer/Counter2 (8-bit) | | | | | | | | 161 |
| (0xB1) | TCCR2B | FOC2A | FOC2B | – | – | WGM22 | CS22 | CS21 | CS20 | 160 |
| (0xB0) | TCCR2A | COM2A1 | COM2A0 | COM2B1 | COM2B0 | – | – | WGM21 | WGM20 | 157 |
| (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 | | | | | | | | 138 |
| (0x8A) | OCR1BL | Timer/Counter1 - output compare register B low byte | | | | | | | | 138 |
| (0x89) | OCR1AH | Timer/Counter1 - output compare register A high byte | | | | | | | | 138 |
| (0x88) | OCR1AL | Timer/Counter1 - output compare register A low byte | | | | | | | | 138 |
| (0x87) | ICR1H | Timer/Counter1 - input capture register high byte | | | | | | | | 139 |
| (0x86) | ICR1L | Timer/Counter1 - input capture register low byte | | | | | | | | 139 |
| (0x85) | TCNT1H | Timer/Counter1 - counter register high byte | | | | | | | | 138 |
| (0x84) | TCNT1L | Timer/Counter1 - counter register low byte | | | | | | | | 138 |
| (0x83) | Reserved | – | – | – | – | – | – | – | – | |
| (0x82) | TCCR1C | FOC1A | FOC1B | – | – | – | – | – | – | 137 |
| (0x81) | TCCR1B | ICNC1 | ICES1 | – | WGM13 | WGM12 | CS12 | CS11 | CS10 | 136 |
| (0x80) | TCCR1A | COM1A1 | COM1A0 | COM1B1 | COM1B0 | – | – | WGM11 | WGM10 | 134 |
| (0x7F) | DIDR1 | – | – | – | – | – | – | AIN1D | AIN0D | 248 |
| (0x7E) | DIDR0 | – | – | ADC5D | ADC4D | ADC3D | ADC2D | ADC1D | ADC0D | 265 |

| 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 | – | MUX3 | MUX2 | MUX1 | MUX0 | 261 |
| (0x7B) | ADCSRB | – | ACME | – | – | – | ADTS2 | ADTS1 | ADTS0 | 264 |
| (0x7A) | ADCSRA | ADEN | ADSC | ADATE | ADIF | ADIE | ADPS2 | ADPS1 | ADPS0 | 262 |
| (0x79) | ADCH | ADC data register high byte | | | | | | | | 264 |
| (0x78) | ADCL | ADC data register low byte | | | | | | | | 264 |
| (0x77) | Reserved | – | – | – | – | – | – | – | – | |
| (0x76) | Reserved | – | – | – | – | – | – | – | – | |
| (0x75) | Reserved | – | – | – | – | – | – | – | – | |
| (0x74) | Reserved | – | – | – | – | – | – | – | – | |
| (0x73) | Reserved | – | – | – | – | – | – | – | – | |
| (0x72) | Reserved | – | – | – | – | – | – | – | – | |
| (0x71) | Reserved | – | – | – | – | – | – | – | – | |
| (0x70) | TIMSK2 | – | – | – | – | – | OCIE2B | OCIE2A | TOIE2 | 162 |
| (0x6F) | TIMSK1 | – | – | ICIE1 | – | – | OCIE1B | OCIE1A | TOIE1 | 139 |
| (0x6E) | TIMSK0 | – | – | – | – | – | OCIE0B | OCIE0A | TOIE0 | 111 |
| (0x6D) | PCMSK2 | PCINT23 | PCINT22 | PCINT21 | PCINT20 | PCINT19 | PCINT18 | PCINT17 | PCINT16 | 75 |
| (0x6C) | PCMSK1 | – | PCINT14 | PCINT13 | PCINT12 | PCINT11 | PCINT10 | PCINT9 | PCINT8 | 75 |
| (0x6B) | PCMSK0 | PCINT7 | PCINT6 | PCINT5 | PCINT4 | PCINT3 | PCINT2 | PCINT1 | PCINT0 | 75 |
| (0x6A) | Reserved | – | – | – | – | – | – | – | – | |
| (0x69) | EICRA | – | – | – | – | ISC11 | ISC10 | ISC01 | ISC00 | 71 |
| (0x68) | PCICR | – | – | – | – | – | PCIE2 | PCIE1 | PCIE0 | |
| (0x67) | Reserved | – | – | – | – | – | – | – | – | |
| (0x66) | OSCCAL | Oscillator calibration register | | | | | | | | 37 |
| (0x65) | Reserved | – | – | – | – | – | – | – | – | |
| (0x64) | PRR | PRTWI | PRTIM2 | PRTIM0 | – | PRTIM1 | PRSPI | PRUSART0 | PRADC | 41 |
| (0x63) | Reserved | – | – | – | – | – | – | – | – | |
| (0x62) | Reserved | – | – | – | – | – | – | – | – | |
| (0x61) | CLKPR | CLKPCE | – | – | – | CLKPS3 | CLKPS2 | CLKPS1 | CLKPS0 | 37 |
| (0x60) | WDTCSR | WDIF | WDIE | WDP3 | WDCE | WDE | WDP2 | WDP1 | WDP0 | 53 |
| 0x3F (0x5F) | SREG | I | T | H | S | V | N | Z | C | 11 |
| 0x3E (0x5E) | SPH | – | – | – | – | – | (SP10) [5.] | SP9 | SP8 | 13 |
| 0x3D (0x5D) | SPL | SP7 | SP6 | SP5 | SP4 | SP3 | SP2 | SP1 | SP0 | 13 |
| 0x3C (0x5C) | Reserved | – | – | – | – | – | – | – | – | |
| 0x3B (0x5B) | Reserved | – | – | – | – | – | – | – | – | |
| 0x3A (0x5A) | Reserved | – | – | – | – | – | – | – | – | |
| 0x39 (0x59) | Reserved | – | – | – | – | – | – | – | – | |
| 0x38 (0x58) | Reserved | – | – | – | – | – | – | – | – | |
| 0x37 (0x57) | SPMCSR | SPMIE | (RWWSB)[5.] | – | (RWWSRE)[5.] | BLBSET | PGWRT | PGERS | SELFPRGEN | 290 |
| 0x36 (0x56) | Reserved | – | – | – | – | – | – | – | – | |
| 0x35 (0x55) | MCUCR | – | – | – | PUD | – | – | IVSEL | IVCE | |
| 0x34 (0x54) | MCUSR | – | – | – | – | WDRF | BORF | EXTRF | PORF | |
| 0x33 (0x53) | SMCR | – | – | – | – | SM2 | SM1 | SM0 | SE | 39 |
| 0x32 (0x52) | Reserved | – | – | – | – | – | – | – | – | |
| 0x31 (0x51) | Reserved | – | – | – | – | – | – | – | – | |
| 0x30 (0x50) | ACSR | ACD | ACBG | ACO | ACI | ACIE | ACIC | ACIS1 | ACIS0 | 247 |
| 0x2F (0x4F) | Reserved | – | – | – | – | – | – | – | – | |
| 0x2E (0x4E) | SPDR | SPI data register | | | | | | | | 174 |
| 0x2D (0x4D) | SPSR | SPIF | WCOL | – | – | – | – | – | SPI2X | 173 |
| 0x2C (0x4C) | SPCR | SPIE | SPE | DORD | MSTR | CPOL | CPHA | SPR1 | SPR0 | 172 |
| 0x2B (0x4B) | GPIOR2 | General purpose I/O register 2 | | | | | | | | 26 |
| 0x2A (0x4A) | GPIOR1 | General purpose I/O register 1 | | | | | | | | 26 |
| 0x29 (0x49) | Reserved | – | – | – | – | – | – | – | – | |
| 0x28 (0x48) | OCR0B | Timer/Counter0 output compare register B | | | | | | | | |
| 0x27 (0x47) | OCR0A | Timer/Counter0 output compare register A | | | | | | | | |
| 0x26 (0x46) | TCNT0 | Timer/Counter0 (8-bit) | | | | | | | | |
| 0x25 (0x45) | TCCR0B | FOC0A | FOC0B | – | – | WGM02 | CS02 | CS01 | CS00 | |
| 0x24 (0x44) | TCCR0A | COM0A1 | COM0A0 | COM0B1 | COM0B0 | – | – | WGM01 | WGM00 | |
| 0x23 (0x43) | GTCCR | TSM | – | – | – | – | – | PSRASY | PSRSYNC | 143/164 |
| 0x22 (0x42) | EEARH | (EEPROM address register high byte) [5.] | | | | | | | | 22 |
| 0x21 (0x41) | EEARL | EEPROM address register low byte | | | | | | | | 22 |
| 0x20 (0x40) | EEDR | EEPROM data register | | | | | | | | 22 |
| 0x1F (0x3F) | EECR | – | – | EEPM1 | EEPM0 | EERIE | EEMPE | EEPE | EERE | 22 |
| 0x1E (0x3E) | GPIOR0 | General purpose I/O register 0 | | | | | | | | 26 |
| 0x1D (0x3D) | EIMSK | – | – | – | – | – | – | INT1 | INT0 | 73 |
| 0x1C (0x3C) | EIFR | – | – | – | – | – | – | INTF1 | INTF0 | 73 |

| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Page |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x1B (0x3B) | PCIFR | – | – | – | – | – | PCIF2 | PCIF1 | PCIF0 | |
| 0x1A (0x3A) | Reserved | – | – | – | – | – | – | – | – | |
| 0x19 (0x39) | Reserved | – | – | – | – | – | – | – | – | |
| 0x18 (0x38) | Reserved | – | – | – | – | – | – | – | – | |
| 0x17 (0x37) | TIFR2 | – | – | – | – | – | OCF2B | OCF2A | TOV2 | 162 |
| 0x16 (0x36) | TIFR1 | – | – | ICF1 | – | – | OCF1B | OCF1A | TOV1 | 140 |
| 0x15 (0x35) | TIFR0 | – | – | – | – | – | OCF0B | OCF0A | TOV0 | |
| 0x14 (0x34) | Reserved | – | – | – | – | – | – | – | – | |
| 0x13 (0x33) | Reserved | – | – | – | – | – | – | – | – | |
| 0x12 (0x32) | Reserved | – | – | – | – | – | – | – | – | |
| 0x11 (0x31) | Reserved | – | – | – | – | – | – | – | – | |
| 0x10 (0x30) | Reserved | – | – | – | – | – | – | – | – | |
| 0x0F (0x2F) | Reserved | – | – | – | – | – | – | – | – | |
| 0x0E (0x2E) | Reserved | – | – | – | – | – | – | – | – | |
| 0x0D (0x2D) | Reserved | – | – | – | – | – | – | – | – | |
| 0x0C (0x2C) | Reserved | – | – | – | – | – | – | – | – | |
| 0x0B (0x2B) | PORTD | PORTD7 | PORTD6 | PORTD5 | PORTD4 | PORTD3 | PORTD2 | PORTD1 | PORTD0 | 93 |
| 0x0A (0x2A) | DDRD | DDD7 | DDD6 | DDD5 | DDD4 | DDD3 | DDD2 | DDD1 | DDD0 | 93 |
| 0x09 (0x29) | PIND | PIND7 | PIND6 | PIND5 | PIND4 | PIND3 | PIND2 | PIND1 | PIND0 | 93 |
| 0x08 (0x28) | PORTC | – | PORTC6 | PORTC5 | PORTC4 | PORTC3 | PORTC2 | PORTC1 | PORTC0 | 92 |
| 0x07 (0x27) | DDRC | – | DDC6 | DDC5 | DDC4 | DDC3 | DDC2 | DDC1 | DDC0 | 92 |
| 0x06 (0x26) | PINC | – | PINC6 | PINC5 | PINC4 | PINC3 | PINC2 | PINC1 | PINC0 | 92 |
| 0x05 (0x25) | PORTB | PORTB7 | PORTB6 | PORTB5 | PORTB4 | PORTB3 | PORTB2 | PORTB1 | PORTB0 | 92 |
| 0x04 (0x24) | DDRB | DDB7 | DDB6 | DDB5 | DDB4 | DDB3 | DDB2 | DDB1 | DDB0 | 92 |
| 0x03 (0x23) | PINB | PINB7 | PINB6 | PINB5 | PINB4 | PINB3 | PINB2 | PINB1 | PINB0 | 92 |
| 0x02 (0x22) | Reserved | – | – | – | – | – | – | – | – | |
| 0x01 (0x21) | Reserved | – | – | – | – | – | – | – | – | |
| 0x0 (0x20) | Reserved | – | – | – | – | – | – | – | – | |

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 Atmel ATmega48/88/168 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.
5. Only valid for ATmega88/168

# 8. 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 ← 0xFF – Rd | Z, C, N, V | 1 |
| NEG | Rd | Two's complement | Rd ← 0x00 – 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 • (0xFF - 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 ← 0xFF | 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[1] | 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[1] | 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 |
| **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 |

Note: 1. These instructions are only available in Atmel ATmega168.

# 9. Ordering information

## 9.1 Atmel ATmega48

| Speed (MHz) | Power supply | Ordering code[2] | Package[1] | Operational range |
|---|---|---|---|---|
| 10[3] | 1.8V - 5.5V | ATmega48V-10AUR[5]<br>ATmega48V-10MUR[5]<br>ATmega48V-10AU<br>ATmega48V-10MMU<br>ATmega48V-10MMUR[5]<br>ATmega48V-10MMH[4]<br>ATmega48V-10MMHR[4][5]<br>ATmega48V-10MU<br>ATmega48V-10PU | 32A<br>32M1-A<br>32A<br>28M1<br>28M1<br>28M1<br>28M1<br>32M1-A<br>28P3 | Industrial<br>(-40°C to 85°C) |
| 20[3] | 2.7V - 5.5V | ATmega48-20AUR[5]<br>ATmega48-20MUR[5]<br>ATmega48-20AU<br>ATmega48-20MMU<br>ATmega48-20MMUR[5]<br>ATmega48-20MMH[4]<br>ATmega48-20MMHR[4][5]<br>ATmega48-20MU<br>ATmega48-20PU | 32A<br>32M1-A<br>32A<br>28M1<br>28M1<br>28M1<br>28M1<br>32M1-A<br>28P3 | Industrial<br>(-40°C to 85°C) |

Note:
1. This device can also be supplied in wafer form. Please contact your local Atmel sales office for detailed ordering information and minimum quantities.
2. Pb-free packaging alternative, complies to the European Directive for Restriction of Hazardous Substances (RoHS directive). Also Halide free and fully Green.
3. See Figure 29-1 on page 312 and Figure 29-2 on page 312.
4. NiPdAu lead finish.
5. Tape & Reel.

| Package type | |
|---|---|
| **32A** | 32-lead, thin (1.0mm) plastic quad flat package (TQFP) |
| **28M1** | 28-pad, 4 × 4 × 1.0 body, lead pitch 0.45mm quad flat no-lead/micro lead frame package (QFN/MLF) |
| **32M1-A** | 32-pad, 5 × 5 × 1.0 body, lead pitch 0.50mm quad flat no-lead/micro lead frame package (QFN/MLF) |
| **28P3** | 28-lead, 0.300” wide, plastic dual inline package (PDIP) |

## 9.2 Atmel ATmega88

| Speed (MHz) | Power supply | Ordering code[2] | Package[1] | Operational range |
|---|---|---|---|---|
| 10[3] | 1.8V - 5.5V | ATmega88V-10AUR[4]<br>ATmega88V-10MUR[4]<br>ATmega88V-10AU<br>ATmega88V-10MU<br>ATmega88V-10PU | 32A<br>32M1-A<br>32A<br>32M1-A<br>28P3 | Industrial<br>(-40°C to 85°C) |
| 20[3] | 2.7V - 5.5V | ATmega88-20AUR[4]<br>ATmega88-20MUR[4]<br>ATmega88-20AU<br>ATmega88-20MU<br>ATmega88-20PU | 32A<br>32M1-A<br>32A<br>32M1-A<br>28P3 | Industrial<br>(-40°C to 85°C) |

Note:
1. This device can also be supplied in wafer form. Please contact your local Atmel sales office for detailed ordering information and minimum quantities.
2. Pb-free packaging alternative, complies to the European Directive for Restriction of Hazardous Substances (RoHS directive). Also Halide free and fully Green.
3. See Figure 29-1 on page 312 and Figure 29-2 on page 312.
4. Tape & reel

| Package type | |
|---|---|
| **32A** | 32-lead, thin (1.0mm) plastic quad flat package (TQFP) |
| **32M1-A** | 32-pad, 5 × 5 × 1.0 body, lead pitch 0.50mm quad flat no-lead/micro lead frame package (QFN/MLF) |
| **28P3** | 28-lead, 0.300” wide, plastic dual inline package (PDIP) |

## 9.3 Atmel ATmega168

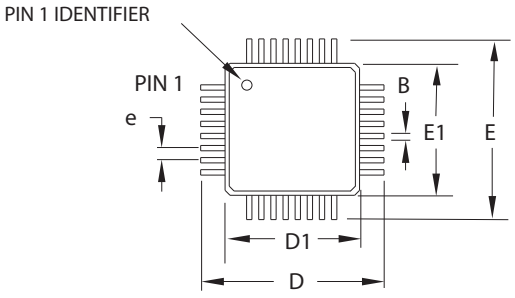
| Speed (MHz)[3] | Power supply | Ordering code[2] | Package[1] | Operational range |
|---|---|---|---|---|
| 10 | 1.8V - 5.5V | ATmega168V-10AUR[4]<br>ATmega168V-10MUR[4]<br>ATmega168V-10AU<br>ATmega168V-10MU<br>ATmega168V-10PU | 32A<br>32M1-A<br>32A<br>32M1-A<br>28P3 | Industrial<br>(-40°C to 85°C) |
| 20 | 2.7V - 5.5V | ATmega168-20AUR[4]<br>ATmega168-20MUR[4]<br>ATmega168-20AU<br>ATmega168-20MU<br>ATmega168-20PU | 32A<br>32M1-A<br>32A<br>32M1-A<br>28P3 | Industrial<br>(-40°C to 85°C) |

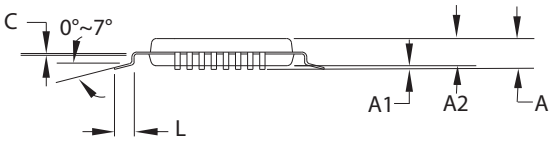
Note:
1. This device can also be supplied in wafer form. Please contact your local Atmel sales office for detailed ordering information and minimum quantities.
2. Pb-free packaging alternative, complies to the European Directive for Restriction of Hazardous Substances (RoHS directive). Also Halide free and fully Green.
3. See Figure 29-1 on page 312 and Figure 29-2 on page 312.
4. Tape & reel

| Package type | |
|---|---|
| **32A** | 32-lead, thin (1.0mm) plastic quad flat package (TQFP) |
| **32M1-A** | 32-pad, 5 × 5 × 1.0 body, lead pitch 0.50mm quad flat no-lead/micro lead frame package (QFN/MLF) |
| **28P3** | 28-lead, 0.300” wide, plastic dual inline package (PDIP) |

# 10. Packaging information

## 10.1 32A




COMMON DIMENSIONS
(Unit of measure = mm)

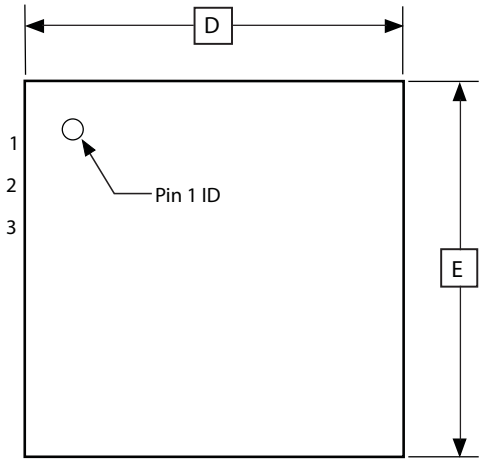
| SYMBOL | MIN | NOM | MAX | NOTE |
|---|---|---|---|---|
| A | – | – | 1.20 | |
| A1 | 0.05 | – | 0.15 | |
| A2 | 0.95 | 1.00 | 1.05 | |
| D | 8.75 | 9.00 | 9.25 | |
| D1 | 6.90 | 7.00 | 7.10 | Note 2 |
| E | 8.75 | 9.00 | 9.25 | |
| E1 | 6.90 | 7.00 | 7.10 | Note 2 |
| B | 0.30 | – | 0.45 | |
| C | 0.09 | – | 0.20 | |
| L | 0.45 | – | 0.75 | |
| e | 0.80 TYP | | | |

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

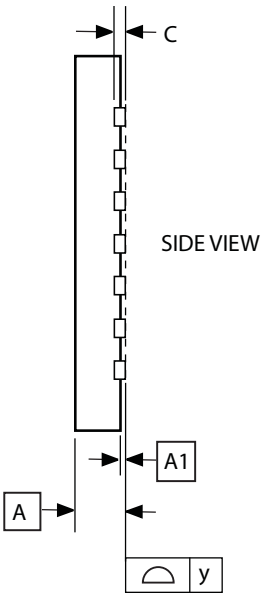
2010-10-20

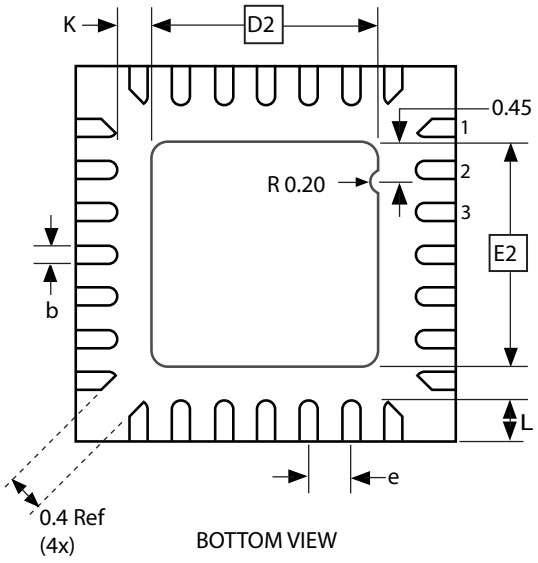
| | TITLE | DRAWING NO. | REV. |
|---|---|---|---|
| Atmel | 32A, 32-lead, 7 x 7mm body size, 1.0mm body thickness, 0.8mm lead pitch, thin profile plastic quad flat package (TQFP) | 32A | C |

## 10.2 28M1




TOP VIEW




BOTTOM VIEW

Note: The terminal #1 ID is a Laser-marked Feature.

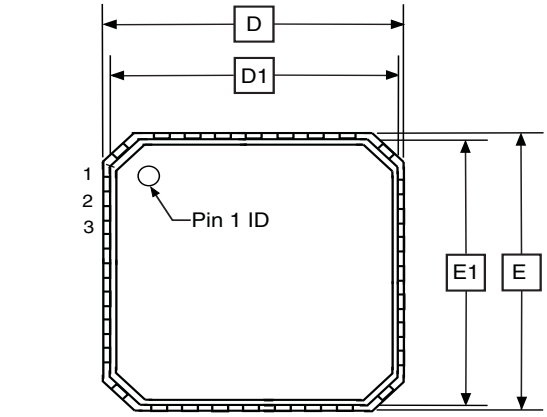
COMMON DIMENSIONS
(Unit of Measure = mm)

| SYMBOL | MIN | NOM | MAX | NOTE |
|---|---|---|---|---|
| A | 0.80 | 0.90 | 1.00 | |
| A1 | 0.00 | 0.02 | 0.05 | |
| b | 0.17 | 0.22 | 0.27 | |
| C | | 0.20 REF | | |
| D | 3.95 | 4.00 | 4.05 | |
| D2 | 2.35 | 2.40 | 2.45 | |
| E | 3.95 | 4.00 | 4.05 | |
| E2 | 2.35 | 2.40 | 2.45 | |
| e | | 0.45 | | |
| L | 0.35 | 0.40 | 0.45 | |
| y | 0.00 | – | 0.08 | |
| K | 0.20 | – | – | |

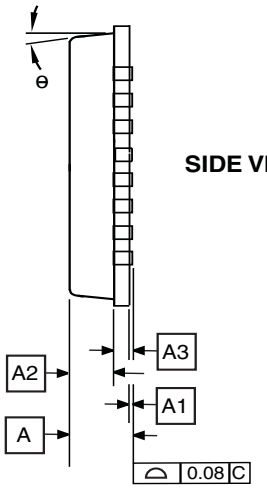
10/24/08

| | | | | |
|---|---|---|---|---|
| Atmel Package Drawing Contact: packagedrawings@atmel.com | TITLE<br>28M1, 28-pad, 4 x 4 x 1.0mm Body, Lead Pitch 0.45mm, 2.4 x 2.4mm Exposed Pad, Thermally Enhanced Plastic Very Thin Quad Flat No Lead Package (VQFN) | GPC<br>ZBV | DRAWING NO.<br>28M1 | REV.<br>B |

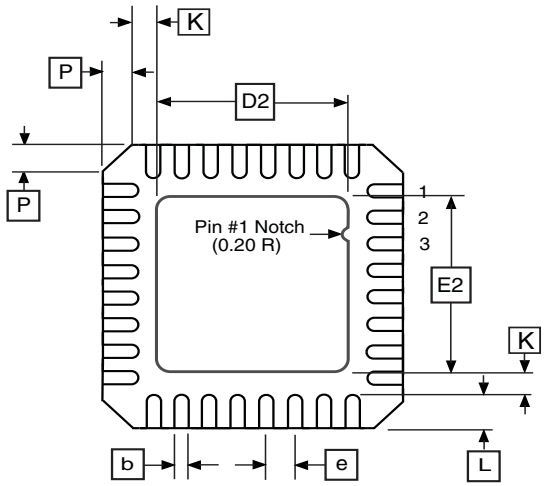
## 10.3 32M1-A




TOP VIEW




SIDE VIEW




BOTTOM VIEW

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

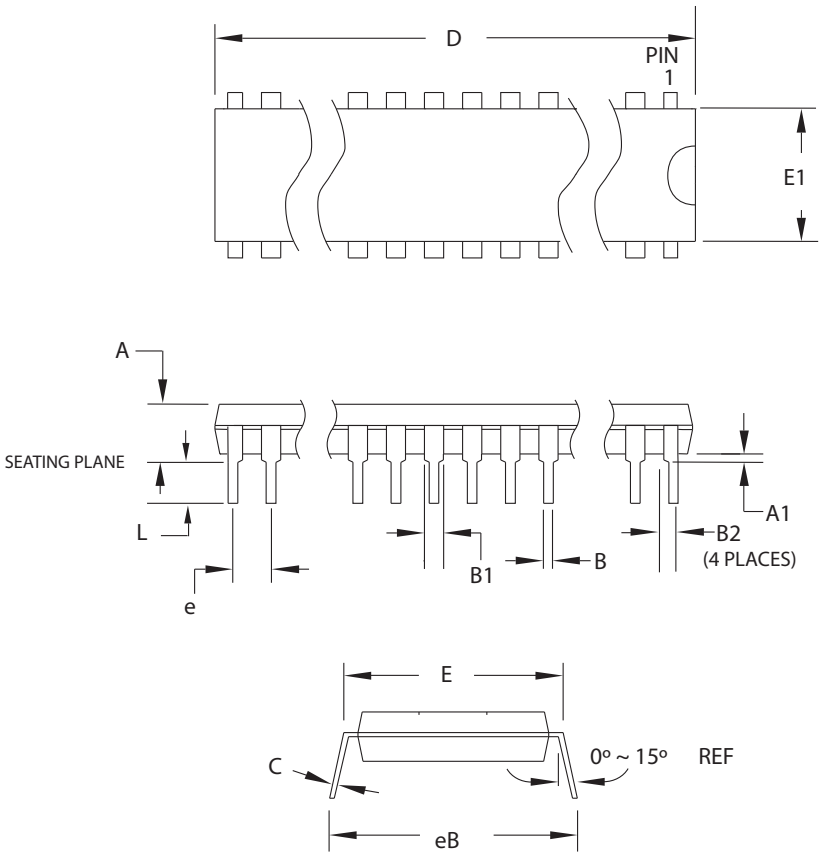
| SYMBOL | MIN | NOM | MAX | NOTE |
|---|---|---|---|---|
| A | 0.80 | 0.90 | 1.00 | |
| A1 | – | 0.02 | 0.05 | |
| A2 | – | 0.65 | 1.00 | |
| A3 | | 0.20 REF | | |
| b | 0.18 | 0.23 | 0.30 | |
| D | 4.90 | 5.00 | 5.10 | |
| D1 | 4.70 | 4.75 | 4.80 | |
| D2 | 2.95 | 3.10 | 3.25 | |
| E | 4.90 | 5.00 | 5.10 | |
| E1 | 4.70 | 4.75 | 4.80 | |
| E2 | 2.95 | 3.10 | 3.25 | |
| e | | 0.50 BSC | | |
| L | 0.30 | 0.40 | 0.50 | |
| P | – | – | 0.60 | |
| θ | – | – | $12^{o}$ | |
| K | 0.20 | – | – | |

Note: JEDEC Standard MO-220, Fig. 2 (Anvil Singulation), VHHD-2.

03/14/2014

| | TITLE | DRAWING NO. | REV. |
|---|---|---|---|
| Atmel Package Drawing Contact: packagedrawings@atmel.com | **32M1-A** , 32-pad, 5 x 5 x 1.0mm Body, Lead Pitch 0.50mm, 3.10mm Exposed Pad, Micro Lead Frame Package (MLF) | 32M1-A | F |

## 10.4 28P3




COMMON DIMENSIONS
(Unit of Measure = mm)

| SYMBOL | MIN | NOM | MAX | NOTE |
|---|---|---|---|---|
| A | – | – | 4.5724 | |
| A1 | 0.508 | – | – | |
| D | 34.544 | – | 34.798 | Note 1 |
| E | 7.620 | – | 8.255 | |
| E1 | 7.112 | – | 7.493 | Note 1 |
| B | 0.381 | – | 0.533 | |
| B1 | 1.143 | – | 1.397 | |
| B2 | 0.762 | – | 1.143 | |
| L | 3.175 | – | 3.429 | |
| C | 0.203 | – | 0.356 | |
| eB | – | – | 10.160 | |
| e | 2.540 TYP | | | |

Note: 1. Dimensions D and E1 do not include mold Flash or Protrusion. Mold Flash or Protrusion shall not exceed 0.25mm (0.010").

09/28/01

| | | | |
|---|---|---|---|
| Atmel 2325 Orchard Parkway San Jose, CA 95131 | TITLE<br>28P3, 28-lead (0.300"/7.62mm Wide) Plastic Dual Inline Package (PDIP) | DRAWING NO.<br>28P3 | REV.<br>B |

# 11. Errata

## 11.1 Errata Atmel ATmega48

The revision letter in this section refers to the revision of the ATmega48 device.

### 11.1.1 Rev K

- **Full swing crystal oscillator not supported**
- **Parallel programming timing modified**
- **Write wait delay for NVM is increased**
- **Changed device ID**

**1. Full swing crystal oscillator not supported**

The full swing crystal oscillator functionality is not available in revision K.

**Problem fix/workaround**

Use alternative clock sources available in the device.

**2. Parallel programming timing modified**

| | | Previous die revision | | | | Revision K | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Symbol** | **Parameter** | **Min** | **Typ.** | **Max** | **Units** | **Min** | **Typ.** | **Max** | **Units** |
| $t_{WLRH\_CE}$ | /WR Low to RDY/BSY High for Chip Erase | 7.5 | | 9 | ms | 9.8 | | 10.5 | ms |
| $t_{BVDV}$ | /BS1 Valid to DATA valid | 0 | | 250 | ns | 0 | | 335 | ns |
| $t_{OLDV}$ | /OE Low to DATA Valid | | | 250 | ns | | | 335 | ns |

**3 Write wait delay for NVM is increased**

The write delay for non-volatile memory (NVM) is increased as follows:

| | Other revisions | Revision K |
|---|---|---|
| **Symbol** | **Minimum Wait Delay** | **Minimum Wait Delay** |
| $t_{WD\_ERASE}$ | 9ms | 10.5ms |

**4. Changed device ID**

The device ID has been modified according to the to the following:

| | Any die revision | | | Previous die revision | Revision K |
|---|---|---|---|---|---|
| | Signature byte address ID (Unchanged) | | | Device ID read via debugWIRE | Device ID read via debugWIRE |
| **Part** | **0x000** | **0x001** | **0x002** | | |
| ATmega48 | 0x1E | 0x92 | 0x05 | 0x9205 | 0x920A |
| ATmega48V | 0x1E | 0x92 | 0x05 | 0x9205 | 0x920A |

### 11.1.2 Rev E to J

Not sampled.

### 11.1.3 Rev. D

- **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).

### 11.1.4 Rev. C

- **Reading EEPROM when system clock frequency is below 900kHz may not work**
- **Interrupts may be lost when writing the timer registers in the asynchronous timer**

1. **Reading EEPROM when system clock frequency is below 900kHz may not work**

   Reading Data from the EEPROM at system clock frequency below 900kHz may result in wrong data read.

   **Problem fix/workaround**

   Avoid using the EEPROM at clock frequency below 900kHz.

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

   The interrupt will be lost if a timer register that is 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).

### 11.1.5 Rev. B

- **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).

### 11.1.6 Rev A

- **Part may hang in reset**
- **Wrong values read after erase only operation**
- **Watchdog timer interrupt disabled**
- **Start-up time with crystal oscillator is higher than expected**
- **High power consumption in power-down with external clock**
- **Asynchronous oscillator does not stop in power-down**
- **Interrupts may be lost when writing the timer registers in the asynchronous timer**

**1. Part may hang in reset**

Some parts may get stuck in a reset state when a reset signal is applied when the internal reset state-machine is in a specific state. The internal reset state-machine is in this state for approximately 10ns immediately before the part wakes up after a reset, and in a 10ns window when altering the system clock prescaler. The problem is most often seen during In-System Programming of the device. There are theoretical possibilities of this happening also in run-mode. The following three cases can trigger the device to get stuck in a reset-state:

- Two succeeding resets are applied where the second reset occurs in the 10ns window before the device is out of the reset-state caused by the first reset.

- A reset is applied in a 10ns window while the system clock prescaler value is updated by software.

- Leaving SPI-programming mode generates an internal reset signal that can trigger this case.

The two first cases can occur during normal operating mode, while the last case occurs only during programming of the device.

**Problem fix/workaround**

The first case can be avoided during run-mode by ensuring that only one reset source is active. If an external reset push button is used, the reset start-up time should be selected such that the reset line is fully debounced during the start-up time.

The second case can be avoided by not using the system clock prescaler.

The third case occurs during In-System programming only. It is most frequently seen when using the internal RC at maximum frequency.

If the device gets stuck in the reset-state, turn power off, then on again to get the device out of this state.

**2. Wrong values read after erase only operation**

At supply voltages below 2.7V, an EEPROM location that is erased by the Erase Only operation may read as programmed (0x00).

**Problem fix/workaround**

If it is necessary to read an EEPROM location after Erase Only, use an Atomic Write operation with 0xFF as data in order to erase a location. In any case, the Write Only operation can be used as intended. Thus no special considerations are needed as long as the erased location is not read before it is programmed.

3. **Watchdog timer interrupt disabled**

   If the watchdog timer interrupt flag is not cleared before a new timeout occurs, the watchdog will be disabled, and the interrupt flag will automatically be cleared. This is only applicable in interrupt only mode. If the Watchdog is configured to reset the device in the watchdog time-out following an interrupt, the device works correctly.

   **Problem fix/workaround**

   Make sure there is enough time to always service the first timeout event before a new watchdog timeout occurs. This is done by selecting a long enough time-out period.

4. **Start-up time with crystal oscillator is higher than expected**

   The clock counting part of the start-up time is about two times higher than expected for all start-up periods when running on an external Crystal. This applies only when waking up by reset. Wake-up from power down is not affected. For most settings, the clock counting parts is a small fraction of the overall start-up time, and thus, the problem can be ignored. The exception is when using a very low frequency crystal like for instance a 32kHz clock crystal.

   **Problem fix/workaround**

   No known workaround.

5. **High power consumption in power-down with external clock**

   The power consumption in power down with an active external clock is about 10 times higher than when using internal RC or external oscillators.

   **Problem fix/workaround**

   Stop the external clock when the device is in power down.

6. **Asynchronous oscillator does not stop in power-down**

   The Asynchronous oscillator does not stop when entering power down mode. This leads to higher power consumption than expected.

   **Problem fix/workaround**

   Manually disable the asynchronous timer before entering power down.

7. **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).

## 11.2 Errata Atmel ATmega88

The revision letter in this section refers to the revision of the ATmega88 device.

### 11.2.1 Rev K

- **Full swing crystal oscillator not supported**
- **Parallel programming timing modified**
- **Write wait delay for NVM is increased**
- **Changed device ID**
- **Interrupts may be lost when writing the timer registers in the asynchronous timer**

**1. Full swing crystal oscillator not supported**

The full swing crystal oscillator functionality is not available in revision K.

**Problem fix/workaround**

Use alternative clock sources available in the device.

**2. Parallel programming timing modified**

| | | Previous die revision | | | | Revision K | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Symbol** | **Parameter** | **Min** | **Typ.** | **Max** | **Units** | **Min** | **Typ.** | **Max** | **Units** |
| $t_{WLRH\_CE}$ | /WR Low to RDY/BSY High for Chip Erase | 7.5 | | 9 | ms | 9.8 | | 10.5 | ms |
| $t_{BVDV}$ | /BS1 Valid to DATA valid | 0 | | 250 | ns | 0 | | 335 | ns |
| $t_{OLDV}$ | /OE Low to DATA Valid | | | 250 | ns | | | 335 | ns |

**3 Write wait delay for NVM is increased**

The write delay for non-volatile memory (NVM) is increased as follows:

| | Other revisions | Revision K |
|---|---|---|
| **Symbol** | **Minimum Wait Delay** | **Minimum Wait Delay** |
| $t_{WD\_ERASE}$ | 9ms | 10.5ms |

**4. Changed device ID**

The device ID has been modified according to the to the following:

| | Any die revision | | | Previous die revision | Revision K |
|---|---|---|---|---|---|
| | Signature byte address ID (Unchanged) | | | Device ID read via debugWIRE | Device ID read via debugWIRE |
| **Part** | **0x000** | **0x001** | **0x002** | | |
| ATmega88 | 0x1E | 0x93 | 0x0A | 0x930A | 0x930F |
| ATmega88V | 0x1E | 0x93 | 0x0A | 0x930A | 0x930F |

5. **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).

### 11.2.2 Rev E to J

Not sampled.

### 11.2.3 Rev. D

- **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).

### 11.2.4 Rev. B/C

Not sampled.

### 11.2.5 Rev. A

- **Writing to EEPROM does not work at low operating voltages**
- **Part may hang in reset**
- **Interrupts may be lost when writing the timer registers in the asynchronous timer**

1. **Writing to EEPROM does not work at low operating voltages**

   Writing to the EEPROM does not work at low voltages.

   **Problem fix/workaround**

   Do not write the EEPROM at voltages below 4.5 Volts.
   This will be corrected in rev. B.

2. **Part may hang in reset**

   Some parts may get stuck in a reset state when a reset signal is applied when the internal reset state-machine is in a specific state. The internal reset state-machine is in this state for approximately 10ns immediately before the part wakes up after a reset, and in a 10ns window when altering the system clock prescaler. The problem is most often seen during In-System Programming of the device. There are theoretical possibilities of this happening

also in run-mode. The following three cases can trigger the device to get stuck in a reset-state:

- Two succeeding resets are applied where the second reset occurs in the 10ns window before the device is out of the reset-state caused by the first reset.

- A reset is applied in a 10ns window while the system clock prescaler value is updated by software.

- Leaving SPI-programming mode generates an internal reset signal that can trigger this case.

The two first cases can occur during normal operating mode, while the last case occurs only during programming of the device.b.

**Problem fix/workaround**

The first case can be avoided during run-mode by ensuring that only one reset source is active. If an external reset push button is used, the reset start-up time should be selected such that the reset line is fully debounced during the start-up time.

The second case can be avoided by not using the system clock prescaler.

The third case occurs during In-System programming only. It is most frequently seen when using the internal RC at maximum frequency.

If the device gets stuck in the reset-state, turn power off, then on again to get the device out of this state.

**3. 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).

## 11.3 Errata Atmel ATmega168

The revision letter in this section refers to the revision of the ATmega168 device.

### 11.3.1 Rev K

- **Full swing crystal oscillator not supported**
- **Parallel programming timing modified**
- **Write wait delay for NVM is increased**
- **Changed device ID**
- **Interrupts may be lost when writing the timer registers in the asynchronous timer**

**1. Full swing crystal oscillator not supported**

The full swing crystal oscillator functionality is not available in revision K.

**Problem fix/workaround**

Use alternative clock sources available in the device.

**2. Parallel programming timing modified**

| | | Previous die revision | | | | Revision K | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Symbol** | **Parameter** | **Min** | **Typ.** | **Max** | **Units** | **Min** | **Typ.** | **Max** | **Units** |
| $t_{WLRH\_CE}$ | /WR Low to RDY/BSY High for Chip Erase | 7.5 | | 9 | ms | 9.8 | | 10.5 | ms |
| $t_{BVDV}$ | /BS1 Valid to DATA valid | 0 | | 250 | ns | 0 | | 335 | ns |
| $t_{OLDV}$ | /OE Low to DATA Valid | | | 250 | ns | | | 335 | ns |

**3 Write wait delay for NVM is increased**

The write delay for non-volatile memory (NVM) is increased as follows:

| | Other revisions | Revision K |
|---|---|---|
| **Symbol** | **Minimum Wait Delay** | **Minimum Wait Delay** |
| $t_{WD\_ERASE}$ | 9ms | 10.5ms |

**4. Changed device ID**

The device ID has been modified according to the to the following:

| | Any die revision | | | Previous die revision | Revision K |
|---|---|---|---|---|---|
| | Signature byte address ID (Unchanged) | | | Device ID read via debugWIRE | Device ID read via debugWIRE |
| **Part** | **0x000** | **0x001** | **0x002** | | |
| ATmega168 | 0x1E | 0x94 | 0x06 | 0x9406 | 0x940B |
| ATmega168V | 0x1E | 0x94 | 0x06 | 0x9406 | 0x940B |

5. **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).

### 11.3.2 Rev D to J

Not sampled.

### 11.3.3 Rev C

- **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).

### 11.3.4 Rev B

- **Part may hang in reset**
- **Interrupts may be lost when writing the timer registers in the asynchronous timer**

1. **Part may hang in reset**

   Some parts may get stuck in a reset state when a reset signal is applied when the internal reset state-machine is in a specific state. The internal reset state-machine is in this state for approximately 10ns immediately before the part wakes up after a reset, and in a 10ns window when altering the system clock prescaler. The problem is most often seen during In-System Programming of the device. There are theoretical possibilities of this happening also in run-mode. The following three cases can trigger the device to get stuck in a reset-state:

   - Two succeeding resets are applied where the second reset occurs in the 10ns window before the device is out of the reset-state caused by the first reset.

   - A reset is applied in a 10ns window while the system clock prescaler value is updated by software.

   - Leaving SPI-programming mode generates an internal reset signal that can trigger this case.

   The two first cases can occur during normal operating mode, while the last case occurs only during programming of the device.

   **Problem fix/workaround**

The first case can be avoided during run-mode by ensuring that only one reset source is active. If an external reset push button is used, the reset start-up time should be selected such that the reset line is fully debounced during the start-up time.

The second case can be avoided by not using the system clock prescaler.

The third case occurs during In-System programming only. It is most frequently seen when using the internal RC at maximum frequency.

If the device gets stuck in the reset-state, turn power off, then on again to get the device out of this state.

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

The interrupt will be lost if a timer register that is 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).

### 11.3.5 Rev A

- **Wrong values read after erase only operation**
- **Part may hang in reset**
- **Interrupts may be lost when writing the timer registers in the asynchronous timer**

**1. Wrong values read after erase only operation**

At supply voltages below 2.7V, an EEPROM location that is erased by the Erase Only operation may read as programmed (0x00).

**Problem fix/workaround**

If it is necessary to read an EEPROM location after Erase Only, use an Atomic Write operation with 0xFF as data in order to erase a location. In any case, the Write Only operation can be used as intended. Thus no special considerations are needed as long as the erased location is not read before it is programmed.

**2. Part may hang in reset**

Some parts may get stuck in a reset state when a reset signal is applied when the internal reset state-machine is in a specific state. The internal reset state-machine is in this state for approximately 10ns immediately before the part wakes up after a reset, and in a 10ns window when altering the system clock prescaler. The problem is most often seen during In-System Programming of the device. There are theoretical possibilities of this happening also in run-mode. The following three cases can trigger the device to get stuck in a reset-state:

- Two succeeding resets are applied where the second reset occurs in the 10ns window before the device is out of the reset-state caused by the first reset.

- A reset is applied in a 10ns window while the system clock prescaler value is updated by software.

- Leaving SPI-programming mode generates an internal reset signal that can trigger this case.

The two first cases can occur during normal operating mode, while the last case occurs only during programming of the device.

**Problem fix/workaround**

The first case can be avoided during run-mode by ensuring that only one reset source is active. If an external reset push button is used, the reset start-up time should be selected such that the reset line is fully debounced during the start-up time.

The second case can be avoided by not using the system clock prescaler.

The third case occurs during In-System programming only. It is most frequently seen when using the internal RC at maximum frequency.

If the device gets stuck in the reset-state, turn power off, then on again to get the device out of this state.

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

The interrupt will be lost if a timer register that is 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).

# 12. 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.

## 12.1 Rev. 2545U-11/15

1. Updated errata sections:
   - “Errata Atmel ATmega48” on page 23: Added errata for rev E to K.
   - “Errata Atmel ATmega88” on page 27: Added errata for rev E to K.
   - “Errata Atmel ATmega168” on page 30: Added errata for rev D to K.

## 12.2 Rev. 2545T-04/11

1. Ordering information has been updated by removing AI and MI and added AUR and MUR (tape & reel).
2. Added and corrected cross references and short-cuts.
3. Document updated according to new Atmel standard.
4. QTouch Library Support Features

## 12.3 Rev. 2545S-07/10

1. Note 6 and Note 7 in Table 29-5, “2-wire serial bus requirements.,” on page 315 have been removed.
2. Document updated according to Atmel standard.

## 12.4 Rev. 2545R-07/09

1. Updated “Errata” on page 23.
2. Updated the last page with the Atmel new addresses.

## 12.5 Rev. 2545Q-06/09

1. Removed the heading “About”. The subsections of this sectionis now separate sections, “Resources”, “Data Retention” and “About Code Examples”
2. Updated “Ordering information” on page 16.

## 12.6 Rev. 2545P-02/09

1. Removed Power-off slope rate from Table 29-3 on page 314.

## 12.7 Rev. 2545O-02/09

1. Changed minimum Power-on Reset Threshold Voltage (falling) to 0.05V in Table 29-3 on page 314.
2. Removed section “Power-on slope rate” from “System and reset characteristics” on page 314.

## 12.8 Rev. 2545N-01/09

1. Updated “Features” on page 1 and added the note “Not recommended for new designs”.
2. Merged the sections Resources, Data Retention and About Code Examples under one common section, “Resources” on page 8.
3. Updated Figure 9-4 on page 35.
4. Updated “System clock prescaler” on page 36.
5. Updated “Alternate functions of port B” on page 83.
6. Added section “” on page 314.
7. Updated “Pin thresholds and hysteresis” on page 337.

## 12.9 Rev. 2545M-09/07

1. Added “Data retention” on page 8.
2. Updated “ADC characteristics” on page 318.
3. “Preliminary“ removed through the datasheet.

## 12.10 Rev. 2545L-08/07

1. Updated “Features” on page 1.
2. Updated code example in “MCUCR – MCU control register” on page 67.
3. Updated “System and reset characteristics” on page 314.
4. Updated Note in Table 9-3 on page 30, Table 9-5 on page 31, Table 9-8 on page 33, Table 9-10 on page 34.

## 12.11 Rev. 2545K-04/07

1. Updated “Interrupts” on page 56.
2. Updated“Errata Atmel ATmega48” on page 23 .
3. Changed description in “Analog-to-digital converter” on page 250.

## 12.12 Rev. 2545J-12/06

1. Updated “Features” on page 1.
2. Updated Table 1-1 on page 2.
3. Updated “Ordering information” on page 16.
4. Updated “Packaging information” on page 19.

## 12.13 Rev. 2545I-11/06

1. Updated “Features” on page 1.
2. Updated Features in “2-wire serial interface” on page 213.
3. Fixed typos in Table 29-3 on page 314.

## 12.14 Rev. 2545H-10/06

1. Updated typos.
2. Updated “Features” on page 1.
3. Updated “Calibrated internal RC oscillator” on page 33.
4. Updated “System control and reset” on page 45.
5. Updated “Brown-out detection” on page 47.
6. Updated “Fast PWM mode” on page 126.
7. Updated bit description in “TCCR1C – Timer/Counter1 control register C” on page 137.
8. Updated code example in “SPI – Serial peripheral interface” on page 165.
9. Updated Table 15-3 on page 106, Table 15-6 on page 107, Table 15-8 on page 108, Table 16-2 on page 134, Table 16-3 on page 135, Table 16-4 on page 136, Table 18-3 on page 158, Table 18-6 on page 159, Table 18-8 on page 160, and Table 28-5 on page 294.
10. Added Note to Table 26-1 on page 271, Table 27-5 on page 285, and Table 28-17 on page 307.
11. Updated “Setting the boot loader lock bits by SPM” on page 283.
12. Updated “Signature bytes” on page 295
13. Updated “Electrical characteristics” on page 310.
14. Updated “Errata” on page 23.

## 12.15 Rev. 2545G-06/06

1. Added Addresses in Registers.
2. Updated “Calibrated internal RC oscillator” on page 33.
3. Updated Table 9-12 on page 35, Table 10-1 on page 39, Table 11-1 on page 54, Table 14-3 on page 83.
4. Updated “ADC noise reduction mode” on page 40.
5. Updated note for Table 10-2 on page 43.
6. Updatad “Bit 2 - PRSPI: Power reduction serial peripheral interface” on page 44.
7. Updated “TCCR0B – Timer/counter control register B” on page 109.
8. Updated “Fast PWM mode” on page 126.
9. Updated “Asynchronous operation of Timer/Counter2” on page 155.
10. Updated “SPI – Serial peripheral interface” on page 165.
11. Updated “UCSRnA – USART MSPIM control and status register n A” on page 210.
12. Updated note in “Bit rate generator unit” on page 220.
13. Updated “Bit 6 – ACBG: Analog comparator bandgap select” on page 247.
14. Updated Features in “Analog-to-digital converter” on page 250.
15. Updated “Prescaling and conversion timing” on page 253.
16. Updated “Limitations of debugWIRE” on page 267.
17. Added Table 29-1 on page 313.
18. Updated Figure 16-7 on page 127, Figure 30-45 on page 346.
19. Updated rev. A in “Errata Atmel ATmega48” on page 23.
20. Added rev. C and D in “Errata Atmel ATmega48” on page 23.

## 12.16 Rev. 2545F-05/05

1. Added Section 3. “Resources” on page 8
2. Update Section 9.6 “Calibrated internal RC oscillator” on page 33.
3. Updated Section 28.8.3 “Serial programming instruction set” on page 307.
4. Table notes in Section 29.2 “DC characteristics” on page 310 updated.
5. Updated Section 11. “Errata” on page 23.

## 12.17 Rev. 2545E-02/05

1. MLF-package alternative changed to “Quad Flat No-Lead/Micro Lead Frame Package QFN/MLF”.
2. Updated “EECR – The EEPROM control register” on page 22.
3. Updated “Calibrated internal RC oscillator” on page 33.
4. Updated “External clock” on page 35.
5. Updated Table 29-3 on page 314, Table 29-6 on page 316, Table 29-2 on page 313 and Table 28-16 on page 307
6. Added “Pin change interrupt timing” on page 70
7. Updated “8-bit timer/counter block diagram.” on page 95.
8. Updated “SPMCSR – Store program memory control and status register” on page 273.

9. Updated “Enter programming mode” on page 298.
10. Updated “DC characteristics” on page 310.
11. Updated “Ordering information” on page 16.
12. Updated “Errata Atmel ATmega88” on page 27 and “Errata Atmel ATmega168” on page 30.

## 12.18 Rev. 2545D-07/04

1. Updated instructions used with WDTCSR in relevant code examples.
2. Updated Table 9-5 on page 31, Table 29-4 on page 314, Table 27-9 on page 288, and Table 27-11 on page 290.
3. Updated “System clock prescaler” on page 36.
4. Moved “TIMSK2 – Timer/Counter2 interrupt mask register” on page 162 and “TIFR2 – Timer/Counter2 interrupt flag register” on page 162 to “Register description” on page 157.
5. Updated cross-reference in “Electrical interconnection” on page 214.
6. Updated equation in “Bit rate generator unit” on page 220.
7. Added “Page size” on page 296.
8. Updated “Serial programming algorithm” on page 306.
9. Updated Ordering Information for “Atmel ATmega168” on page 18.
10. Updated “Errata Atmel ATmega88” on page 27 and “Errata Atmel ATmega168” on page 30.
11. Updated equation in “Bit rate generator unit” on page 220.

## 12.19 Rev. 2545C-04/04

1. Speed Grades changed: 12MHz to 10MHz and 24MHz to 20MHz
2. Updated “Speed grades” on page 312.
3. Updated “Ordering information” on page 16.
4. Updated “Errata Atmel ATmega88” on page 27.

## 12.20 Rev. 2545B-01/04

1. Added PDIP to “I/O and Packages”, updated “Speed Grade” and Power Consumption Estimates in 12.“Features” on page 1.
2. Updated “Stack pointer” on page 13 with RAMEND as recommended Stack Pointer value.
3. Added section “Power reduction register” on page 41 and a note regarding the use of the PRR bits to 2-wire, Timer/Counters, USART, Analog Comparator and ADC sections.
4. Updated “Watchdog timer” on page 49.
5. Updated Figure 16-2 on page 134 and Table 16-3 on page 135.
6. Extra Compare Match Interrupt OCF2B added to features in section “8-bit Timer/Counter2 with PWM and asynchronous operation” on page 144

7. Updated Table 10-1 on page 39, Table 24-5 on page 265, Table 28-4 to Table 28-7 on page 293 to 295 and Table 24-1 on page 255. Added note 2 to Table 28-1 on page 292. Fixed typo in Table 13-1 on page 71.
8. Updated whole “Typical characteristics” on page 322.
9. Added item 2 to 5 in “Errata Atmel ATmega48” on page 23.
10. Renamed the following bits:
    - SPMEN to SELFPRGEN
    - PSR2 to PSRASY
    - PSR10 to PSRSYNC
    - Watchdog Reset to Watchdog System Reset
11. Updated C code examples containing old IAR syntax.
12. Updated BLBSET description in “SPMCSR – Store program memory control and status register” on page 290.

Rev. 2545US-AVR-11/2015