

Features

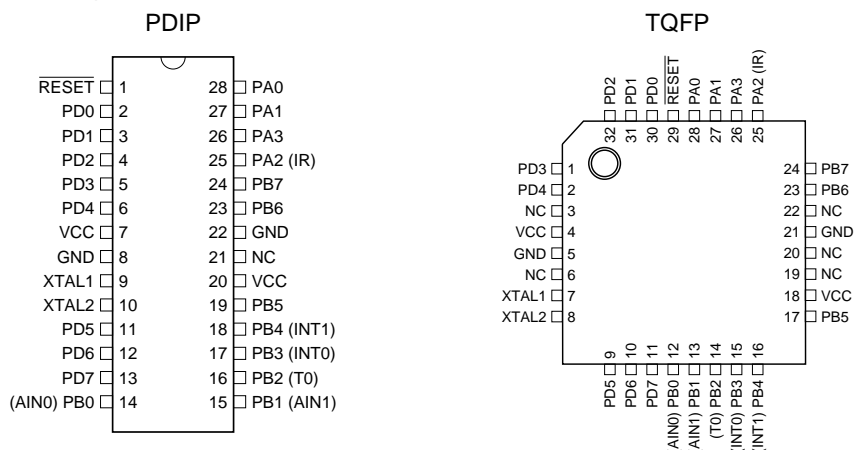
- Utilizes the AVR[®] RISC Architecture
- AVR - High-performance and Low-power RISC Architecture
 - 90 Powerful Instructions - Most Single Clock Cycle Execution
 - 32 x 8 General Purpose Working Registers
 - Up to 4 MIPS Throughput at 4 MHz
- Nonvolatile Program Memory
 - 2K Bytes of Flash Program Memory
 - Endurance: 1,000 Write/Erase Cycles
 - Programmable Lock for Flash Program Data Security
- Peripheral Features
 - Interrupt and Wake-up on Low-level Input
 - One 8-bit Timer/Counter with Separate Prescaler
 - On-chip Analog Comparator
 - Programmable Watchdog Timer with On-chip Oscillator
 - Built-in High-current LED Driver with Programmable Modulation
- Special Microcontroller Features
 - Low-power, Idle and Power-down Modes
 - External and Internal Interrupt Sources
 - Power-on Reset Circuit with Programmable Start-up Time
 - Internal Calibrated RC Oscillator
- Power Consumption 2V at 1 MHz, 25°C
 - Active: TBD
 - Idle Mode: TBD
 - Power-down Mode: <1 μA
- I/O and Packages
 - 11 Programmable I/O Lines, 8 Input Lines and a High-current LED Driver
 - 28-pin PDIP and 32-pin TQFP
- Operating Voltages
 - V_{CC}: 1.8V - 5.5V (ATtiny28V)
 - V_{CC}: 2.7V - 5.5V (ATtiny28L)
- Speed Grades
 - 0 - 1 MHz (ATtiny28V)
 - 0 - 4 MHz (ATtiny28L)

Description

The ATtiny28 is a low-power CMOS 8-bit microcontroller based on the AVR RISC architecture. By executing powerful instructions in a single clock cycle, the ATtiny28 achieves throughputs approaching 1 MIPS per MHz allowing the system designer to

(continued)

Pin Configuration



Rev. 1062BS-10/99



8-bit AVR[®]
Microcontroller
with 2K Bytes
Flash

ATtiny28L
ATtiny28V

Preliminary

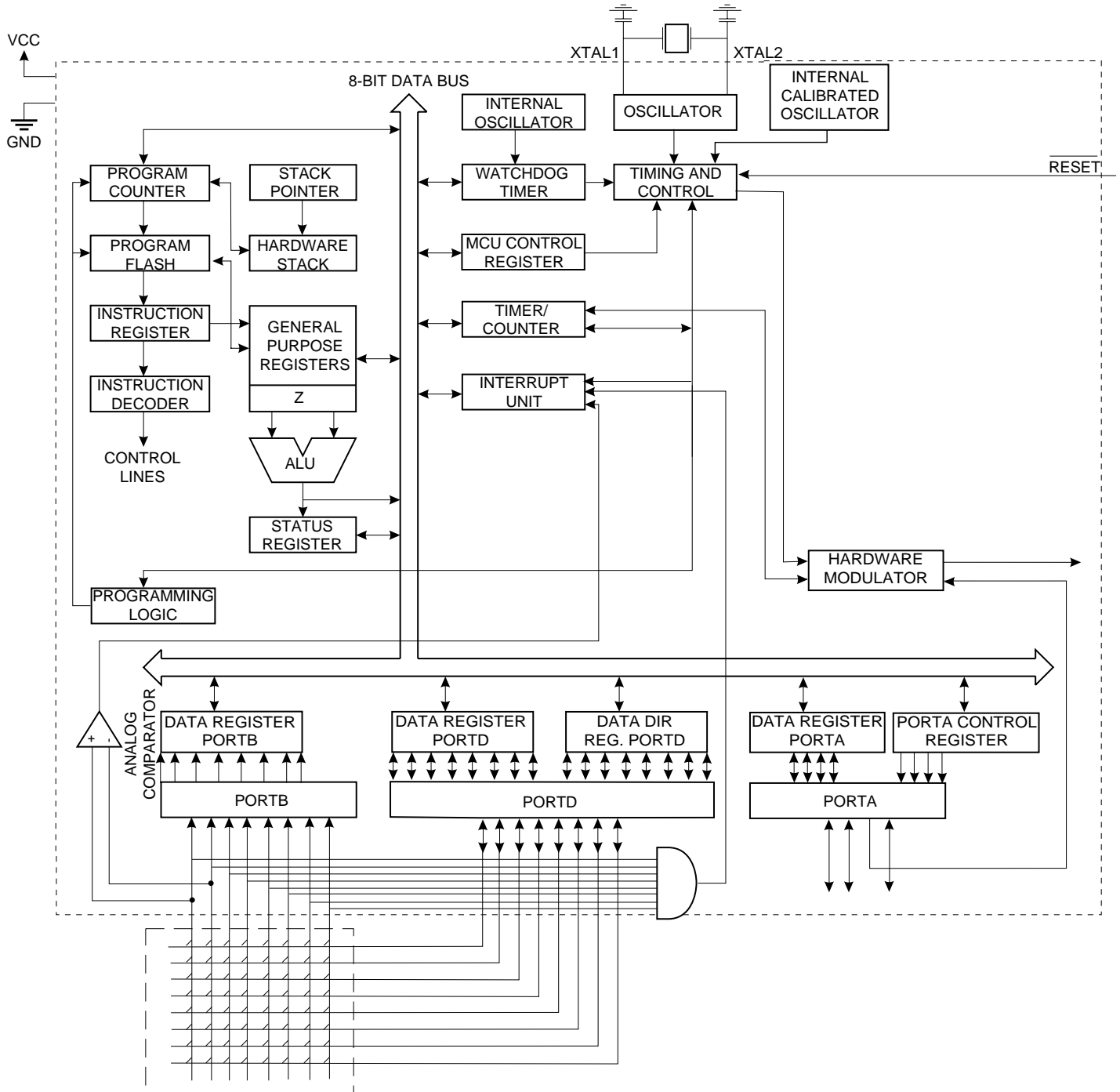


Note: This is a summary document. For the complete 61-page document, please visit our web site at www.atmel.com or e-mail at literature@atmel.com and request literature #1062B.

optimize power consumption versus processing speed. 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.

Block Diagram

Figure 1. The ATtiny28 Block Diagram



The ATtiny28 provides the following features: 2K bytes of Flash, 11 general purpose I/O lines, 8 input lines, a high-current LED driver, 32 general purpose working registers, an 8-bit timer/counter, internal and external interrupts, programmable Watchdog Timer with internal oscillator, and two software selectable power saving modes. The Idle Mode stops the CPU while allowing the timer/counter, 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. The wakeup or interrupt on low level input feature, enables the ATtiny28 to be highly responsive to external events, still featuring the lowest power consumption while in the power-down modes.

The device is manufactured using Atmel's high density nonvolatile memory technology. By combining an enhanced RISC 8-bit CPU with Flash on a monolithic chip, the Atmel ATtiny28 is a powerful microcontroller that provides a highly flexible and cost-effective solution to many embedded control applications. The ATtiny28 AVR is supported with a full suite of program and system development tools including: macro assemblers, program debugger/simulators, in-circuit emulators, and evaluation kits.

Pin Descriptions

VCC

Supply voltage pin

GND

Ground pin

Port A (PA3..PA0)

Port A is a 4-bit I/O port. PA2 is output only and can be used as a high-current LED driver. At $V_{CC} = 1.8V$, the PA2 output buffer can sink 25 mA. PA3, PA1 and PA0 are bidirectional I/O pins with internal pullups (selected for each bit). The port pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port B (PB7..PB0)

Port B is an 8-bit input port with internal pullups (selected for all Port B pins). Port B pins that are externally pulled low will source current if the pullups are activated.

Port B also serves the functions of various special features of the ATtiny28 as listed on page 35. If any of the special features are enabled, the pullup(s) on the corresponding pin(s) is automatically disabled. The port pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port D (PD7..PD0)

Port D is an 8-bit I/O port. Port pins can provide internal pullup resistors (selected for each bit). The port pins are tri-stated when a reset condition becomes active, even if the clock is not running.

XTAL1

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

XTAL2

Output from the inverting oscillator amplifier.

RESET

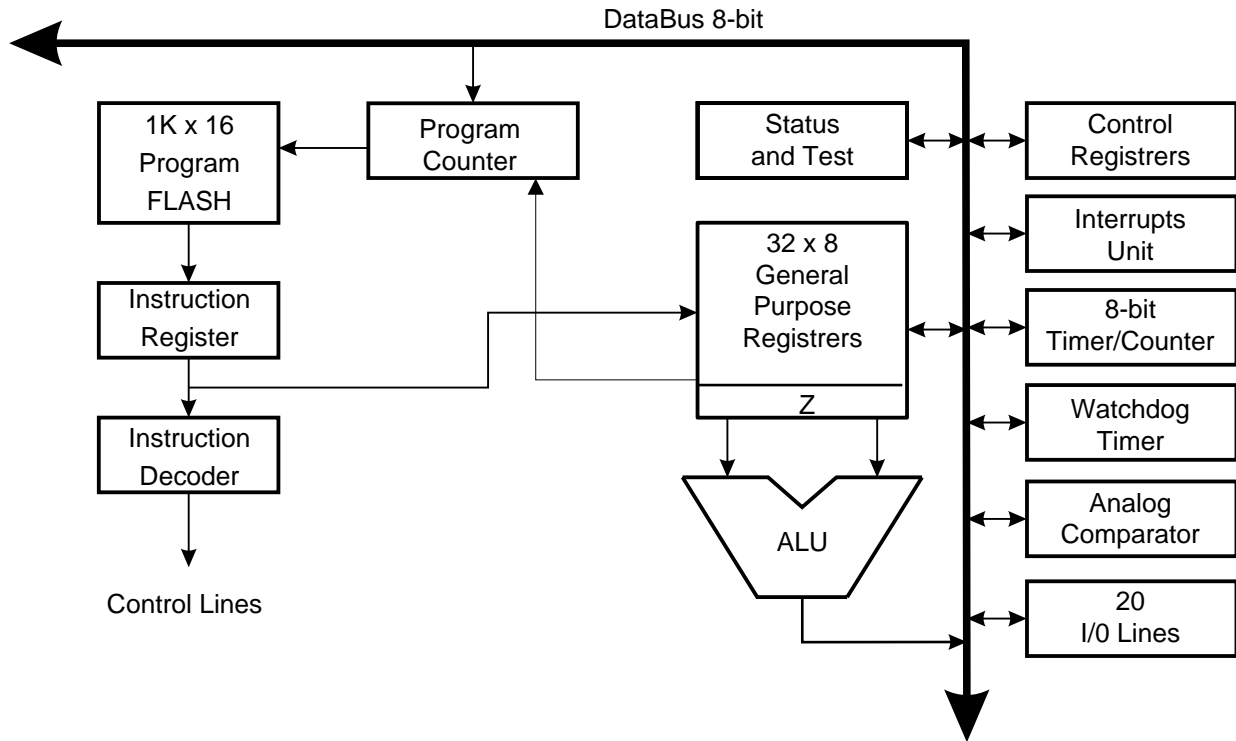
Reset input. An external reset is generated by a low level on the $\overline{\text{RESET}}$ pin. Reset pulses longer than 50 ns will generate a reset, even if the clock is not running. Shorter pulses are not guaranteed to generate a reset.

Architectural Overview

The fast-access register file concept contains 32 x 8-bit general purpose working registers with a single clock cycle access time. This means that during one single clock cycle, one ALU (Arithmetic Logic Unit) operation is executed. Two operands are output from the register file, the operation is executed, and the result is stored back in the register file – in one clock cycle.

Two of the 32 registers can be used as a 16-bit pointer for indirect memory access. This pointer is called the Z-pointer, and can address the register file and the Flash program memory.

Figure 2. The ATtiny28 AVR RISC Architecture



The ALU supports arithmetic and logic functions between registers or between a constant and a register. Single register operations are also executed in the ALU. Figure 2 shows the ATtiny28 AVR RISC microcontroller architecture. The AVR uses a Harvard architecture concept – with separate memories and buses for program and data memories. The program memory is accessed with a two stage pipeline. While one instruction is being executed, the next instruction is pre-fetched from the program memory. This concept enables instructions to be executed every clock cycle. The program memory is reprogrammable Flash memory.

With the relative jump and relative call instructions, the whole 1K address space is directly accessed. All AVR instructions have a single 16-bit word format, meaning that every program memory address contains a single 16-bit instruction.

During interrupts and subroutine calls, the return address program counter (PC) is stored on the stack. The stack is a 3 level deep hardware stack dedicated for subroutines and interrupts.

The I/O memory space contains 64 addresses for CPU peripheral functions as Control Registers, Timer/Counters, and other I/O functions. The memory spaces in the AVR architecture are all linear and regular memory maps.

A flexible interrupt module has its control registers in the I/O space with an additional global interrupt enable bit in the status register. All the different interrupts have a separate interrupt vector in the interrupt vector table at the beginning of the program memory. The different interrupts have priority in accordance with their interrupt vector position. The lower the interrupt vector address, the higher the priority.

Register Summary

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page
\$3F	SREG	I	T	H	S	V	N	Z	C	page 12
\$3E	Reserved									
...	Reserved									
\$20	Reserved									
\$1F	Reserved									
\$1E	Reserved									
\$1D	Reserved									
\$1C	Reserved									
\$1B	PORTA	-	-	-	-	PORTA3	PORTA2	PORTA1	PORTA0	page 32
\$1A	PACR	-	-	-	-	DDA3	PA2HC	DDA1	DDA0	page 32
\$19	PINA	-	-	-	-	PINA3	-	PINA1	PINA0	page 33
\$18	Reserved									
\$17	Reserved									
\$16	PINB	PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0	page 35
\$15	Reserved									
\$14	Reserved									
\$13	Reserved									
\$12	PORTD	PORTD7	PORTD6	PORTD5	PORTD4	PORTD3	PORTD2	PORTD1	PORTD0	page 38
\$11	DDRD	DDD7	DDD6	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0	page 38
\$10	PIND	PIND7	PIND6	PIND5	PIND4	PIND3	PIND2	PIND1	PIND0	page 38
\$0F	Reserved									
\$0E	Reserved									
\$0D	Reserved									
\$0C	Reserved									
\$0B	Reserved									
\$0A	Reserved									
\$09	Reserved									
\$08	ACSR	ACD	-	ACO	ACI	ACIE	-	ACIS1	ACIS0	page 31
\$07	MCUCS	PLUPB	-	SE	SM	WDRF	-	EXTRF	PORF	page 18
\$06	ICR	INT1	INT0	LLIE	TOIE0	ISC11	ISC10	ISC01	ISC00	page 19
\$05	IFR	INTF1	INTF0	-	TOV0	-	-	-	-	page 20
\$04	TCCR0	FOV0	-	-	OOM01	OOM00	CS02	CS01	CS00	page 23
\$03	TCNT0	Timer/Counter0 (8 BIT)								page 24
\$02	MODCR	ONTIM4	ONTIM3	ONTIM2	ONTIM1	ONTIM0	MCONF2	MCONF1	MCONF0	page 27
\$01	WDTCR	-	-	-	WDTOE	WDE	WDP2	WDP1	WDP0	page 25
\$00	OSCCAL	Oscillator Calibration Register								page 26

- 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	$Rd \leftarrow Rd + Rr$	Z,C,N,V,H	1
ADC	Rd, Rr	Add with Carry two Registers	$Rd \leftarrow Rd + Rr + C$	Z,C,N,V,H	1
SUB	Rd, Rr	Subtract two Registers	$Rd \leftarrow Rd - Rr$	Z,C,N,V,H	1
SUBI	Rd, K	Subtract Constant from Register	$Rd \leftarrow Rd - K$	Z,C,N,V,H	1
SBC	Rd, Rr	Subtract with Carry two Registers	$Rd \leftarrow Rd - Rr - C$	Z,C,N,V,H	1
SBCI	Rd, K	Subtract with Carry Constant from Reg.	$Rd \leftarrow Rd - K - C$	Z,C,N,V,H	1
AND	Rd, Rr	Logical AND Registers	$Rd \leftarrow Rd \cdot Rr$	Z,N,V	1
ANDI	Rd, K	Logical AND Register and Constant	$Rd \leftarrow Rd \cdot K$	Z,N,V	1
OR	Rd, Rr	Logical OR Registers	$Rd \leftarrow Rd \vee Rr$	Z,N,V	1
ORI	Rd, K	Logical OR Register and Constant	$Rd \leftarrow Rd \vee K$	Z,N,V	1
EOR	Rd, Rr	Exclusive OR Registers	$Rd \leftarrow Rd \oplus Rr$	Z,N,V	1
COM	Rd	One's Complement	$Rd \leftarrow \text{\$FF} - Rd$	Z,C,N,V	1
NEG	Rd	Two's Complement	$Rd \leftarrow \text{\$00} - Rd$	Z,C,N,V,H	1
SBR	Rd, K	Set Bit(s) in Register	$Rd \leftarrow Rd \vee K$	Z,N,V	1
CBR	Rd, K	Clear Bit(s) in Register	$Rd \leftarrow Rd \cdot (\text{\$FF} - K)$	Z,N,V	1
INC	Rd	Increment	$Rd \leftarrow Rd + 1$	Z,N,V	1
DEC	Rd	Decrement	$Rd \leftarrow Rd - 1$	Z,N,V	1
TST	Rd	Test for Zero or Minus	$Rd \leftarrow Rd \cdot Rd$	Z,N,V	1
CLR	Rd	Clear Register	$Rd \leftarrow Rd \oplus Rd$	Z,N,V	1
SER	Rd	Set Register	$Rd \leftarrow \text{\$FF}$	None	1
BRANCH INSTRUCTIONS					
RJMP	k	Relative Jump	$PC \leftarrow PC + k + 1$	None	2
RCALL	k	Relative Subroutine Call	$PC \leftarrow PC + k + 1$	None	3
RET		Subroutine Return	$PC \leftarrow \text{STACK}$	None	4
RETI		Interrupt Return	$PC \leftarrow \text{STACK}$	I	4
CPSE	Rd,Rr	Compare, Skip if Equal	if $(Rd = Rr)$ $PC \leftarrow PC + 2$ or 3	None	1 / 2
CP	Rd,Rr	Compare	$Rd - Rr$	Z, N, V, C, H	1
CPC	Rd,Rr	Compare with Carry	$Rd - Rr - C$	Z, N, V, C, H	1
CPI	Rd,K	Compare Register with Immediate	$Rd - K$	Z, N, V, C, H	1
SBRC	Rr, b	Skip if Bit in Register Cleared	if $(Rr(b)=0)$ $PC \leftarrow PC + 2$ or 3	None	1 / 2
SBRS	Rr, b	Skip if Bit in Register is Set	if $(Rr(b)=1)$ $PC \leftarrow PC + 2$ or 3	None	1 / 2
SBIC	P, b	Skip if Bit in I/O Register Cleared	if $(P(b)=0)$ $PC \leftarrow PC + 2$ or 3	None	1 / 2
SBIS	P, b	Skip if Bit in I/O Register is Set	if $(P(b)=1)$ $PC \leftarrow PC + 2$ or 3	None	1 / 2
BRBS	s, k	Branch if Status Flag Set	if $(SREG(s) = 1)$ then $PC \leftarrow PC + k + 1$	None	1 / 2
BRBC	s, k	Branch if Status Flag Cleared	if $(SREG(s) = 0)$ then $PC \leftarrow PC + k + 1$	None	1 / 2
BREQ	k	Branch if Equal	if $(Z = 1)$ then $PC \leftarrow PC + k + 1$	None	1 / 2
BRNE	k	Branch if Not Equal	if $(Z = 0)$ then $PC \leftarrow PC + k + 1$	None	1 / 2
BRCS	k	Branch if Carry Set	if $(C = 1)$ then $PC \leftarrow PC + k + 1$	None	1 / 2
BRCC	k	Branch if Carry Cleared	if $(C = 0)$ then $PC \leftarrow PC + k + 1$	None	1 / 2
BRSH	k	Branch if Same or Higher	if $(C = 0)$ then $PC \leftarrow PC + k + 1$	None	1 / 2
BRLO	k	Branch if Lower	if $(C = 1)$ then $PC \leftarrow PC + k + 1$	None	1 / 2
BRMI	k	Branch if Minus	if $(N = 1)$ then $PC \leftarrow PC + k + 1$	None	1 / 2
BRPL	k	Branch if Plus	if $(N = 0)$ then $PC \leftarrow PC + k + 1$	None	1 / 2
BRGE	k	Branch if Greater or Equal, Signed	if $(N \oplus V = 0)$ then $PC \leftarrow PC + k + 1$	None	1 / 2
BRLT	k	Branch if Less Than Zero, Signed	if $(N \oplus V = 1)$ then $PC \leftarrow PC + k + 1$	None	1 / 2
BRHS	k	Branch if Half Carry Flag Set	if $(H = 1)$ then $PC \leftarrow PC + k + 1$	None	1 / 2
BRHC	k	Branch if Half Carry Flag Cleared	if $(H = 0)$ then $PC \leftarrow PC + k + 1$	None	1 / 2
BRTS	k	Branch if T Flag Set	if $(T = 1)$ then $PC \leftarrow PC + k + 1$	None	1 / 2
BRTC	k	Branch if T Flag Cleared	if $(T = 0)$ then $PC \leftarrow PC + k + 1$	None	1 / 2
BRVS	k	Branch if Overflow Flag is Set	if $(V = 1)$ then $PC \leftarrow PC + k + 1$	None	1 / 2
BRVC	k	Branch if Overflow Flag is Cleared	if $(V = 0)$ then $PC \leftarrow PC + k + 1$	None	1 / 2
BRIE	k	Branch if Interrupt Enabled	if $(I = 1)$ then $PC \leftarrow PC + k + 1$	None	1 / 2
BRID	k	Branch if Interrupt Disabled	if $(I = 0)$ then $PC \leftarrow PC + k + 1$	None	1 / 2

Instruction Set Summary (Continued)

Mnemonics	Operands	Description	Operation	Flags	#Clocks
DATA TRANSFER INSTRUCTIONS					
LD	Rd,Z	Load Register Indirect	$Rd \leftarrow (Z)$	None	2
ST	Z,Rr	Store Register Indirect	$(Z) \leftarrow Rr$	None	2
MOV	Rd, Rr	Move Between Registers	$Rd \leftarrow Rr$	None	1
LDI	Rd, K	Load Immediate	$Rd \leftarrow K$	None	1
IN	Rd, P	In Port	$Rd \leftarrow P$	None	1
OUT	P, Rr	Out Port	$P \leftarrow Rr$	None	1
LPM		Load Program Memory	$R0 \leftarrow (Z)$	None	3
BIT AND BIT-TEST INSTRUCTIONS					
SBI	Pb	Set Bit in I/O Register	$I/O(P;b) \leftarrow 1$	None	2
CBI	Pb	Clear Bit in I/O Register	$I/O(P;b) \leftarrow 0$	None	2
LSL	Rd	Logical Shift Left	$Rd(n+1) \leftarrow Rd(n), Rd(0) \leftarrow 0$	Z,C,N,V	1
LSR	Rd	Logical Shift Right	$Rd(n) \leftarrow Rd(n+1), Rd(7) \leftarrow 0$	Z,C,N,V	1
ROL	Rd	Rotate Left Through Carry	$Rd(0) \leftarrow C, Rd(n+1) \leftarrow Rd(n), C \leftarrow Rd(7)$	Z,C,N,V	1
ROR	Rd	Rotate Right Through Carry	$Rd(7) \leftarrow C, Rd(n) \leftarrow Rd(n+1), C \leftarrow Rd(0)$	Z,C,N,V	1
ASR	Rd	Arithmetic Shift Right	$Rd(n) \leftarrow Rd(n+1), n=0..6$	Z,C,N,V	1
SWAP	Rd	Swap Nibbles	$Rd(3..0) \leftarrow Rd(7..4), Rd(7..4) \leftarrow Rd(3..0)$	None	1
BSET	s	Flag Set	$SREG(s) \leftarrow 1$	SREG(s)	1
BCLR	s	Flag Clear	$SREG(s) \leftarrow 0$	SREG(s)	1
BST	Rr, b	Bit Store from Register to T	$T \leftarrow Rr(b)$	T	1
BLD	Rd, b	Bit load from T to Register	$Rd(b) \leftarrow T$	None	1
SEC		Set Carry	$C \leftarrow 1$	C	1
CLC		Clear Carry	$C \leftarrow 0$	C	1
SEN		Set Negative Flag	$N \leftarrow 1$	N	1
CLN		Clear Negative Flag	$N \leftarrow 0$	N	1
SEZ		Set Zero Flag	$Z \leftarrow 1$	Z	1
CLZ		Clear Zero Flag	$Z \leftarrow 0$	Z	1
SEI		Global Interrupt Enable	$I \leftarrow 1$	I	1
CLI		Global Interrupt Disable	$I \leftarrow 0$	I	1
SES		Set Signed Test Flag	$S \leftarrow 1$	S	1
CLS		Clear Signed Test Flag	$S \leftarrow 0$	S	1
SEV		Set Twos Complement Overflow	$V \leftarrow 1$	V	1
CLV		Clear Twos Complement Overflow	$V \leftarrow 0$	V	1
SET		Set T in SREG	$T \leftarrow 1$	T	1
CLT		Clear T in SREG	$T \leftarrow 0$	T	1
SEH		Set Half Carry Flag in SREG	$H \leftarrow 1$	H	1
CLH		Clear Half Carry Flag in SREG	$H \leftarrow 0$	H	1
NOP		No Operation		None	1
SLEEP		Sleep	(see specific descr. for Sleep function)	None	3
WDR		Watch Dog Reset	(see specific descr. for WDR/timer)	None	1



Atmel Headquarters

Corporate Headquarters

2325 Orchard Parkway
San Jose, CA 95131
TEL (408) 441-0311
FAX (408) 487-2600

Europe

Atmel U.K., Ltd.
Coliseum Business Centre
Riverside Way
Camberley, Surrey GU15 3YL
England
TEL (44) 1276-686-677
FAX (44) 1276-686-697

Asia

Atmel Asia, Ltd.
Room 1219
Chinachem Golden Plaza
77 Mody Road Tsimhatsui
East Kowloon
Hong Kong
TEL (852) 2721-9778
FAX (852) 2722-1369

Japan

Atmel Japan K.K.
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

Atmel Colorado Springs

1150 E. Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906
TEL (719) 576-3300
FAX (719) 540-1759

Atmel Rousset

Zone Industrielle
13106 Rousset Cedex
France
TEL (33) 4-4253-6000
FAX (33) 4-4253-6001

Fax-on-Demand

North America:
1-(800) 292-8635
International:
1-(408) 441-0732

e-mail

literature@atmel.com

Web Site

<http://www.atmel.com>

BBS

1-(408) 436-4309

© Atmel Corporation 1999.

Atmel Corporation makes no warranty for the use of its products, other than those expressly contained in the Company's standard warranty which is detailed in Atmel's Terms and Conditions located on the Company's web site. The Company assumes no responsibility for any errors which may appear in this document, reserves the right to change devices or specifications detailed herein at any time without notice, and does not make any commitment to update the information contained herein. No licenses to patents or other intellectual property of Atmel are granted by the Company in connection with the sale of Atmel products, expressly or by implication. Atmel's products are not authorized for use as critical components in life support devices or systems.

Marks bearing ® and/or ™ are registered trademarks and trademarks of Atmel Corporation.

Terms and product names in this document may be trademarks of others.



Printed on recycled paper.

1062BS-10/99/xM