Features

- High-performance, Low-power AVR® 8-bit Microcontroller
- Advanced RISC Architecture
  - 133 Powerful Instructions – Most Single Clock Cycle Execution
  - 32 x 8 General Purpose Working Registers + Peripheral Control Registers
  - Fully Static Operation
  - Up to 16 MIPS Throughput at 16 MHz
  - On-chip 2-cycle Multiplier
- Nonvolatile Program and Data Memories
  - 128K Bytes of In-System Reprogrammable Flash
    Endurance: 10,000 Write/Erase Cycles
  - Optional Boot Code Section with Independent Lock Bits
    In-System Programming by On-chip Boot Program
    True Read-While-Write Operation
  - 4K Bytes EEPROM
    Endurance: 100,000 Write/Erase Cycles
  - 4K Bytes Internal SRAM
  - Up to 64K Bytes 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
  - 2.7 - 5.5V for ATmega128L
  - 4.5 - 5.5V for ATmega128
- Speed Grades
  - 0 - 8 MHz for ATmega128L
  - 0 - 16 MHz for ATmega128

8-bit AVR® Microcontroller with 128K Bytes In-System Programmable Flash

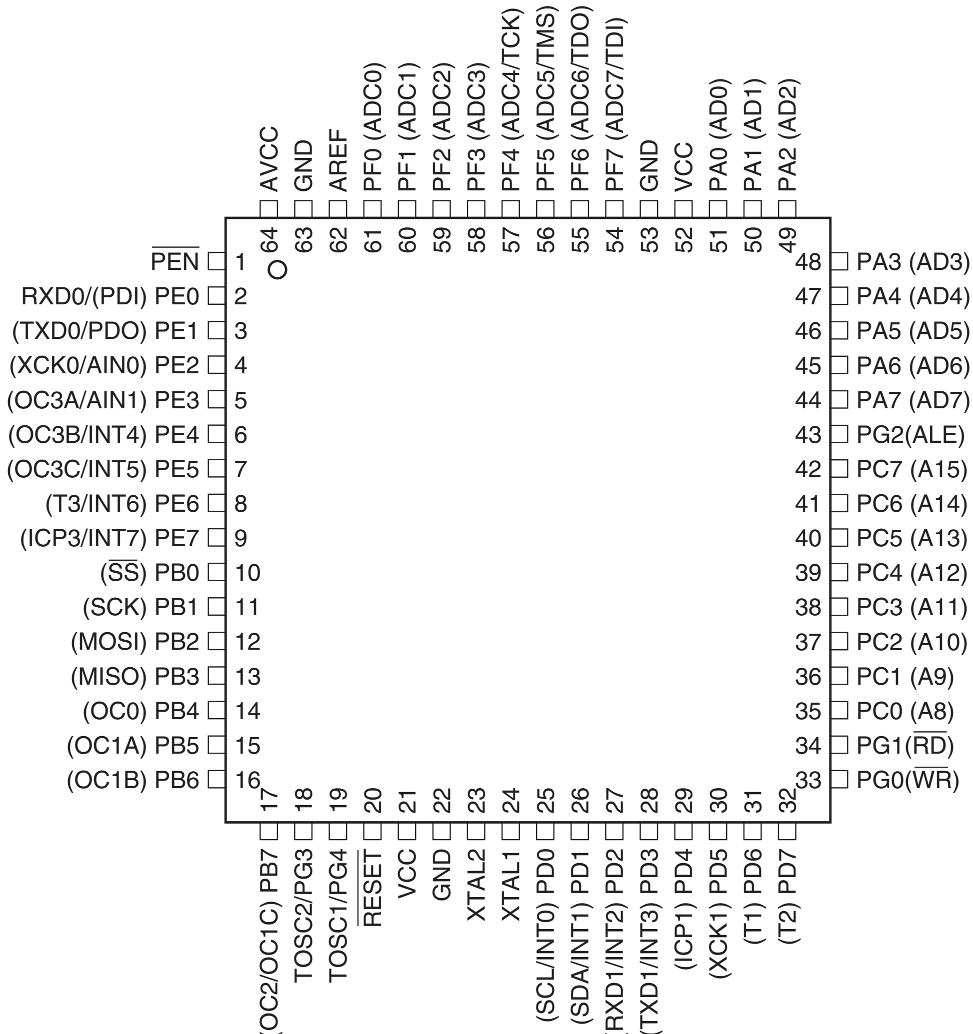
ATmega128
ATmega128L

Summary

Rev. 2467NS–AVR–03/06

Pin Configurations

Figure 1. Pinout ATmega128



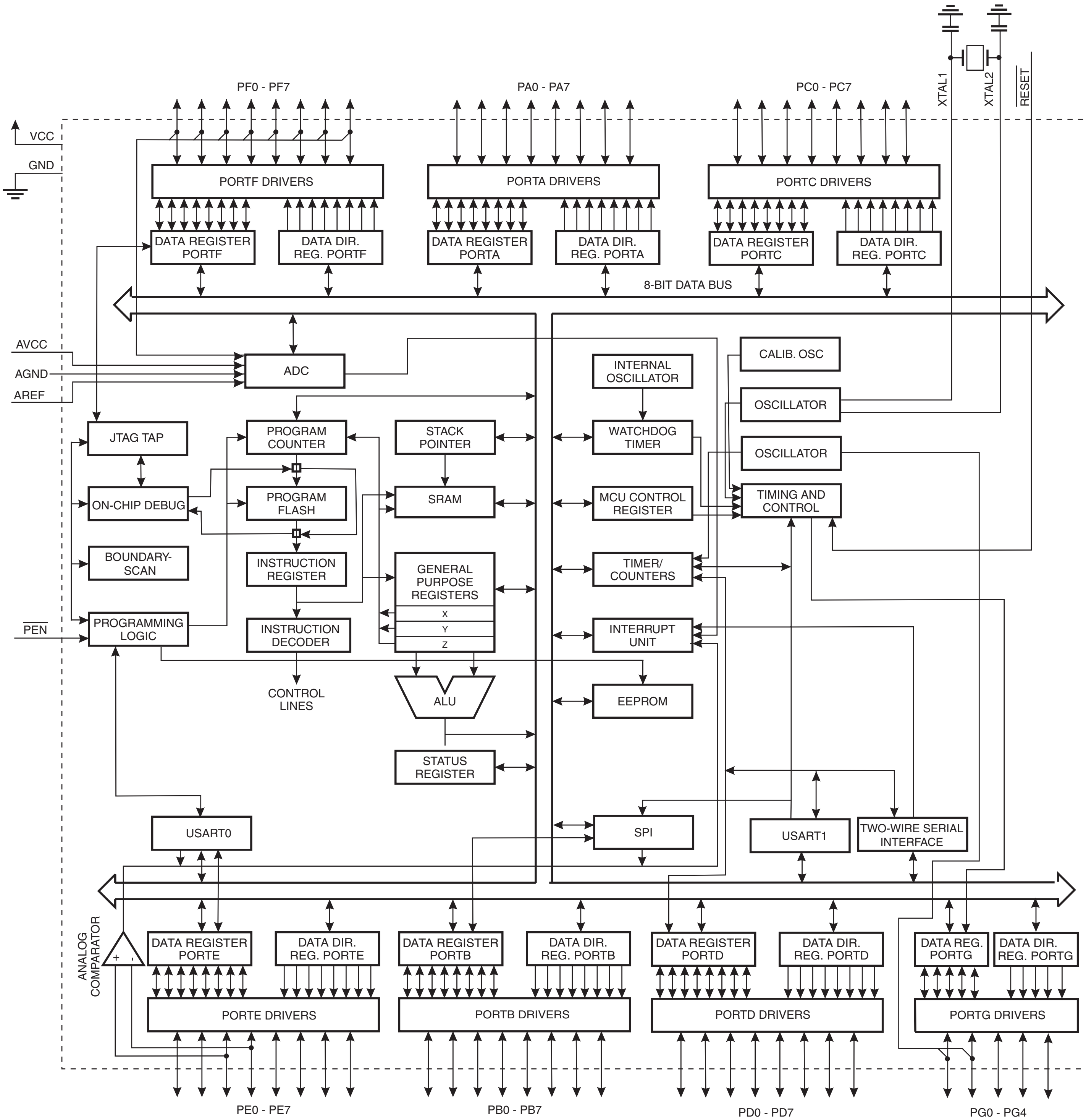
Note: The Pinout figure applies to both TQFP and MLF packages. The bottom pad under the QFN/MLF package should be soldered to ground.

Overview

The ATmega128 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 ATmega128 achieves throughputs approaching 1 MIPS per MHz allowing the system designer to optimize power consumption versus processing speed.

Block Diagram

Figure 2. Block Diagram



The AVR core combines a rich instruction set with 32 general purpose working registers. All the 32 registers are directly connected to the Arithmetic Logic Unit (ALU), allowing two independent registers to be accessed in one single instruction executed in one clock cycle. The resulting architecture is more code efficient while achieving throughputs up to ten times faster than conventional CISC microcontrollers.

The ATmega128 provides the following features: 128K bytes of In-System Programmable Flash with Read-While-Write capabilities, 4K bytes EEPROM, 4K bytes 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 ATmega128 is a powerful microcontroller that provides a highly flexible and cost effective solution to many embedded control applications.

The ATmega128 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.

ATmega103 and ATmega128 Compatibility

The ATmega128 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 ATmega128. Most additional I/O locations are added in an Extended I/O space starting from $60 to $FF, (i.e., 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 ATmega128 is 100% pin compatible with ATmega103, and can replace the ATmega103 on current Printed Circuit Boards. The application note "Replacing ATmega103 by ATmega128" describes what the user should be aware of replacing the ATmega103 by an ATmega128.

ATmega103 Compatibility Mode

By programming the M103C fuse, the ATmega128 will be compatible with the ATmega103 regards to RAM, I/O pins and interrupt vectors as described above. However, some new features in ATmega128 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 ATmega128.

Pin Descriptions

VCC

Digital supply voltage.

GND

Ground.

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 ATmega128 as listed on page 72.

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 ATmega128 as listed on page 73.

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 ATmega128 as listed on page 76. 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 ATmega128 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.

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 ATmega128 as listed on page 77.

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 ATmega128 as listed on page 80.

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.

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 ATmega103 compatibility mode, these pins only serves as strobes signals to the external memory as well as input to the 32 kHz 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.

$\overline{\text{RESET}}$

Reset input. A low level on this pin for longer than the minimum pulse length will generate a reset, even if the clock is not running. The minimum pulse length is given in Table 19 on page 50. Shorter pulses are not guaranteed to generate a reset.

XTAL1

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

XTAL2

Output from the inverting Oscillator amplifier.

AVCC

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

AREF

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

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.

Resources

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

Register Summary

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page
($FF)	Reserved	–	–	–	–	–	–	–	–	
..	Reserved	–	–	–	–	–	–	–	–	
($9E)	Reserved	–	–	–	–	–	–	–	–	
($9D)	UCSR1C	–	UMSEL1	UPM11	UPM10	USBS1	UCSZ11	UCSZ10	UCPOL1	192
($9C)	UDR1	USART1 I/O Data Register								190
($9B)	UCSR1A	RXC1	TXC1	UDRE1	FE1	DOR1	UPE1	U2X1	MPCM1	190
($9A)	UCSR1B	RXCIE1	TXCIE1	UDRIE1	RXEN1	TXEN1	UCSZ12	RXB81	TXB81	191
($99)	UBRR1L	USART1 Baud Rate Register Low								194
($98)	UBRR1H	–	–	–	–	USART1 Baud Rate Register High				194
($97)	Reserved	–	–	–	–	–	–	–	–	
($96)	Reserved	–	–	–	–	–	–	–	–	
($95)	UCSR0C	–	UMSEL0	UPM01	UPM00	USBS0	UCSZ01	UCSZ00	UCPOL0	192
($94)	Reserved	–	–	–	–	–	–	–	–	
($93)	Reserved	–	–	–	–	–	–	–	–	
($92)	Reserved	–	–	–	–	–	–	–	–	
($91)	Reserved	–	–	–	–	–	–	–	–	
($90)	UBRR0H	–	–	–	–	USART0 Baud Rate Register High				194
($8F)	Reserved	–	–	–	–	–	–	–	–	
($8E)	Reserved	–	–	–	–	–	–	–	–	
($8D)	Reserved	–	–	–	–	–	–	–	–	
($8C)	TCCR3C	FOC3A	FOC3B	FOC3C	–	–	–	–	–	137
($8B)	TCCR3A	COM3A1	COM3A0	COM3B1	COM3B0	COM3C1	COM3C0	WGM31	WGM30	133
($8A)	TCCR3B	ICNC3	ICES3	–	WGM33	WGM32	CS32	CS31	CS30	136
($89)	TCNT3H	Timer/Counter3 – Counter Register High Byte								138
($88)	TCNT3L	Timer/Counter3 – Counter Register Low Byte								138
($87)	OCR3AH	Timer/Counter3 – Output Compare Register A High Byte								138
($86)	OCR3AL	Timer/Counter3 – Output Compare Register A Low Byte								138
($85)	OCR3BH	Timer/Counter3 – Output Compare Register B High Byte								139
($84)	OCR3BL	Timer/Counter3 – Output Compare Register B Low Byte								139
($83)	OCR3CH	Timer/Counter3 – Output Compare Register C High Byte								139
($82)	OCR3CL	Timer/Counter3 – Output Compare Register C Low Byte								139
($81)	ICR3H	Timer/Counter3 – Input Capture Register High Byte								139
($80)	ICR3L	Timer/Counter3 – Input Capture Register Low Byte								139
($7F)	Reserved	–	–	–	–	–	–	–	–	
($7E)	Reserved	–	–	–	–	–	–	–	–	
($7D)	ETIMSK	–	–	TICIE3	OCIE3A	OCIE3B	TOIE3	OCIE3C	OCIE1C	140
($7C)	ETIFR	–	–	ICF3	OCF3A	OCF3B	TOV3	OCF3C	OCF1C	141
($7B)	Reserved	–	–	–	–	–	–	–	–	
($7A)	TCCR1C	FOC1A	FOC1B	FOC1C	–	–	–	–	–	137
($79)	OCR1CH	Timer/Counter1 – Output Compare Register C High Byte								138
($78)	OCR1CL	Timer/Counter1 – Output Compare Register C Low Byte								138
($77)	Reserved	–	–	–	–	–	–	–	–	
($76)	Reserved	–	–	–	–	–	–	–	–	
($75)	Reserved	–	–	–	–	–	–	–	–	
($74)	TWCR	TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	–	TWIE	207
($73)	TWDR	Two-wire Serial Interface Data Register								209
($72)	TWAR	TWA6	TWA5	TWA4	TWA3	TWA2	TWA1	TWA0	TWGCE	209
($71)	TWSR	TWS7	TWS6	TWS5	TWS4	TWS3	–	TWPS1	TWPS0	208
($70)	TWBR	Two-wire Serial Interface Bit Rate Register								207
($6F)	OSCCAL	Oscillator Calibration Register								41
($6E)	Reserved	–	–	–	–	–	–	–	–	
($6D)	XMCRA	–	SRL2	SRL1	SRL0	SRW01	SRW00	SRW11		31
($6C)	XMCRB	XMBK	–	–	–	–	XMM2	XMM1	XMM0	33
($6B)	Reserved	–	–	–	–	–	–	–	–	
($6A)	EICRA	ISC31	ISC30	ISC21	ISC20	ISC11	ISC10	ISC01	ISC00	89
($69)	Reserved	–	–	–	–	–	–	–	–	
($68)	SPMCSR	SPMIE	RWWSB	–	RWWSRE	BLBSET	PGWRT	PGERS	SPMEN	280
($67)	Reserved	–	–	–	–	–	–	–	–	
($66)	Reserved	–	–	–	–	–	–	–	–	
($65)	PORTG	–	–	–	PORTG4	PORTG3	PORTG2	PORTG1	PORTG0	88
($64)	DDRG	–	–	–	DDG4	DDG3	DDG2	DDG1	DDG0	88
($63)	PING	–	–	–	PING4	PING3	PING2	PING1	PING0	88
($62)	PORTF	PORTF7	PORTF6	PORTF5	PORTF4	PORTF3	PORTF2	PORTF1	PORTF0	87

Register Summary (Continued)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page
($61)	DDRF	DDF7	DDF6	DDF5	DDF4	DDF3	DDF2	DDF1	DDF0	88
($60)	Reserved	–	–	–	–	–	–	–	–	
$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	43
$3B ($5B)	RAMPZ	–	–	–	–	–	–	–	RAMPZ0	14
$3A ($5A)	EICRB	ISC71	ISC70	ISC61	ISC60	ISC51	ISC50	ISC41	ISC40	90
$39 ($59)	EIMSK	INT7	INT6	INT5	INT4	INT3	INT2	INT1	INT0	91
$38 ($58)	EIFR	INTF7	INTF6	INTF5	INTF4	INTF3	INTF	INTF1	INTF0	91
$37 ($57)	TIMSK	OCIE2	TOIE2	TICIE1	OCIE1A	OCIE1B	TOIE1	OCIE0	TOIE0	108, 140, 160
$36 ($56)	TIFR	OCF2	TOV2	ICF1	OCF1A	OCF1B	TOV1	OCF0	TOV0	108, 141, 160
$35 ($55)	MCUCR	SRE	SRW10	SE	SM1	SM0	SM2	IVSEL	IVCE	31, 44, 63
$34 ($54)	MCUCSR	JTD	–	–	JTRF	WDRF	BORF	EXTRF	PORF	53, 257
$33 ($53)	TCCR0	FOC0	WGM00	COM01	COM00	WGM01	CS02	CS01	CS00	103
$32 ($52)	TCNT0	Timer/Counter0 (8 Bit)								105
$31 ($51)	OCR0	Timer/Counter0 Output Compare Register								105
$30 ($50)	ASSR	–	–	–	–	AS0	TCN0UB	OCR0UB	TCR0UB	106
$2F ($4F)	TCCR1A	COM1A1	COM1A0	COM1B1	COM1B0	COM1C1	COM1C0	WGM11	WGM10	133
$2E ($4E)	TCCR1B	ICNC1	ICES1	–	WGM13	WGM12	CS12	CS11	CS10	136
$2D ($4D)	TCNT1H	Timer/Counter1 – Counter Register High Byte								138
$2C ($4C)	TCNT1L	Timer/Counter1 – Counter Register Low Byte								138
$2B ($4B)	OCR1AH	Timer/Counter1 – Output Compare Register A High Byte								138
$2A ($4A)	OCR1AL	Timer/Counter1 – Output Compare Register A Low Byte								138
$29 ($49)	OCR1BH	Timer/Counter1 – Output Compare Register B High Byte								138
$28 ($48)	OCR1BL	Timer/Counter1 – Output Compare Register B Low Byte								138
$27 ($47)	ICR1H	Timer/Counter1 – Input Capture Register High Byte								139
$26 ($46)	ICR1L	Timer/Counter1 – Input Capture Register Low Byte								139
$25 ($45)	TCCR2	FOC2	WGM20	COM21	COM20	WGM21	CS22	CS21	CS20	158
$24 ($44)	TCNT2	Timer/Counter2 (8 Bit)								160
$23 ($43)	OCR2	Timer/Counter2 Output Compare Register								160
$22 ($42)	OCDR	IDRD/OCDR7	OCDR6	OCDR5	OCDR4	OCDR3	OCDR2	OCDR1	OCDR0	254
$21 ($41)	WDTCR	–	–	–	WDCE	WDE	WDP2	WDP1	WDP0	55
$20 ($40)	SFIOR	TSM	–	–	–	ACME	PUD	PSR0	PSR321	72, 109, 145, 229
$1F ($3F)	EEARH	–	–	–	–	EEPROM Address Register High				21
$1E ($3E)	EEARL	EEPROM Address Register Low Byte								21
$1D ($3D)	EEDR	EEPROM Data Register								22
$1C ($3C)	EECR	–	–	–	–	EERIE	EEMWE	EEWE	EERE	22
$1B ($3B)	PORTA	PORTA7	PORTA6	PORTA5	PORTA4	PORTA3	PORTA2	PORTA1	PORTA0	86
$1A ($3A)	DDRA	DDA7	DDA6	DDA5	DDA4	DDA3	DDA2	DDA1	DDA0	86
$19 ($39)	PINA	PINA7	PINA6	PINA5	PINA4	PINA3	PINA2	PINA1	PINA0	86
$18 ($38)	PORTB	PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0	86
$17 ($37)	DDRB	DDB7	DDB6	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0	86
$16 ($36)	PINB	PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0	86
$15 ($35)	PORTC	PORTC7	PORTC6	PORTC5	PORTC4	PORTC3	PORTC2	PORTC1	PORTC0	86
$14 ($34)	DDRC	DDC7	DDC6	DDC5	DDC4	DDC3	DDC2	DDC1	DDC0	86
$13 ($33)	PINC	PINC7	PINC6	PINC5	PINC4	PINC3	PINC2	PINC1	PINC0	87
$12 ($32)	PORTD	PORTD7	PORTD6	PORTD5	PORTD4	PORTD3	PORTD2	PORTD1	PORTD0	87
$11 ($31)	DDRD	DDD7	DDD6	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0	87
$10 ($30)	PIND	PIND7	PIND6	PIND5	PIND4	PIND3	PIND2	PIND1	PIND0	87
$0F ($2F)	SPDR	SPI Data Register								170
$0E ($2E)	SPSR	SPIF	WCOL	–	–	–	–	–	SPI2X	170
$0D ($2D)	SPCR	SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR0	168
$0C ($2C)	UDR0	USART0 I/O Data Register								190
$0B ($2B)	UCSR0A	RXC0	TXC0	UDRE0	FE0	DOR0	UPE0	U2X0	MPCM0	190
$0A ($2A)	UCSR0B	RXCIE0	TXCIE0	UDRIE0	RXEN0	TXEN0	UCSZ02	RXB80	TXB80	191
$09 ($29)	UBRR0L	USART0 Baud Rate Register Low								194
$08 ($28)	ACSR	ACD	ACBG	ACO	ACI	ACIE	ACIC	ACIS1	ACIS0	229
$07 ($27)	ADMUX	REFS1	REFS0	ADLAR	MUX4	MUX3	MUX2	MUX1	MUX0	245
$06 ($26)	ADCSRA	ADEN	ADSC	ADFR	ADIF	ADIE	ADPS2	ADPS1	ADPS0	246
$05 ($25)	ADCH	ADC Data Register High Byte								247
$04 ($24)	ADCL	ADC Data Register Low byte								247
$03 ($23)	PORTE	PORTE7	PORTE6	PORTE5	PORTE4	PORTE3	PORTE2	PORTE1	PORTE0	87
$02 ($22)	DDRE	DDE7	DDE6	DDE5	DDE4	DDE3	DDE2	DDE1	DDE0	87

Register Summary (Continued)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page
$01 ($21)	PINE	PINE7	PINE6	PINE5	PINE4	PINE3	PINE2	PINE1	PINE0	87
$00 ($20)	PINF	PINF7	PINF6	PINF5	PINF4	PINF3	PINF2	PINF1	PINF0	88

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.

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

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

Instruction Set Summary (Continued)

Mnemonics	Operands	Description	Operation	Flags	#Clocks
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
MCU CONTROL INSTRUCTIONS					
NOP		No Operation		None	1
SLEEP		Sleep	(see specific descr. for Sleep function)	None	1
WDR		Watchdog Reset	(see specific descr. for WDR/timer)	None	1
BREAK		Break	For On-chip Debug Only	None	N/A

Ordering Information

Speed (MHz)	Power Supply	Ordering Code	Package[1]	Operation Range
8	2.7 - 5.5V	ATmega128L-8AC ATmega128L-8MC	64A 64M1	Commercial (0°C to 70°C)
		ATmega128L-8AI ATmega128L-8AU[2] ATmega128L-8MI ATmega128L-8MU[2]	64A 64A 64M1 64M1	Industrial (-40°C to 85°C)
16	4.5 - 5.5V	ATmega128-16AC ATmega128-16MC	64A 64M1	Commercial (0°C to 70°C)
		ATmega128-16AI ATmega128-16AU[2] ATmega128-16MI ATmega128-16MU[2]	64A 64A 64M1 64M1	Industrial (-40°C to 85°C)

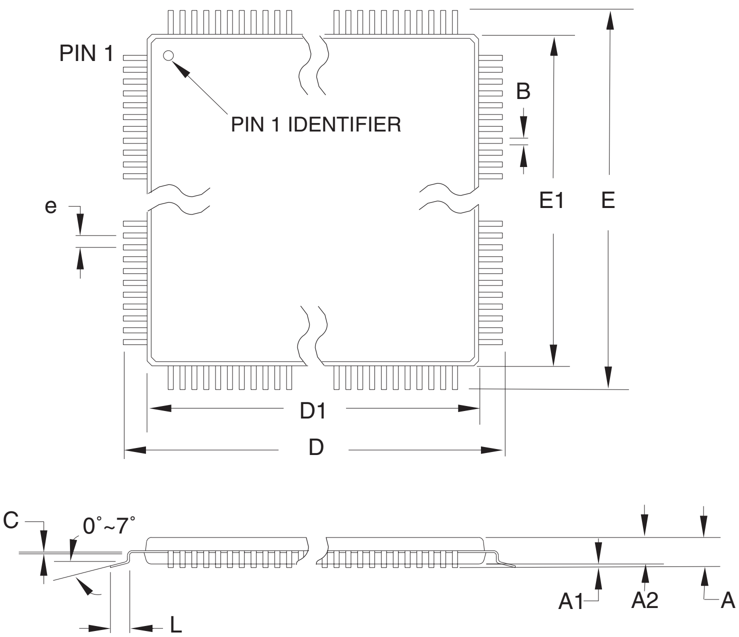
Notes: 1. The device can also be supplied in wafer form. Please contact your local Atmel sales office for detailed ordering information and minimum quantities.

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

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

Packaging Information

64A



COMMON DIMENSIONS
(Unit of Measure = mm)

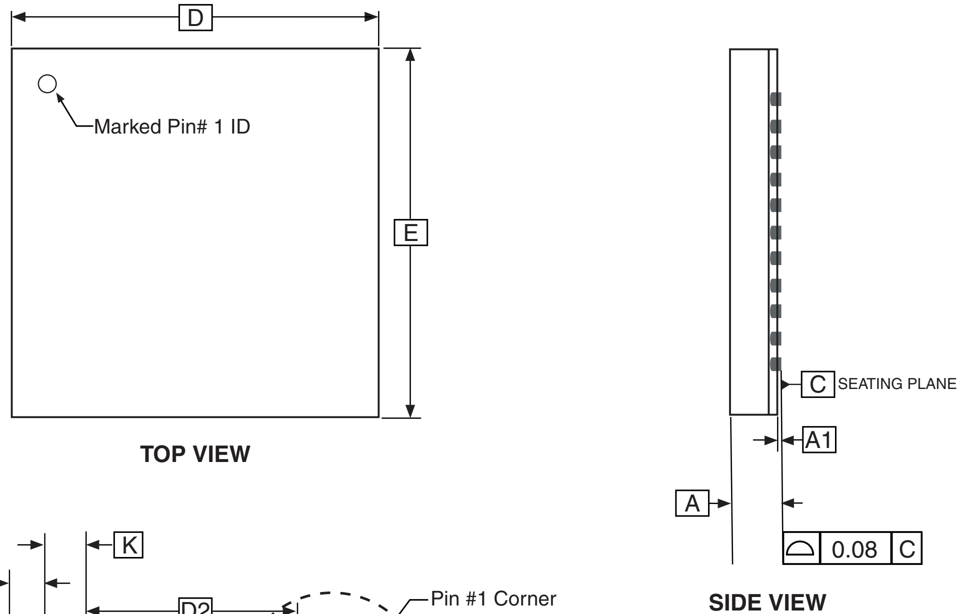
SYMBOL	MIN	NOM	MAX	NOTE
A	–	–	1.20	
A1	0.05	–	0.15	
A2	0.95	1.00	1.05	
D	15.75	16.00	16.25	
D1	13.90	14.00	14.10	Note 2
E	15.75	16.00	16.25	
E1	13.90	14.00	14.10	Note 2
B	0.30	–	0.45	
C	0.09	–	0.20	
L	0.45	–	0.75	
e	0.80 TYP			

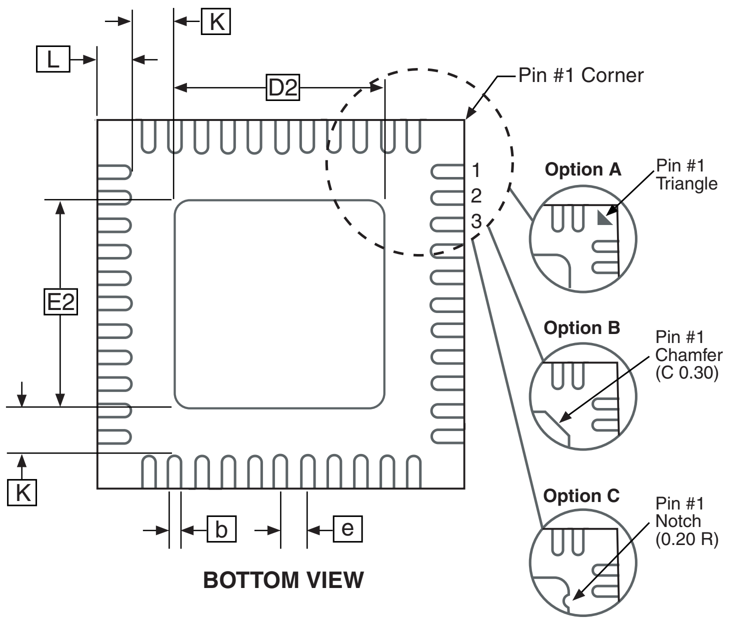
Notes:
1. This package conforms to JEDEC reference MS-026, Variation AEB.
2. Dimensions D1 and E1 do not include mold protrusion. Allowable protrusion is 0.25 mm per side. Dimensions D1 and E1 are maximum plastic body size dimensions including mold mismatch.
3. Lead coplanarity is 0.10 mm maximum.

10/5/2001

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

64M1





COMMON DIMENSIONS
(Unit of Measure = mm)

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

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

7/19/05

	TITLE	DRAWING NO.	REV.
ATMEL 2325 Orchard Parkway San Jose, CA 95131	**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	F

Errata

The revision letter in this section refers to the revision of the ATmega128 device.

ATmega128 Rev. I

- **Stabilizing time needed when changing XDIV Register**
- **Stabilizing time needed when changing OSCCAL Register**

1. **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                  ; clear global interrupt enable
```

2. **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 1., and the same Fix / Workaround is applicable on this errata.

A proposal for solving problems regarding the JTAG instruction IDCODE is presented below.

IDCODE masks data from TDI input

The public but optional JTAG instruction IDCODE is not implemented correctly according to IEEE1149.1; a logic one is scanned into the shift register instead of the TDI input while shifting the Device ID Register. Hence, captured data from the preceding devices in the boundary scan chain are lost and replaced by all-ones, and data to succeeding devices are replaced by all-ones during Update-DR.

If ATmega128 is the only device in the scan chain, the problem is not visible.

Problem Fix / Workaround

Select the Device ID Register of the ATmega128 (Either 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. Note that data to succeeding devices cannot be entered during this scan, but data to preceding devices can. Issue the BYPASS instruction to the ATmega128 to select its Bypass Register while reading the Device ID Registers of preceding devices of the boundary scan chain. Never read data from succeeding devices in the boundary scan chain or upload data to the succeeding devices while the Device ID Register is selected for the ATmega128. Note that the IDCODE instruction is the default instruction selected by the Test-Logic-Reset state of the TAP-controller.

Alternative Problem Fix / Workaround

If the Device IDs of all devices in the boundary scan chain must be captured simultaneously (for instance if blind interrogation is used), the boundary scan chain can be connected in such way that the ATmega128 is the fist device in the chain. Update-DR will still not work for the succeeding devices in the boundary scan chain as long as IDCODE is present in the JTAG Instruction Register, but the Device ID registered cannot be uploaded in any case.

ATmega128 Rev. H

- **Stabilizing time needed when changing XDIV Register**
- **Stabilizing time needed when changing OSCCAL Register**

1. 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               ; clear global interrupt enable
```

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

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

   Problem Fix / Workaround

   The behavior follows errata number 1., and the same Fix / Workaround is applicable on this errata.

A proposal for solving problems regarding the JTAG instruction IDCODE is presented below.

IDCODE masks data from TDI input

The public but optional JTAG instruction IDCODE is not implemented correctly according to IEEE1149.1; a logic one is scanned into the shift register instead of the TDI input while shifting the Device ID Register. Hence, captured data from the preceding devices in the boundary scan chain are lost and replaced by all-ones, and data to succeeding devices are replaced by all-ones during Update-DR.

If ATmega128 is the only device in the scan chain, the problem is not visible.

Problem Fix / Workaround

Select the Device ID Register of the ATmega128 (Either 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. Note that data to succeeding devices cannot be entered during this scan, but data to preceding devices can. Issue the BYPASS instruction to the ATmega128 to select its Bypass Register while reading the Device ID Registers of preceding devices of the boundary scan chain. Never read data from succeeding devices in the boundary scan chain or upload data to the succeeding devices while the Device ID Register is selected for the ATmega128. Note that the IDCODE instruction is the default instruction selected by the Test-Logic-Reset state of the TAP-controller.

Alternative Problem Fix / Workaround

If the Device IDs of all devices in the boundary scan chain must be captured simultaneously (for instance if blind interrogation is used), the boundary scan chain can be connected in such way that the ATmega128 is the fist device in the chain. Update-DR will still not work for the succeeding devices in the boundary scan chain as long as IDCODE is present in the JTAG Instruction Register, but the Device ID registered cannot be uploaded in any case.

ATmega128 Rev. G

- **Stabilizing time needed when changing XDIV Register**
- **Stabilizing time needed when changing OSCCAL Register**

1. **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
```

2. 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 1., and the same Fix / Workaround is applicable on this errata.

A proposal for solving problems regarding the JTAG instruction IDCODE is presented below.

IDCODE masks data from TDI input

The public but optional JTAG instruction IDCODE is not implemented correctly according to IEEE1149.1; a logic one is scanned into the shift register instead of the TDI input while shifting the Device ID Register. Hence, captured data from the preceding devices in the boundary scan chain are lost and replaced by all-ones, and data to succeeding devices are replaced by all-ones during Update-DR.

If ATmega128 is the only device in the scan chain, the problem is not visible.

Problem Fix / Workaround

Select the Device ID Register of the ATmega128 (Either 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. Note that data to succeeding devices cannot be entered during this scan, but data to preceding devices can. Issue the BYPASS instruction to the ATmega128 to select its Bypass Register while reading the Device ID Registers of preceding devices of the boundary scan chain. Never read data from succeeding devices in the boundary scan chain or upload data to the succeeding devices while the Device ID Register is selected for the ATmega128. Note that the IDCODE instruction is the default instruction selected by the Test-Logic-Reset state of the TAP-controller.

Alternative Problem Fix / Workaround

If the Device IDs of all devices in the boundary scan chain must be captured simultaneously (for instance if blind interrogation is used), the boundary scan chain can be connected in such way that the ATmega128 is the fist device in the chain. Update-DR will still not work for the succeeding devices in the boundary scan chain as long as IDCODE is present in the JTAG Instruction Register, but the Device ID registered cannot be uploaded in any case.

ATmega128 Rev. F

- **Stabilizing time needed when changing XDIV Register**
- **Stabilizing time needed when changing OSCCAL Register**

1. 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
```

2. 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 1., and the same Fix / Workaround is applicable on this errata.

A proposal for solving problems regarding the JTAG instruction IDCODE is presented below.

IDCODE masks data from TDI input

The public but optional JTAG instruction IDCODE is not implemented correctly according to IEEE1149.1; a logic one is scanned into the shift register instead of the TDI input while shifting the Device ID Register. Hence, captured data from the preceding devices in the boundary scan chain are lost and replaced by all-ones, and data to succeeding devices are replaced by all-ones during Update-DR.

If ATmega128 is the only device in the scan chain, the problem is not visible.

Problem Fix / Workaround

Select the Device ID Register of the ATmega128 (Either 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. Note that data to succeeding devices cannot be entered during this scan, but data to preceding devices can. Issue the BYPASS instruction to the ATmega128 to select its Bypass Register while reading the Device ID Registers of preceding devices of the boundary scan chain. Never read data from succeeding devices in the boundary scan chain or upload data to the succeeding devices while the Device ID Register is selected for the ATmega128. Note that the IDCODE instruction is the default instruction selected by the Test-Logic-Reset state of the TAP-controller.

Alternative Problem Fix / Workaround

If the Device IDs of all devices in the boundary scan chain must be captured simultaneously (for instance if blind interrogation is used), the boundary scan chain can be connected in such way that the ATmega128 is the fist device in the chain. Update-DR will still not work for the succeeding devices in the boundary scan chain as long as IDCODE is present in the JTAG Instruction Register, but the Device ID registered cannot be uploaded in any case.

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.

Changes from Rev. 2467M-11/04 to Rev. 2467N-03/06

1. Updated note for Figure 1 on page 2.
2. Updated "Alternate Functions of Port D" on page 77.
3. Updated "Alternate Functions of Port G" on page 84.
4. Updated "Phase Correct PWM Mode" on page 100.
5. Updated Table 59 on page 134, Table 60 on page 134.
6. Updated "Bit 2 – TOV3: Timer/Counter3, Overflow Flag" on page 142.
7. Updated "Serial Peripheral Interface – SPI" on page 164.
8. Updated Features in "Analog to Digital Converter" on page 232
9. Added note in "Input Channel and Gain Selections" on page 245.
10. Updated "Errata" on page 17.

Changes from Rev. 2467L-05/04 to Rev. 2467M-11/04

1. Removed "analog ground", replaced by "ground".
2. Updated Table 11 on page 40, Table 114 on page 288, Table 128 on page 307, and Table 132 on page 324. Updated Figure 114 on page 240.
3. Added note to "Port C (PC7..PC0)" on page 6.
4. Updated "Ordering Information" on page 14.

Changes from Rev. 2467K-03/04 to Rev. 2467L-05/04

1. Removed "Preliminary" and "TBD" from the datasheet, replaced occurrences of ICx with ICPx.
2. Updated Table 8 on page 38, Table 19 on page 50, Table 22 on page 56, Table 96 on page 244, Table 126 on page 303, Table 128 on page 307, Table 132 on page 324, and Table 134 on page 326.
3. Updated "External Memory Interface" on page 26.
4. Updated "Device Identification Register" on page 256.
5. Updated "Electrical Characteristics" on page 322.
6. Updated "ADC Characteristics" on page 328.
7. Updated "ATmega128 Typical Characteristics" on page 336.
8. Updated "Ordering Information" on page 14.

Changes from Rev. 2467J-12/03 to Rev. 2467K-03/04

1. Updated "Errata" on page 17.

Changes from Rev. 2467I-09/03 to Rev. 2467J-12/03

1. Updated "Calibrated Internal RC Oscillator" on page 41.

Changes from Rev. 2467H-02/03 to Rev. 2467I-09/03

1. Updated note in "XTAL Divide Control Register – XDIV" on page 43.
2. Updated "JTAG Interface and On-chip Debug System" on page 48.
3. Updated values for V_{BOT} (BODLEVEL = 1) in Table 19 on page 50.
4. Updated "Test Access Port – TAP" on page 249 regarding JTAGEN.
5. Updated description for the JTD bit on page 258.
6. Added a note regarding JTAGEN fuse to Table 118 on page 291.
7. Updated R_{PU} values in "DC Characteristics" on page 322.
8. Added a proposal for solving problems regarding the JTAG instruction IDCODE in "Errata" on page 17.

Changes from Rev. 2467G-09/02 to Rev. 2467H-02/03

1. Corrected the names of the two Prescaler bits in the SFIOR Register.
2. Added Chip Erase as a first step under "Programming the Flash" on page 319 and "Programming the EEPROM" on page 320.
3. Removed reference to the "Multipurpose Oscillator" application note and the "32 kHz Crystal Oscillator" application note, which do not exist.
4. Corrected OCn waveforms in Figure 52 on page 125.
5. Various minor Timer1 corrections.
6. Added information about PWM symmetry for Timer0 and Timer2.
7. Various minor TWI corrections.
8. Added reference to Table 124 on page 294 from both SPI Serial Programming and Self Programming to inform about the Flash Page size.
9. Added note under "Filling the Temporary Buffer (Page Loading)" on page 283 about writing to the EEPROM during an SPM Page load.
10. Removed ADHSM completely.
11. Added section "EEPROM Write During Power-down Sleep Mode" on page 25.
12. Updated drawings in "Packaging Information" on page 15.

Changes from Rev. 2467F-09/02 to Rev. 2467G-09/02

1. **Changed the Endurance on the Flash to 10,000 Write/Erase Cycles.**

Changes from Rev. 2467E-04/02 to Rev. 2467F-09/02

1. **Added 64-pad QFN/MLF Package and updated "Ordering Information" on page 14.**
2. **Added the section "Using all Locations of External Memory Smaller than 64 KB" on page 33.**
3. **Added the section "Default Clock Source" on page 37.**
4. **Renamed SPMCR to SPMCSR in entire document.**
5. **When using external clock there are some limitations regards to change of frequency. This is descried in "External Clock" on page 42 and Table 131, "External Clock Drive," on page 324.**
6. **Added a sub section regarding OCD-system and power consumption in the section "Minimizing Power Consumption" on page 47.**
7. **Corrected typo (WGM-bit setting) for:**

   "Fast PWM Mode" on page 98 (Timer/Counter0).

   "Phase Correct PWM Mode" on page 100 (Timer/Counter0).

   "Fast PWM Mode" on page 152 (Timer/Counter2).

   "Phase Correct PWM Mode" on page 154 (Timer/Counter2).
8. **Corrected Table 81 on page 193 (USART).**
9. **Corrected Table 102 on page 262 (Boundary-Scan)**
10. **Updated Vil parameter in "DC Characteristics" on page 322.**

Changes from Rev. 2467D-03/02 to Rev. 2467E-04/02

1. **Updated the Characterization Data in Section "ATmega128 Typical Characteristics" on page 336.**
2. **Updated the following tables:**

   Table 19 on page 50, Table 20 on page 54, Table 68 on page 159, Table 102 on page 262, and Table 136 on page 328.
3. **Updated Description of OSCCAL Calibration Byte.**

   In the data sheet, it was not explained how to take advantage of the calibration bytes for 2, 4, and 8 MHz Oscillator selections. This is now added in the following sections:

   Improved description of "Oscillator Calibration Register – OSCCAL" on page 41 and "Calibration Byte" on page 292.

Changes from Rev. 2467C-02/02 to Rev. 2467D-03/02

1. **Added more information about "ATmega103 Compatibility Mode" on page 5.**
2. **Updated Table 2, "EEPROM Programming Time," on page 23.**

3. **Updated typical Start-up Time in Table 7 on page 37, Table 9 and Table 10 on page 39, Table 12 on page 40, Table 14 on page 41, and Table 16 on page 42.**

4. **Updated Table 22 on page 56 with typical WDT Time-out.**

5. **Corrected description of ADSC bit in "ADC Control and Status Register A – ADCSRA" on page 246.**

6. **Improved description on how to do a polarity check of the ADC differential results in "ADC Conversion Result" on page 243.**

7. **Corrected JTAG version numbers in "JTAG Version Numbers" on page 256.**

8. **Improved description of addressing during SPM (usage of RAMPZ) on "Addressing the Flash During Self-Programming" on page 281, "Performing Page Erase by SPM" on page 283, and "Performing a Page Write" on page 283.**

9. **Added not regarding OCDEN Fuse below Table 118 on page 291.**

10. **Updated Programming Figures:**

    Figure 135 on page 293 and Figure 144 on page 305 are updated to also reflect that AVCC must be connected during Programming mode. Figure 139 on page 300 added to illustrate how to program the fuses.

11. **Added a note regarding usage of the PROG_PAGELOAD and PROG_PAGEREAD instructions on page 311.**

12. **Added Calibrated RC Oscillator characterization curves in section "ATmega128 Typical Characteristics" on page 336.**

13. **Updated "Two-wire Serial Interface" section.**

    More details regarding use of the TWI Power-down operation and using the TWI as master with low TWBRR values are added into the data sheet. Added the note at the end of the "Bit Rate Generator Unit" on page 205. Added the description at the end of "Address Match Unit" on page 206.

14. **Added a note regarding usage of Timer/Counter0 combined with the clock. See "XTAL Divide Control Register – XDIV" on page 43.**

Changes from Rev. 2467B-09/01 to Rev. 2467C-02/02

1. **Corrected Description of Alternate Functions of Port G**

   Corrected description of TOSC1 and TOSC2 in "Alternate Functions of Port G" on page 84.

2. **Added JTAG Version Numbers for rev. F and rev. G**

   Updated Table 100 on page 256.

3 **Added Some Preliminary Test Limits and Characterization Data**

Removed some of the TBD's in the following tables and pages:

Table 19 on page 50, Table 20 on page 54, "DC Characteristics" on page 322, Table 131 on page 324, Table 134 on page 326, and Table 136 on page 328.

4. **Corrected "Ordering Information" on page 14.**

5. **Added some Characterization Data in Section "ATmega128 Typical Characteristics" on page 336.**

6. **Removed Alternative Algortihm for Leaving JTAG Programming Mode.**
   See "Leaving Programming Mode" on page 319.

7. **Added Description on How to Access the Extended Fuse Byte Through JTAG Programming Mode.**
   See "Programming the Fuses" on page 321 and "Reading the Fuses and Lock Bits" on page 321.

Atmel Corporation

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

Regional Headquarters

Europe
Atmel Sarl
Route des Arsenaux 41
Case Postale 80
CH-1705 Fribourg
Switzerland
Tel: (41) 26-426-5555
Fax: (41) 26-426-5500

Asia
Room 1219
Chinachem Golden Plaza
77 Mody Road Tsimshatsui
East Kowloon
Hong Kong
Tel: (852) 2721-9778
Fax: (852) 2722-1369

Japan
9F, Tonetsu Shinkawa Bldg.
1-24-8 Shinkawa
Chuo-ku, Tokyo 104-0033
Japan
Tel: (81) 3-3523-3551
Fax: (81) 3-3523-7581

Atmel Operations

Memory
2325 Orchard Parkway
San Jose, CA 95131, USA
Tel: 1(408) 441-0311
Fax: 1(408) 436-4314

Microcontrollers
2325 Orchard Parkway
San Jose, CA 95131, USA
Tel: 1(408) 441-0311
Fax: 1(408) 436-4314

La Chantrerie
BP 70602
44306 Nantes Cedex 3, France
Tel: (33) 2-40-18-18-18
Fax: (33) 2-40-18-19-60

ASIC/ASSP/Smart Cards
Zone Industrielle
13106 Rousset Cedex, France
Tel: (33) 4-42-53-60-00
Fax: (33) 4-42-53-60-01

1150 East Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906, USA
Tel: 1(719) 576-3300
Fax: 1(719) 540-1759

Scottish Enterprise Technology Park
Maxwell Building
East Kilbride G75 0QR, Scotland
Tel: (44) 1355-803-000
Fax: (44) 1355-242-743

RF/Automotive
Theresienstrasse 2
Postfach 3535
74025 Heilbronn, Germany
Tel: (49) 71-31-67-0
Fax: (49) 71-31-67-2340

1150 East Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906, USA
Tel: 1(719) 576-3300
Fax: 1(719) 540-1759

Biometrics/Imaging/Hi-Rel MPU/
High Speed Converters/RF Datacom
Avenue de Rochepleine
BP 123
38521 Saint-Egreve Cedex, France
Tel: (33) 4-76-58-30-00
Fax: (33) 4-76-58-34-80

Literature Requests
www.atmel.com/literature

2467NS–AVR–03/06