# ATmega128A

## 8-bit Microcontroller with 128Kbytes In-System Programmable Flash

## DATASHEET SUMMARY

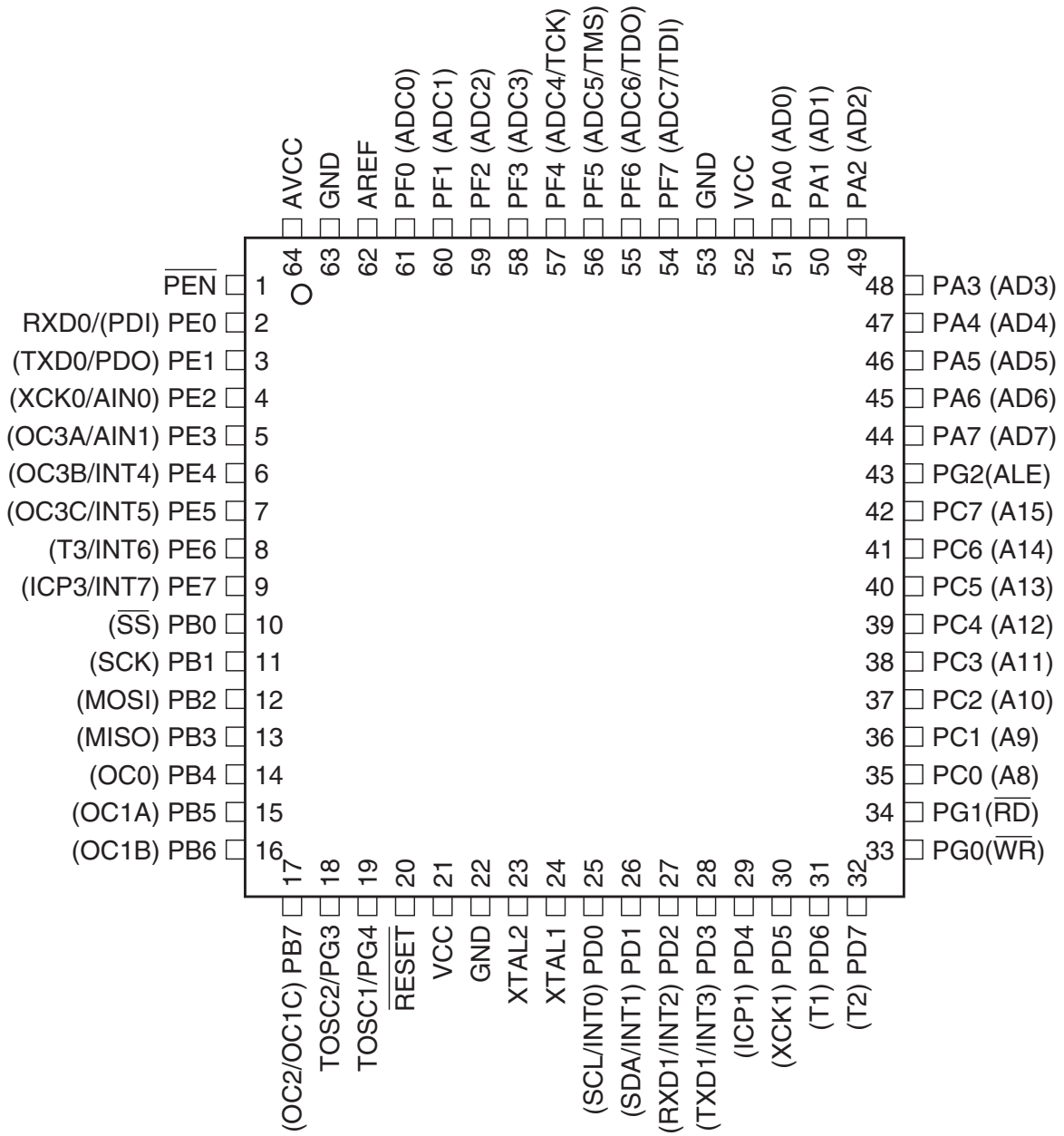
## Features

- High-performance, Low-power Atmel® AVR® 8-bit Microcontroller
- Advanced RISC Architecture
  - 133 Powerful Instructions – Most Single Clock Cycle Execution
  - 32 × 8 General Purpose Working Registers + Peripheral Control Registers
  - Fully Static Operation
  - Up to 16MHz Throughput at 16MIPS
  - On-chip 2-cycle Multiplier
- High Endurance Non-volatile Memory segments
  - 128Kbytes of In-System Self-programmable Flash program memory
  - 4Kbytes EEPROM
  - 4Kbytes Internal SRAM
  - Write/Erase cycles: 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
  - Up to 64 Kbytes Optional External Memory Space
  - Programming Lock for Software Security
  - SPI Interface for In-System Programming
- JTAG (IEEE std. 1149.1 Compliant) Interface
  - Boundary-scan Capabilities According to the JTAG Standard
  - Extensive On-chip Debug Support
  - Programming of Flash, EEPROM, Fuses and Lock Bits through the JTAG Interface
- Peripheral Features
  - Two 8-bit Timer/Counters with Separate Prescalers and Compare Modes
  - Two Expanded 16-bit Timer/Counters with Separate Prescaler, Compare Mode and Capture Mode
  - Real Time Counter with Separate Oscillator
  - Two 8-bit PWM Channels
  - 6 PWM Channels with Programmable Resolution from 2 to 16 Bits
  - Output Compare Modulator
  - 8-channel, 10-bit ADC
    - 8 Single-ended Channels
    - 7 Differential Channels
    - 2 Differential Channels with Programmable Gain at 1x, 10x, or 200x
  - Byte-oriented Two-wire Serial Interface
  - Dual Programmable Serial USARTs
  - Master/Slave SPI Serial Interface

  - Programmable Watchdog Timer with On-chip Oscillator
  - On-chip Analog Comparator
- Special Microcontroller Features
  - Power-on Reset and Programmable Brown-out Detection
  - Internal Calibrated RC Oscillator
  - External and Internal Interrupt Sources
  - Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby, and Extended Standby
  - Software Selectable Clock Frequency
  - ATmega103 Compatibility Mode Selected by a Fuse
  - Global Pull-up Disable
- I/O and Packages
  - 53 Programmable I/O Lines
  - 64-lead TQFP and 64-pad QFN/MLF
- Operating Voltages
  - 2.7V - 5.5V
- Speed Grades
  - 0 - 16MHz

# 1. Pin Configurations

**Figure 1-1. Pinout ATmega128A**



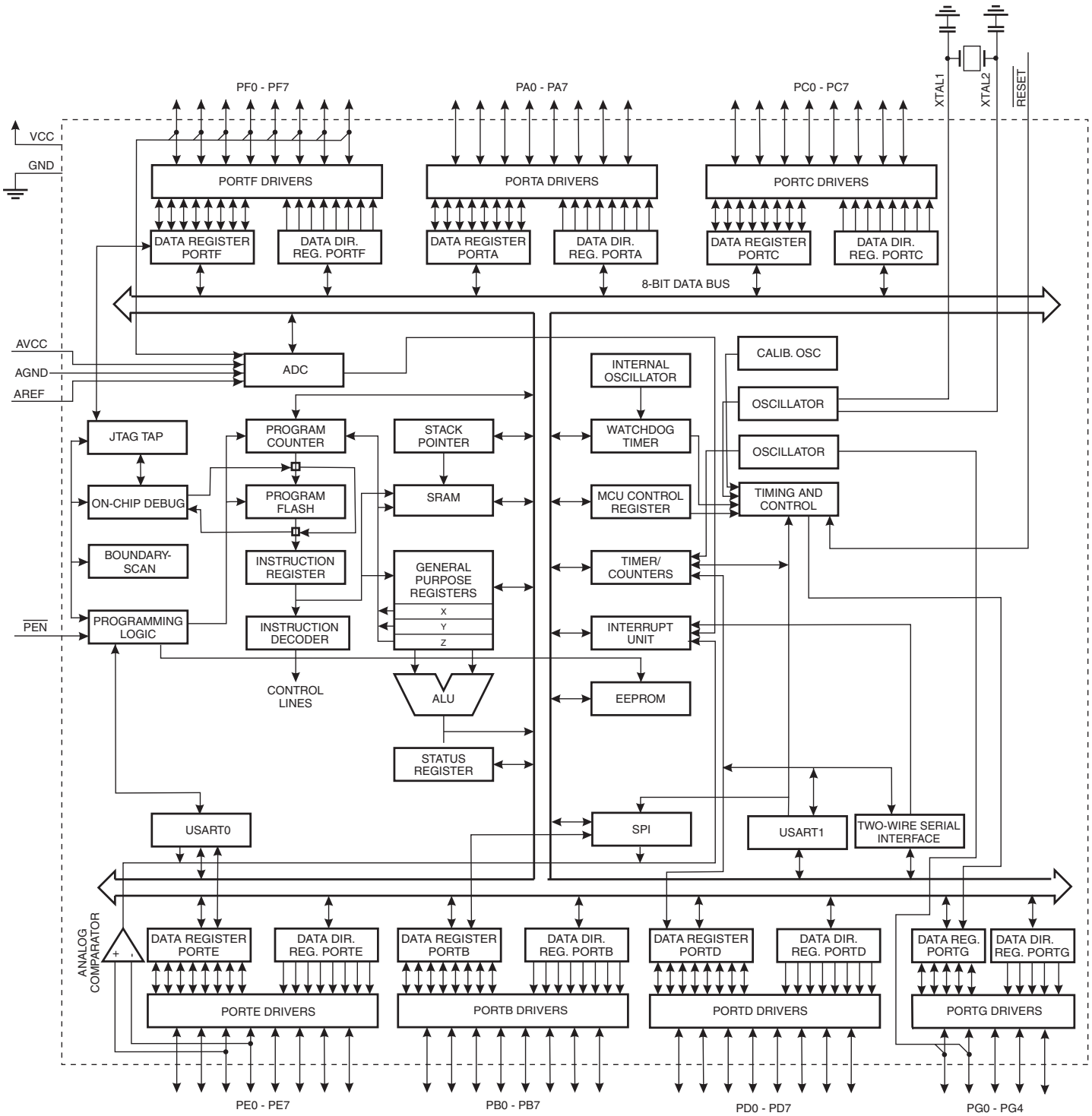

Note: The Pinout figure applies to both TQFP and MLF packages. The bottom pad under the QFN/MLF package should be soldered to ground.

# 2. Overview

The Atmel®AVR®ATmega128A 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 ATmega128A achieves throughputs approaching 1MIPS per MHz allowing the system designer to optimize power consumption versus processing speed.

## 2.1 Block Diagram

**Figure 2-1.** Block Diagram




The Atmel®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 ATmega128A provides the following features: 128 Kbytes of In-System Programmable Flash with Read-While-Write capabilities, 4 Kbytes EEPROM, 4 Kbytes SRAM, 53 general purpose I/O lines, 32 general purpose working registers, Real Time Counter (RTC), four flexible Timer/Counters with compare modes and PWM, 2 USARTs, a byte oriented Two-wire Serial Interface, an 8-channel, 10-bit ADC with optional differential input stage with programmable gain, programmable Watchdog Timer with Internal Oscillator, an SPI serial port, IEEE std. 1149.1 compliant JTAG test interface, also used for accessing the On-chip Debug system and programming and six 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. In Extended Standby mode, both the main Oscillator and the Asynchronous Timer continue to run.

The device is manufactured using Atmel's high-density nonvolatile memory technology. The On-chip ISP Flash allows the program memory to be reprogrammed in-system through an SPI serial interface, by a conventional nonvolatile memory programmer, or by an On-chip Boot program running on the AVR core. The boot program can use any interface to download the application program in the application Flash memory. Software in the Boot Flash section will continue to run while the Application Flash section is updated, providing true Read-While-Write operation. By combining an 8-bit RISC CPU with In-System Self-Programmable Flash on a monolithic chip, the Atmel ATmega128A is a powerful microcontroller that provides a highly flexible and cost effective solution to many embedded control applications.

The ATmega128A 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 ATmega103 and ATmega128A Compatibility

The ATmega128A is a highly complex microcontroller where the number of I/O locations supersedes the 64 I/O locations reserved in the AVR instruction set. To ensure backward compatibility with the ATmega103, all I/O locations present in ATmega103 have the same location in ATmega128A. Most additional I/O locations are added in an Extended I/O space starting from $60 to $FF, (that is, in the ATmega103 internal RAM space). These locations can be reached by using LD/LDS/LDD and ST/STS/STD instructions only, not by using IN and OUT instructions. The relocation of the internal RAM space may still be a problem for ATmega103 users. Also, the increased number of interrupt vectors might be a problem if the code uses absolute addresses. To solve these problems, an ATmega103 compatibility mode can be selected by programming the fuse M103C. In this mode, none of the functions in the Extended I/O space are in use, so the internal RAM is located as in ATmega103. Also, the Extended Interrupt vectors are removed.

The Atmel®AVR®ATmega128A is 100% pin compatible with ATmega103, and can replace the ATmega103 on current Printed Circuit Boards. The application note "Replacing ATmega103 by ATmega128A" describes what the user should be aware of replacing the ATmega103 by an ATmega128A.

### 2.2.1 ATmega103 Compatibility Mode

By programming the M103C fuse, the ATmega128A will be compatible with the ATmega103 regards to RAM, I/O pins and interrupt vectors as described above. However, some new features in ATmega128A are not available in this compatibility mode, these features are listed below:

- One USART instead of two, Asynchronous mode only. Only the eight least significant bits of the Baud Rate Register is available.
- One 16 bits Timer/Counter with two compare registers instead of two 16-bit Timer/Counters with three compare registers.
- Two-wire serial interface is not supported.
- Port C is output only.
- Port G serves alternate functions only (not a general I/O port).
- Port F serves as digital input only in addition to analog input to the ADC.
- Boot Loader capabilities is not supported.
- It is not possible to adjust the frequency of the internal calibrated RC Oscillator.
- The External Memory Interface can not release any Address pins for general I/O, neither configure different wait-states to different External Memory Address sections.
- In addition, there are some other minor differences to make it more compatible to ATmega103:
- Only EXTRF and PORF exists in MCUCSR.
- Timed sequence not required for Watchdog Time-out change.
- External Interrupt pins 3 - 0 serve as level interrupt only.
- USART has no FIFO buffer, so data overrun comes earlier.

Unused I/O bits in ATmega103 should be written to 0 to ensure same operation in ATmega128A.

## 2.3 Pin Descriptions

### 2.3.1 VCC

Digital supply voltage.

### 2.3.2 GND

Ground.

### 2.3.3 Port A (PA7:PA0)

Port A is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port A output buffers have symmetrical drive characteristics with both high sink and source capability. 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.

Port A also serves the functions of various special features of the ATmega128A as listed on page 71.

### 2.3.4 Port B (PB7:PB0)

Port B is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port B output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port B pins that are externally pulled low will source current if the pull-up resistors are activated. The Port B pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port B also serves the functions of various special features of the ATmega128A as listed on page 72.

### 2.3.5 Port C (PC7:PC0)

Port C is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port C output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port C pins that are externally pulled low will source current if the pull-up resistors are activated. The Port C pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port C also serves the functions of special features of the ATmega128A as listed on page 74. In ATmega103 compatibility mode, Port C is output only, and the port C pins are **not** tri-stated when a reset condition becomes active.

Note: The Atmel®AVR® ATmega128A is by default shipped in ATmega103 compatibility mode. Thus, if the parts are not programmed before they are put on the PCB, PORTC will be output during first power up, and until the ATmega103 compatibility mode is disabled.

### 2.3.6 Port D (PD7:PD0)

Port D is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port D output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port D pins that are externally pulled low will source current if the pull-up resistors are activated. The Port D pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port D also serves the functions of various special features of the ATmega128A as listed on page 76.

### 2.3.7 Port E (PE7:PE0)

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.

Port E also serves the functions of various special features of the ATmega128A as listed on page 78.

### 2.3.8 Port F (PF7:PF0)

Port F serves as the analog inputs to the A/D Converter.

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

The TDO pin is tri-stated unless TAP states that shift out data are entered.

Port F also serves the functions of the JTAG interface.

In ATmega103 compatibility mode, Port F is an input Port only.

### 2.3.9 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.

The port G pins are tri-stated when a reset condition becomes active, even if the clock is not running.

In Atmel®AVR®ATmega103 compatibility mode, these pins only serves as strobes signals to the external memory as well as input to the 32kHz Oscillator, and the pins are initialized to PG0 = 1, PG1 = 1, and PG2 = 0 asynchronously when a reset condition becomes active, even if the clock is not running. PG3 and PG4 are oscillator pins.

### 2.3.10 $\overline{\text{RESET}}$

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 “System and Reset Characteristics” on page 306. Shorter pulses are not guaranteed to generate a reset.

### 2.3.11 XTAL1

Input to the inverting Oscillator amplifier and input to the internal clock operating circuit.

### 2.3.12 XTAL2

Output from the inverting Oscillator amplifier.

### 2.3.13 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.

### 2.3.14 AREF

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

### 2.3.15 PEN

PEN is a programming enable pin for the SPI Serial Programming mode, and is internally pulled high. By holding this pin low during a Power-on Reset, the device will enter the SPI Serial Programming mode. $\overline{\text{PEN}}$ has no function during normal operation.

## 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 datasheet 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. Register Summary

| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Page |
|---|---|---|---|---|---|---|---|---|---|---|
| ($FF) | Reserved | – | – | – | – | – | – | – | – | |
| : | Reserved | – | – | – | – | – | – | – | – | |
| ($9E) | Reserved | – | – | – | – | – | – | – | – | |
| ($9D) | UCSR1C | – | UMSEL1 | UPM11 | UPM10 | USBS1 | UCSZ11 | UCSZ10 | UCPOL1 | 186 |
| ($9C) | UDR1 | USART1 I/O Data Register | | | | | | | | 184 |
| ($9B) | UCSR1A | RXC1 | TXC1 | UDRE1 | FE1 | DOR1 | UPE1 | U2X1 | MPCM1 | 184 |
| ($9A) | UCSR1B | RXCIE1 | TXCIE1 | UDRIE1 | RXEN1 | TXEN1 | UCSZ12 | RXB81 | TXB81 | 185 |
| ($99) | UBRR1L | USART1 Baud Rate Register Low | | | | | | | | 188 |
| ($98) | UBRR1H | – | – | – | – | USART1 Baud Rate Register High | | | | 188 |
| ($97) | Reserved | – | – | – | – | – | – | – | – | |
| ($96) | Reserved | – | – | – | – | – | – | – | – | |
| ($95) | UCSR0C | – | UMSEL0 | UPM01 | UPM00 | USBS0 | UCSZ01 | UCSZ00 | UCPOL0 | 186 |
| ($94) | Reserved | – | – | – | – | – | – | – | – | |
| ($93) | Reserved | – | – | – | – | – | – | – | – | |
| ($92) | Reserved | – | – | – | – | – | – | – | – | |
| ($91) | Reserved | – | – | – | – | – | – | – | – | |
| ($90) | UBRR0H | – | – | – | – | USART0 Baud Rate Register High | | | | 188 |
| ($8F) | Reserved | – | – | – | – | – | – | – | – | |
| ($8E) | Reserved | – | – | – | – | – | – | – | – | |
| ($8D) | Reserved | – | – | – | – | – | – | – | – | |
| ($8C) | TCCR3C | FOC3A | FOC3B | FOC3C | – | – | – | – | – | 134 |
| ($8B) | TCCR3A | COM3A1 | COM3A0 | COM3B1 | COM3B0 | COM3C1 | COM3C0 | WGM31 | WGM30 | 130 |
| ($8A) | TCCR3B | ICNC3 | ICES3 | – | WGM33 | WGM32 | CS32 | CS31 | CS30 | 132 |
| ($89) | TCNT3H | Timer/Counter3 – Counter Register High Byte | | | | | | | | 134 |
| ($88) | TCNT3L | Timer/Counter3 – Counter Register Low Byte | | | | | | | | 134 |
| ($87) | OCR3AH | Timer/Counter3 – Output Compare Register A High Byte | | | | | | | | 135 |
| ($86) | OCR3AL | Timer/Counter3 – Output Compare Register A Low Byte | | | | | | | | 135 |
| ($85) | OCR3BH | Timer/Counter3 – Output Compare Register B High Byte | | | | | | | | 135 |
| ($84) | OCR3BL | Timer/Counter3 – Output Compare Register B Low Byte | | | | | | | | 135 |
| ($83) | OCR3CH | Timer/Counter3 – Output Compare Register C High Byte | | | | | | | | 135 |
| ($82) | OCR3CL | Timer/Counter3 – Output Compare Register C Low Byte | | | | | | | | 135 |
| ($81) | ICR3H | Timer/Counter3 – Input Capture Register High Byte | | | | | | | | 136 |
| ($80) | ICR3L | Timer/Counter3 – Input Capture Register Low Byte | | | | | | | | 136 |
| ($7F) | Reserved | – | – | – | – | – | – | – | – | |
| ($7E) | Reserved | – | – | – | – | – | – | – | – | |
| ($7D) | ETIMSK | – | – | TICIE3 | OCIE3A | OCIE3B | TOIE3 | OCIE3C | OCIE1C | 137 |
| ($7C) | ETIFR | – | – | ICF3 | OCF3A | OCF3B | TOV3 | OCF3C | OCF1C | 138 |
| ($7B) | Reserved | – | – | – | – | – | – | – | – | |
| ($7A) | TCCR1C | FOC1A | FOC1B | FOC1C | – | – | – | – | – | 133 |
| ($79) | OCR1CH | Timer/Counter1 – Output Compare Register C High Byte | | | | | | | | 135 |
| ($78) | OCR1CL | Timer/Counter1 – Output Compare Register C Low Byte | | | | | | | | 135 |
| ($77) | Reserved | – | – | – | – | – | – | – | – | |
| ($76) | Reserved | – | – | – | – | – | – | – | – | |
| ($75) | Reserved | – | – | – | – | – | – | – | – | |
| ($74) | TWCR | TWINT | TWEA | TWSTA | TWSTO | TWWC | TWEN | – | TWIE | 214 |
| ($73) | TWDR | Two-wire Serial Interface Data Register | | | | | | | | 216 |
| ($72) | TWAR | TWA6 | TWA5 | TWA4 | TWA3 | TWA2 | TWA1 | TWA0 | TWGCE | 216 |
| ($71) | TWSR | TWS7 | TWS6 | TWS5 | TWS4 | TWS3 | – | TWPS1 | TWPS0 | 215 |
| ($70) | TWBR | Two-wire Serial Interface Bit Rate Register | | | | | | | | 214 |
| ($6F) | OSCCAL | Oscillator Calibration Register | | | | | | | | 45 |
| ($6E) | Reserved | – | – | – | – | – | – | – | – | |
| ($6D) | XMCRA | – | SRL2 | SRL1 | SRL0 | SRW01 | SRW00 | SRW11 | | 35 |
| ($6C) | XMCRB | XMBK | – | – | – | – | XMM2 | XMM1 | XMM0 | 36 |
| ($6B) | Reserved | – | – | – | – | – | – | – | – | |
| ($6A) | EICRA | ISC31 | ISC30 | ISC21 | ISC20 | ISC11 | ISC10 | ISC01 | ISC00 | 88 |
| ($69) | Reserved | – | – | – | – | – | – | – | – | |
| ($68) | SPMCSR | SPMIE | RWWSB | – | RWWSRE | BLBSET | PGWRT | PGERS | SPMEN | 272 |
| ($67) | Reserved | – | – | – | – | – | – | – | – | |
| ($66) | Reserved | – | – | – | – | – | – | – | – | |
| ($65) | PORTG | – | – | – | PORTG4 | PORTG3 | PORTG2 | PORTG1 | PORTG0 | 87 |
| ($64) | DDRG | – | – | – | DDG4 | DDG3 | DDG2 | DDG1 | DDG0 | 87 |
| ($63) | PING | – | – | – | PING4 | PING3 | PING2 | PING1 | PING0 | 87 |
| ($62) | PORTF | PORTF7 | PORTF6 | PORTF5 | PORTF4 | PORTF3 | PORTF2 | PORTF1 | PORTF0 | 86 |
| ($61) | DDRF | DDF7 | DDF6 | DDF5 | DDF4 | DDF3 | DDF2 | DDF1 | DDF0 | 86 |
| ($60) | Reserved | – | – | – | – | – | – | – | – | |

# 6. Register Summary (Continued)

| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Page |
|---|---|---|---|---|---|---|---|---|---|---|
| $3F ($5F) | SREG | I | T | H | S | V | N | Z | C | 11 |
| $3E ($5E) | SPH | SP15 | SP14 | SP13 | SP12 | SP11 | SP10 | SP9 | SP8 | 14 |
| $3D ($5D) | SPL | SP7 | SP6 | SP5 | SP4 | SP3 | SP2 | SP1 | SP0 | 14 |
| $3C ($5C) | XDIV | XDIVEN | XDIV6 | XDIV5 | XDIV4 | XDIV3 | XDIV2 | XDIV1 | XDIV0 | 39 |
| $3B ($5B) | RAMPZ | – | – | – | – | – | – | – | RAMPZ0 | 15 |
| $3A ($5A) | EICRB | ISC71 | ISC70 | ISC61 | ISC60 | ISC51 | ISC50 | ISC41 | ISC40 | 89 |
| $39 ($59) | EIMSK | INT7 | INT6 | INT5 | INT4 | INT3 | INT2 | INT1 | INT0 | 90 |
| $38 ($58) | EIFR | INTF7 | INTF6 | INTF5 | INTF4 | INTF3 | INTF | INTF1 | INTF0 | 90 |
| $37 ($57) | TIMSK | OCIE2 | TOIE2 | TICIE1 | OCIE1A | OCIE1B | TOIE1 | OCIE0 | TOIE0 | 107, 136, 155 |
| $36 ($56) | TIFR | OCF2 | TOV2 | ICF1 | OCF1A | OCF1B | TOV1 | OCF0 | TOV0 | 107, 138, 155 |
| $35 ($55) | MCUCR | SRE | SRW10 | SE | SM1 | SM0 | SM2 | IVSEL | IVCE | 34, 50, 63 |
| $34 ($54) | MCUCSR | JTD | – | – | JTRF | WDRF | BORF | EXTRF | PORF | 56, 243 |
| $33 ($53) | TCCR0 | FOC0 | WGM00 | COM01 | COM00 | WGM01 | CS02 | CS01 | CS00 | 104 |
| $32 ($52) | TCNT0 | Timer/Counter0 (8 Bit) | | | | | | | | 106 |
| $31 ($51) | OCR0 | Timer/Counter0 Output Compare Register | | | | | | | | 106 |
| $30 ($50) | ASSR | – | – | – | – | AS0 | TCN0UB | OCR0UB | TCR0UB | 106 |
| $2F ($4F) | TCCR1A | COM1A1 | COM1A0 | COM1B1 | COM1B0 | COM1C1 | COM1C0 | WGM11 | WGM10 | 130 |
| $2E ($4E) | TCCR1B | ICNC1 | ICES1 | – | WGM13 | WGM12 | CS12 | CS11 | CS10 | 132 |
| $2D ($4D) | TCNT1H | Timer/Counter1 – Counter Register High Byte | | | | | | | | 134 |
| $2C ($4C) | TCNT1L | Timer/Counter1 – Counter Register Low Byte | | | | | | | | 134 |
| $2B ($4B) | OCR1AH | Timer/Counter1 – Output Compare Register A High Byte | | | | | | | | 135 |
| $2A ($4A) | OCR1AL | Timer/Counter1 – Output Compare Register A Low Byte | | | | | | | | 135 |
| $29 ($49) | OCR1BH | Timer/Counter1 – Output Compare Register B High Byte | | | | | | | | 135 |
| $28 ($48) | OCR1BL | Timer/Counter1 – Output Compare Register B Low Byte | | | | | | | | 135 |
| $27 ($47) | ICR1H | Timer/Counter1 – Input Capture Register High Byte | | | | | | | | 136 |
| $26 ($46) | ICR1L | Timer/Counter1 – Input Capture Register Low Byte | | | | | | | | 136 |
| $25 ($45) | TCCR2 | FOC2 | WGM20 | COM21 | COM20 | WGM21 | CS22 | CS21 | CS20 | 152 |
| $24 ($44) | TCNT2 | Timer/Counter2 (8 Bit) | | | | | | | | 155 |
| $23 ($43) | OCR2 | Timer/Counter2 Output Compare Register | | | | | | | | 155 |
| $22 ($42) | OCDR | IDRD/OCDR7 | OCDR6 | OCDR5 | OCDR4 | OCDR3 | OCDR2 | OCDR1 | OCDR0 | 260 |
| $21 ($41) | WDTCR | – | – | – | WDCE | WDE | WDP2 | WDP1 | WDP0 | 57 |
| $20 ($40) | SFIOR | TSM | – | – | – | ACME | PUD | PSR0 | PSR321 | 84, 108, 141, 218 |
| $1F ($3F) | EEARH | – | – | – | – | EEPROM Address Register High | | | | 31 |
| $1E ($3E) | EEARL | EEPROM Address Register Low Byte | | | | | | | | 31 |
| $1D ($3D) | EEDR | EEPROM Data Register | | | | | | | | 31 |
| $1C ($3C) | EECR | – | – | – | – | EERIE | EEMWE | EEWE | EERE | 31 |
| $1B ($3B) | PORTA | PORTA7 | PORTA6 | PORTA5 | PORTA4 | PORTA3 | PORTA2 | PORTA1 | PORTA0 | 84 |
| $1A ($3A) | DDRA | DDA7 | DDA6 | DDA5 | DDA4 | DDA3 | DDA2 | DDA1 | DDA0 | 84 |
| $19 ($39) | PINA | PINA7 | PINA6 | PINA5 | PINA4 | PINA3 | PINA2 | PINA1 | PINA0 | 84 |
| $18 ($38) | PORTB | PORTB7 | PORTB6 | PORTB5 | PORTB4 | PORTB3 | PORTB2 | PORTB1 | PORTB0 | 85 |
| $17 ($37) | DDRB | DDB7 | DDB6 | DDB5 | DDB4 | DDB3 | DDB2 | DDB1 | DDB0 | 85 |
| $16 ($36) | PINB | PINB7 | PINB6 | PINB5 | PINB4 | PINB3 | PINB2 | PINB1 | PINB0 | 85 |
| $15 ($35) | PORTC | PORTC7 | PORTC6 | PORTC5 | PORTC4 | PORTC3 | PORTC2 | PORTC1 | PORTC0 | 85 |
| $14 ($34) | DDRC | DDC7 | DDC6 | DDC5 | DDC4 | DDC3 | DDC2 | DDC1 | DDC0 | 85 |
| $13 ($33) | PINC | PINC7 | PINC6 | PINC5 | PINC4 | PINC3 | PINC2 | PINC1 | PINC0 | 85 |
| $12 ($32) | PORTD | PORTD7 | PORTD6 | PORTD5 | PORTD4 | PORTD3 | PORTD2 | PORTD1 | PORTD0 | 85 |
| $11 ($31) | DDRD | DDD7 | DDD6 | DDD5 | DDD4 | DDD3 | DDD2 | DDD1 | DDD0 | 86 |
| $10 ($30) | PIND | PIND7 | PIND6 | PIND5 | PIND4 | PIND3 | PIND2 | PIND1 | PIND0 | 86 |
| $0F ($2F) | SPDR | SPI Data Register | | | | | | | | 166 |
| $0E ($2E) | SPSR | SPIF | WCOL | – | – | – | – | – | SPI2X | 165 |
| $0D ($2D) | SPCR | SPIE | SPE | DORD | MSTR | CPOL | CPHA | SPR1 | SPR0 | 164 |
| $0C ($2C) | UDR0 | USART0 I/O Data Register | | | | | | | | 184 |
| $0B ($2B) | UCSR0A | RXC0 | TXC0 | UDRE0 | FE0 | DOR0 | UPE0 | U2X0 | MPCM0 | 184 |
| $0A ($2A) | UCSR0B | RXCIE0 | TXCIE0 | UDRIE0 | RXEN0 | TXEN0 | UCSZ02 | RXB80 | TXB80 | 185 |
| $09 ($29) | UBRR0L | USART0 Baud Rate Register Low | | | | | | | | 188 |
| $08 ($28) | ACSR | ACD | ACBG | ACO | ACI | ACIE | ACIC | ACIS1 | ACIS0 | 218 |
| $07 ($27) | ADMUX | REFS1 | REFS0 | ADLAR | MUX4 | MUX3 | MUX2 | MUX1 | MUX0 | 232 |
| $06 ($26) | ADCSRA | ADEN | ADSC | ADFR | ADIF | ADIE | ADPS2 | ADPS1 | ADPS0 | 234 |
| $05 ($25) | ADCH | ADC Data Register High Byte | | | | | | | | 235 |
| $04 ($24) | ADCL | ADC Data Register Low byte | | | | | | | | 235 |
| $03 ($23) | PORTE | PORTE7 | PORTE6 | PORTE5 | PORTE4 | PORTE3 | PORTE2 | PORTE1 | PORTE0 | 86 |
| $02 ($22) | DDRE | DDE7 | DDE6 | DDE5 | DDE4 | DDE3 | DDE2 | DDE1 | DDE0 | 86 |
| $01 ($21) | PINE | PINE7 | PINE6 | PINE5 | PINE4 | PINE3 | PINE2 | PINE1 | PINE0 | 86 |
| $00 ($20) | PINF | PINF7 | PINF6 | PINF5 | PINF4 | PINF3 | PINF2 | PINF1 | PINF0 | 86 |

Notes:
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. Some of the status flags are cleared by writing a logical one to them. Note that the CBI and SBI instructions will operate on all bits in the I/O register, writing a one back into any flag read as set, thus clearing the flag. The CBI and SBI instructions work with registers $00 to $1F only.

# 7. Instruction Set Summary

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

# 7. Instruction Set Summary (Continued)

| Mnemonics | Operands | Description | Operation | Flags | #Clocks |
|---|---|---|---|---|---|
| BRIE | k | Branch if Interrupt Enabled | if ( I = 1) then PC ← PC + k + 1 | None | 1 / 2 |
| BRID | k | Branch if Interrupt Disabled | if ( I = 0) then PC ← PC + k + 1 | None | 1 / 2 |
| **DATA TRANSFER INSTRUCTIONS** | | | | | |
| MOV | Rd, Rr | Move Between Registers | Rd ← Rr | None | 1 |
| MOVW | Rd, Rr | Copy Register Word | Rd+1:Rd ← Rr+1:Rr | None | 1 |
| LDI | Rd, K | Load Immediate | Rd ← K | None | 1 |
| LD | Rd, X | Load Indirect | Rd ← (X) | None | 2 |
| LD | Rd, X+ | Load Indirect and Post-Inc. | Rd ← (X), X ← X + 1 | None | 2 |
| LD | Rd, - X | Load Indirect and Pre-Dec. | X ← X - 1, Rd ← (X) | None | 2 |
| LD | Rd, Y | Load Indirect | Rd ← (Y) | None | 2 |
| LD | Rd, Y+ | Load Indirect and Post-Inc. | Rd ← (Y), Y ← Y + 1 | None | 2 |
| LD | Rd, - Y | Load Indirect and Pre-Dec. | Y ← Y - 1, Rd ← (Y) | None | 2 |
| LDD | Rd,Y+q | Load Indirect with Displacement | Rd ← (Y + q) | None | 2 |
| LD | Rd, Z | Load Indirect | Rd ← (Z) | None | 2 |
| LD | Rd, Z+ | Load Indirect and Post-Inc. | Rd ← (Z), Z ← Z+1 | None | 2 |
| LD | Rd, -Z | Load Indirect and Pre-Dec. | Z ← Z - 1, Rd ← (Z) | None | 2 |
| LDD | Rd, Z+q | Load Indirect with Displacement | Rd ← (Z + q) | None | 2 |
| LDS | Rd, k | Load Direct from SRAM | Rd ← (k) | None | 2 |
| ST | X, Rr | Store Indirect | (X) ← Rr | None | 2 |
| ST | X+, Rr | Store Indirect and Post-Inc. | (X) ← Rr, X ← X + 1 | None | 2 |
| ST | - X, Rr | Store Indirect and Pre-Dec. | X ← X - 1, (X) ← Rr | None | 2 |
| ST | Y, Rr | Store Indirect | (Y) ← Rr | None | 2 |
| ST | Y+, Rr | Store Indirect and Post-Inc. | (Y) ← Rr, Y ← Y + 1 | None | 2 |
| ST | - Y, Rr | Store Indirect and Pre-Dec. | Y ← Y - 1, (Y) ← Rr | None | 2 |
| STD | Y+q,Rr | Store Indirect with Displacement | (Y + q) ← Rr | None | 2 |
| ST | Z, Rr | Store Indirect | (Z) ← Rr | None | 2 |
| ST | Z+, Rr | Store Indirect and Post-Inc. | (Z) ← Rr, Z ← Z + 1 | None | 2 |
| ST | -Z, Rr | Store Indirect and Pre-Dec. | Z ← Z - 1, (Z) ← Rr | None | 2 |
| STD | Z+q,Rr | Store Indirect with Displacement | (Z + q) ← Rr | None | 2 |
| STS | k, Rr | Store Direct to SRAM | (k) ← Rr | None | 2 |
| LPM | | Load Program Memory | R0 ← (Z) | None | 3 |
| LPM | Rd, Z | Load Program Memory | Rd ← (Z) | None | 3 |
| LPM | Rd, Z+ | Load Program Memory and Post-Inc | Rd ← (Z), Z ← Z+1 | None | 3 |
| ELPM | | Extended Load Program Memory | R0 ← (RAMPZ:Z) | None | 3 |
| ELPM | Rd, Z | Extended Load Program Memory | Rd ← (RAMPZ:Z) | None | 3 |
| ELPM | Rd, Z+ | Extended Load Program Memory and Post-Inc | Rd ← (RAMPZ:Z), RAMPZ:Z ← RAMPZ:Z+1 | None | 3 |
| SPM | | Store Program Memory | (Z) ← R1:R0 | None | - |
| IN | Rd, P | In Port | Rd ← P | None | 1 |
| OUT | P, Rr | Out Port | P ← Rr | None | 1 |
| PUSH | Rr | Push Register on Stack | STACK ← Rr | None | 2 |
| POP | Rd | Pop Register from Stack | Rd ← STACK | None | 2 |
| **BIT AND BIT-TEST INSTRUCTIONS** | | | | | |
| SBI | P,b | Set Bit in I/O Register | I/O(P,b) ← 1 | None | 2 |
| CBI | P,b | Clear Bit in I/O Register | I/O(P,b) ← 0 | None | 2 |
| LSL | Rd | Logical Shift Left | Rd(n+1) ← Rd(n), Rd(0) ← 0 | Z,C,N,V | 1 |
| LSR | Rd | Logical Shift Right | Rd(n) ← Rd(n+1), Rd(7) ← 0 | Z,C,N,V | 1 |
| ROL | Rd | Rotate Left Through Carry | Rd(0)←C,Rd(n+1)← Rd(n),C←Rd(7) | Z,C,N,V | 1 |
| ROR | Rd | Rotate Right Through Carry | Rd(7)←C,Rd(n)← Rd(n+1),C←Rd(0) | Z,C,N,V | 1 |
| ASR | Rd | Arithmetic Shift Right | Rd(n) ← Rd(n+1), n=0:6 | Z,C,N,V | 1 |
| SWAP | Rd | Swap Nibbles | Rd(3:0)←Rd(7:4),Rd(7:4)←Rd(3:0) | None | 1 |
| BSET | s | Flag Set | SREG(s) ← 1 | SREG(s) | 1 |
| BCLR | s | Flag Clear | SREG(s) ← 0 | SREG(s) | 1 |
| BST | Rr, b | Bit Store from Register to T | T ← Rr(b) | T | 1 |
| BLD | Rd, b | Bit load from T to Register | Rd(b) ← T | None | 1 |
| SEC | | Set Carry | C ← 1 | C | 1 |
| CLC | | Clear Carry | C ← 0 | C | 1 |
| SEN | | Set Negative Flag | N ← 1 | N | 1 |
| CLN | | Clear Negative Flag | N ← 0 | N | 1 |
| SEZ | | Set Zero Flag | Z ← 1 | Z | 1 |
| CLZ | | Clear Zero Flag | Z ← 0 | Z | 1 |
| SEI | | Global Interrupt Enable | I ← 1 | I | 1 |
| CLI | | Global Interrupt Disable | I ← 0 | I | 1 |
| SES | | Set Signed Test Flag | S ← 1 | S | 1 |
| CLS | | Clear Signed Test Flag | S ← 0 | S | 1 |
| **Mnemonics** | **Operands** | **Description** | **Operation** | **Flags** | **#Clocks** |
| SEV | | Set Twos Complement Overflow. | V ← 1 | V | 1 |

# 7. Instruction Set Summary (Continued)

| | | | | | |
|---|---|---|---|---|---|
| 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 |
| **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 |

# 8. Ordering Information

## 8.1 ATmega128A

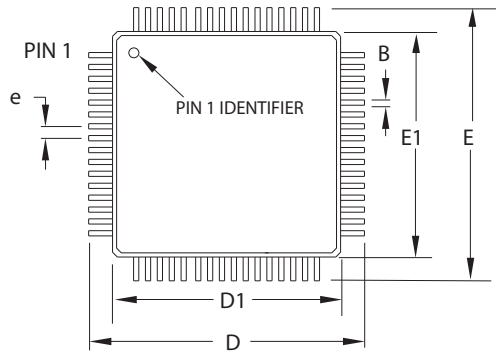
| Speed (MHz) | Power Supply | Ordering Code[2] | Package[1] | Operation Range |
|---|---|---|---|---|
| 16 | 2.7V - 5.5V | ATmega128A-AU<br>ATmega128A-AUR[3]<br>ATmega128A-MU<br>ATmega128A-MUR[3] | 64A<br>64A<br>64M1<br>64M1 | Industrial<br>(-40°C to 85°C) |

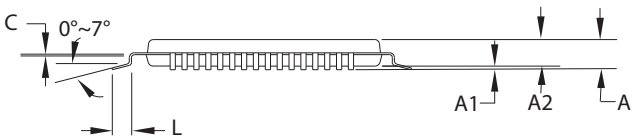
Notes:
1. The device can also be supplied in wafer form. Please contact your local Atmel sales office for detailed ordering information and minimum quantities.
2. Pb-free packaging complies to the European Directive for Restriction of Hazardous Substances (RoHS directive). Also Halide free and fully Green.
3. Tape & Reel

| Package Type | |
|---|---|
| **64A** | 64-lead, 14 × 14 × 1.0 mm, Thin Profile Plastic Quad Flat Package (TQFP) |
| **64M1** | 64-pad, 9 × 9 × 1.0 mm, Quad Flat No-Lead/Micro Lead Frame Package (QFN/MLF) |

# 9. Packaging Information

## 9.1 64A




COMMON DIMENSIONS
(Unit of measure = mm)

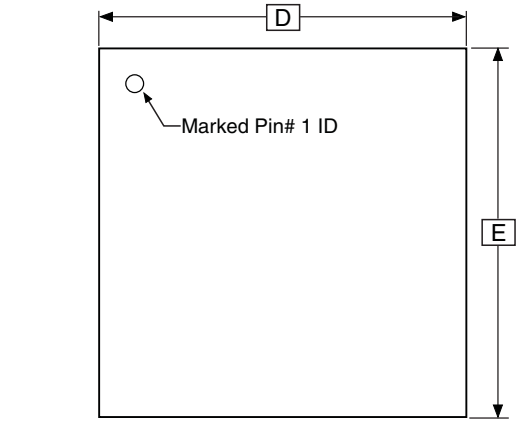
| SYMBOL | MIN | NOM | MAX | NOTE |
|---|---|---|---|---|
| A | – | – | 1.20 | |
| A1 | 0.05 | – | 0.15 | |
| A2 | 0.95 | 1.00 | 1.05 | |
| D | 15.75 | 16.00 | 16.25 | |
| D1 | 13.90 | 14.00 | 14.10 | Note 2 |
| E | 15.75 | 16.00 | 16.25 | |
| E1 | 13.90 | 14.00 | 14.10 | Note 2 |
| B | 0.30– | 0.45 | | |
| 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 0.25mm per side. Dimensions D1 and E1 are maximum plastic body size dimensions including mold mismatch.
3. Lead coplanarity is 0.10mm maximum.

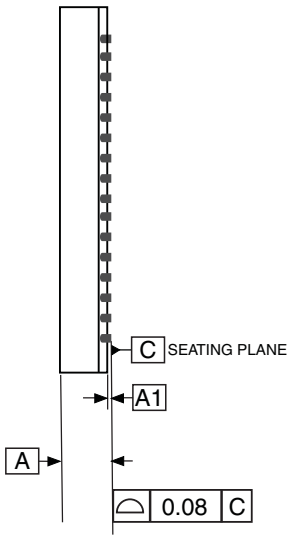
2010-10-20

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

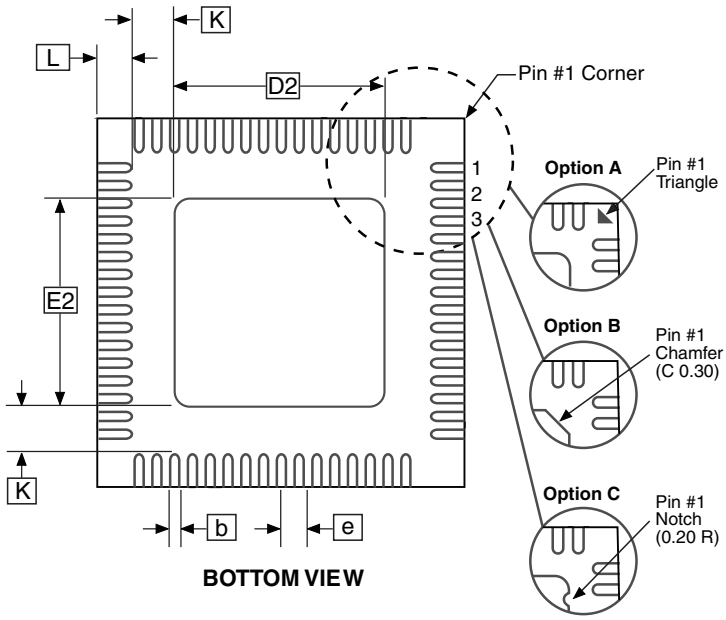
## 9.2 64M1




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 | |
| 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 | |

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

2010-10-19

| Atmel 2325 Orchard Parkway San Jose, CA 95131 | TITLE | DRAWING NO. | REV. |
|---|---|---|---|
| | **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) | 64M1 | H |

# 10. Errata

The revision letter in this section refers to the revision of the ATmega128A device.

## 10.1 ATmega128A Rev. U

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

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

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

**Problem Fix/Workaround**

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

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

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

**3. Stabilizing time needed when changing XDIV Register**

After increasing the source clock frequency more than 2% with settings in the XDIV register, the device may execute some of the subsequent instructions incorrectly.

**Problem Fix / Workaround**

The NOP instruction will always be executed correctly also right after a frequency change. Thus, the next 8 instructions after the change should be NOP instructions. To ensure this, follow this procedure:

1.Clear the I bit in the SREG Register.

2.Set the new pre-scaling factor in XDIV register.

3.Execute 8 NOP instructions

4.Set the I bit in SREG

This will ensure that all subsequent instructions will execute correctly.

Assembly Code Example:

```
CLI               ; clear global interrupt enable
OUT  XDIV, temp   ; set new prescale value
NOP               ; no operation
NOP               ; no operation
NOP               ; no operation
NOP               ; no operation
NOP               ; no operation
NOP               ; no operation
NOP               ; no operation
```

```
NOP                 ; no operation
SEI                 ; set global interrupt enable
```

4. **Stabilizing time needed when changing OSCCAL Register**

   After increasing the source clock frequency more than 2% with settings in the OSCCAL register, the device may execute some of the subsequent instructions incorrectly.

   **Problem Fix / Workaround**

   The behavior follows errata number 3., and the same Fix / Workaround is applicable on this errata.

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

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

   **Problem Fix / Workaround**

   - If ATmega128A is the only device in the scan chain, the problem is not visible.
   - Select the Device ID Register of the ATmega128A by issuing the IDCODE instruction or by entering the Test-Logic-Reset state of the TAP controller to read out the contents of its Device ID Register and possibly data from succeeding devices of the scan chain. Issue the BYPASS instruction to the ATmega128A while reading the Device ID Registers of preceding devices of the boundary scan chain.
   - If the Device IDs of all devices in the boundary scan chain must be captured simultaneously, the ATmega128A must be the fist device in the chain.

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

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

   **Problem Fix / Workaround**

   Always use OUT or SBI to set EERE in EECR.

# 11. 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.

## 11.1 Rev. 8151I – 08/2014

1. Updated with new template from 2014_050
2. Added values for 2.7V BOD levels in Table 27-3 on page 306.

## 11.2 Rev. 8151H – 02/11

1. Editing update according to the Atmel new style guide. No more space between the numbers and their units.
2. Updated the last page.

## 11.3 Rev. 8151G – 07/10

1. Updated the table note of Table 27-3 on page 306. The test is performed using BODLEVEL=0 and BODLEVEL=1

## 11.4 Rev. 8151F – 06/10

1. Inserted cross reference in “Minimizing Power Consumption” on page 48
2. Updated Technical Terminology according to Atmel standard
3. Note 6 and Note 7 below “Two-wire Serial Bus Requirements” on page 307 have been removed
4. The text in “Bit 6 – TXCIEn: TX Complete Interrupt Enable” on page 185 has been corrected by adding an “n”

## 11.5 Rev. 8151E – 02/10

1. Updated “Receiving Frames with 9 Data Bits” on page 177. The C code updated.
2. Updated “Packaging Information” on page 17.
3. Updated “Performing Page Erase by SPM” on page 267.

## 11.6 Rev. 8151D – 07/09

1. Updated “Errata” on page 19.
2. Updated the last page with Atmel’s new addresses.

## 11.7 Rev. 8151C – 05/09

1. Updated “Errata” on page 19. ATmega128A Rev. U.

## 11.8 Rev. 8151B – 03/09

1. Updated view of “Typical Characteristics” on page 320.
2. Editorial updates.

## 11.9 Rev. 8151A – 08/08

1. Initial revision. (Based on the ATmega128/L datasheet 2467R-AVR-06/08)

   Changes done compared to the ATmega128/L datasheet 2467R-AVR-06/08:

   - Updated “Stack Pointer” on page 14 description.
   - “Power Management and Sleep Modes” on page 46 is reorganized.
   - All Electrical characteristics is moved to “Electrical Characteristics” on page 303.
   - Output Low Voltage ($V_{OL}$) and Reset Pull-up Resistor ($R_{RST}$) limits updated in “DC Characteristics” on page 303.
   - Register descriptions are moved to sub sections at the end of each chapter.
   - New graphics in “Typical Characteristics” on page 320.
   - New “Ordering Information” on page 16.


**Atmel Corporation** 1600 Technology Drive, San Jose, CA 95110 USA **T:** (+1)(408) 441.0311 **F:** (+1)(408) 436.4200 | **www.atmel.com**