# E·XFL



Welcome to E-XFL.COM

#### What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

## Applications of "<u>Embedded -</u> <u>Microcontrollers</u>"

#### Details

Product Status	Obsolete
Core Processor	AVR
Core Size	8-Bit
Speed	4MHz
Connectivity	SPI
Peripherals	Brown-out Detect/Reset, POR, WDT
Number of I/O	3
Program Memory Size	2KB (1K x 16)
Program Memory Type	FLASH
EEPROM Size	128 x 8
RAM Size	128 x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 6V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	0°C ~ 70°C
Mounting Type	Through Hole
Package / Case	8-DIP (0.300", 7.62mm)
Supplier Device Package	8-PDIP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/at90ls2323-4pc

Email: info@E-XFL.COM

Address: Room A, 16/F, Full Win Commercial Centre, 573 Nathan Road, Mongkok, Hong Kong



## Description

The AT90S/LS2323 and AT90S/LS2343 are low-power, CMOS, 8-bit microcontrollers based on the AVR RISC architecture. By executing powerful instructions in a single clock cycle, the AT90S2323/2343 achieves throughputs approaching 1 MIPS per MHz allowing the system designer to 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.



Figure 1. The AT90S/LS2343 Block Diagram





Figure 2. The AT90S/LS2323 Block Diagram

The AT90S2323/2343 provides the following features: 2K bytes of In-System Programmable Flash, 128 bytes EEPROM, 128 bytes SRAM, 3 (AT90S/LS2323)/5 (AT90S/LS2343) general-purpose I/O lines, 32 general-purpose working registers, an 8bit timer/counter, internal and external interrupts, programmable Watchdog Timer with internal oscillator, an SPI serial port for Flash Memory downloading and two softwareselectable 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.

The device is manufactured using Atmel's high-density nonvolatile memory technology. The On-chip Flash allows the program memory to be reprogrammed in-system through an SPI serial interface. By combining an 8-bit RISC CPU with ISP Flash on a monolithic





chip, the Atmel AT90S2323/2343 is a powerful microcontroller that provides a highly flexible and cost-effective solution to many embedded control applications.

The AT90S2323/2343 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.

## Comparison between AT90S/LS2323 and AT90S/LS2343

The AT90S/LS2323 is intended for use with external quartz crystal or ceramic resonator as the clock source. The start-up time is fuse-selectable as either 1 ms (suitable for ceramic resonator) or 16 ms (suitable for crystal). The device has three I/O pins.

The AT90S/LS2343 is intended for use with either an external clock source or the internal RC oscillator as clock source. The device has five I/O pins.

Table 1 summarizes the differences in features of the two devices.

Table 1.	Feature	Difference	Summary
----------	---------	------------	---------

Part	AT90S/LS2323	AT90S/LS2343
On-chip Oscillator Amplifier	yes	no
Internal RC Clock	no	yes
PB3 available as I/O pin	never	internal clock mode
PB4 available as I/O pin	never	always
Start-up time	1 ms/16 ms	16 µs fixed

## Pin Descriptions AT90S/LS2323

VCC	Supply voltage pin.
GND	Ground pin.
Port B (PB2PB0)	Port B is a 3-bit bi-directional I/O port with internal pull-up resistors. The Port B output buffers can sink 20 mA. As inputs, Port B pins that are externally pulled low, will source current if the pull-up resistors are activated.
	Port B also serves the functions of various special features.
	Port pins can provide internal pull-up resistors (selected for each bit). The Port B pins are tri-stated when a reset condition becomes active.
RESET	Reset input. An external reset is generated by a low level on the 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.
XTAL1	Input to the inverting oscillator amplifier and input to the internal clock operating circuit.
XTAL2	Output from the inverting oscillator amplifier.



Program and Data Addressing Modes The AT90S2323/2343 AVR RISC microcontroller supports powerful and efficient addressing modes for access to the program memory (Flash) and data memory. This section describes the different addressing modes supported by the AVR architecture. In the figures, OP means the operation code part of the instruction word. To simplify, not all figures show the exact location of the addressing bits.

Figure 10. Direct Single Register Addressing

Register Direct, Single Register Rd





**Register Direct, Two Registers** Figure 11. Direct Register Addressing, Two Registers Rd and Rr



Operands are contained in register r (Rr) and d (Rd). The result is stored in register d (Rd).

# AMEL

## Memory Access and Instruction Execution Timing

This section describes the general access timing concepts for instruction execution and internal memory access.

The AVR CPU is driven by the System Clock Ø, directly generated from the external clock signal applied to the CLOCK pin. No internal clock division is used.

Figure 21. shows the parallel instruction fetches and instruction executions enabled by the Harvard architecture and the fast-access register file concept. This is the basic pipelining concept to obtain up to 1 MIPS per MHz with the corresponding unique results for functions per cost, functions per clocks and functions per power unit.





Figure 22. shows the internal timing concept for the register file. In a single clock cycle an ALU operation using two register operands is executed and the result is stored back to the destination register.





The internal data SRAM access is performed in two System Clock cycles as described in Figure 23.



and OUT, the I/O addresses \$00 - \$3F must be used. When addressing I/O registers as SRAM, \$20 must be added to these addresses. All I/O register addresses throughout this document are shown with the SRAM address in parentheses.

For compatibility with future devices, reserved bits should be written to zero if accessed. Reserved I/O memory addresses should never be written.

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

The I/O and peripherals control registers are explained in the following sections.

Status Register – SREG The AVR Status Register (SREG) at I/O space location \$3F (\$5F) is defined as:

Bit	7	6	5	4	3	2	1	0	
\$3F (\$5F)	I	Т	Н	S	v	Ν	Z	С	SREG
Read/Write	R/W	-							
Initial Value	0	0	0	0	0	0	0	0	

## • Bit 7 – I: Global Interrupt Enable

The global interrupt enable bit must be set (one) for the interrupts to be enabled. The individual interrupt enable control is then performed in separate control registers. If the global interrupt enable register is cleared (zero), none of the interrupts are enabled independent of the individual interrupt enable settings. The I-bit is cleared by hardware after an interrupt has occurred and is set by the RETI instruction to enable subsequent interrupts.

## • Bit 6 – T: Bit Copy Storage

The bit copy instructions BLD (Bit LoaD) and BST (Bit STore) use the T-bit as source and destination for the operated bit. A bit from a register in the register file can be copied into T by the BST instruction and a bit in T can be copied into a bit in a register in the register file by the BLD instruction.

## • Bit 5 – H: Half-carry Flag

The half-carry flag H indicates a half-carry in some arithmetic operations. See the Instruction Set description for detailed information.

• Bit 4 – S: Sign Bit, S = N ⊕ V

The S-bit is always an exclusive or between the negative flag N and the two's complement overflow flag V. See the Instruction Set description for detailed information.

## • Bit 3 – V: Two's Complement Overflow Flag

The two's complement overflow flag V supports two's complement arithmetics. See the Instruction Set description for detailed information.

## • Bit 2 – N: Negative Flag

The negative flag N indicates a negative result from an arithmetical or logical operation. See the Instruction Set description for detailed information.

## • Bit 1 – Z: Zero Flag

The zero flag Z indicates a zero result from an arithmetical or logical operation. See the Instruction Set description for detailed information.



## The most typical program setup for the Reset and Interrupt vector addresses are:

Address	Labels	Code	Comments
\$000		rjmp RESET	; Reset Handler
\$001		rjmp EXT_INTO	; IRQ0 Handler
\$002		rjmp TIM_OVF0	; Timer0 Overflow ; Handler;
\$003	MAIN:	ldi r16, low(RAMEND) out SPL, r16 <instr> xxx</instr>	; Main program start

## **Reset Sources**

The AT90S2323/2343 provides three sources of reset:

- Power-on Reset. The MCU is reset when the supply voltage is below the Power-on Reset threshold ( $V_{POT}$ ).
- External Reset. The MCU is reset when a low level is present on the RESET pin for more than 50 ns.
- Watchdog Reset. The MCU is reset when the Watchdog timer period expires and the Watchdog is enabled.

During reset, all I/O registers are set to their initial values and the program starts execution from address \$000. The instruction placed in address \$000 must be an RJMP (relative jump) instruction to the reset handling routine. If the program never enables an interrupt source, the interrupt vectors are not used and regular program code can be placed at these locations. The circuit diagram in Figure 24 shows the reset logic. Table 4 defines the timing and electrical parameters of the reset circuitry.

Figure 24. Reset Logic



The AT90S/LS2323 has a programmable start-up time. A fuse bit (FSTRT) in the Flash memory selects the shortest start-up time when programmed ("0"). The AT90S/LS2323 is shipped with this bit unprogrammed.

The AT90S/LS2343 has a fixed start-up time.



## Timer/Counter

The AT90S2323/2343 provides one general-purpose 8-bit Timer/Counter – Timer/Counter0. The Timer/Counter has prescaling selection from the 10-bit prescaling timer. The Timer/Counter can be used either as a timer with an internal clock time base or as a counter with an external pin connection that triggers the counting.

**Timer/Counter Prescaler** 

Figure 29 shows the Timer/Counter prescaler.



The four different prescaled selections are: CK/8, CK/64, CK/256 and CK/1024, where CK is the oscillator clock. CK, external source and stop can also be selected as clock sources.

## 8-bit Timer/Counter0 Figure 30 shows the block diagram for Timer/Counter0.

The 8-bit Timer/Counter0 can select clock source from CK, prescaled CK or an external pin. In addition, it can be stopped as described in the specification for the Timer/Counter0 Control Register (TCCR0). The overflow status flag is found in the Timer/Counter Interrupt Flag Register (TIFR). Control signals are found in the Timer/Counter0 Control Register (TCCR0). The interrupt enable/disable settings for Timer/Counter0 are found in the Timer/Counter Interrupt Mask Register (TIMSK).

When Timer/Counter0 is externally clocked, the external signal is synchronized with the oscillator frequency of the CPU. To ensure proper sampling of the external clock, the minimum time between two external clock transitions must be at least one internal CPU clock period. The external clock signal is sampled on the rising edge of the internal CPU clock.

The 8-bit Timer/Counter0 features both a high-resolution and a high-accuracy usage with the lower prescaling opportunities. Similarly, the high prescaling opportunities make the Timer/Counter0 useful for lower speed functions or exact timing functions with infrequent actions.





• Bits 7..3 - Res: Reserved Bits

These bits are reserved bits in the AT90S2323/2343 and always read zero.

## • Bits 2, 1, 0 – CS02, CS01, CS00: Clock Select0, Bits 2, 1 and 0

The Clock Select0 bits 2, 1 and 0 define the prescaling source of Timer/Counter0.

Table 10. Clock 0 Prescale Select

CS02	CS01	CS00	Description
0	0	0	Stop, the Timer/Counter0 is stopped.
0	0	1	СК
0	1	0	СК/8
0	1	1	CK/64
1	0	0	CK/256
1	0	1	CK/1024
1	1	0	External Pin T0, falling edge
1	1	1	External Pin T0, rising edge



Timer/Counter0 Control Register – TCCR0

# AIMEL

## Prevent EEPROM Corruption

During periods of low V<sub>CC</sub>, the EEPROM data can be corrupted because the supply voltage is too low for the CPU and the EEPROM to operate properly. These issues are the same as for board level systems using the EEPROM and the same design solutions should be applied.

An EEPROM data corruption can be caused by two situations when the voltage is too low. First, a regular write sequence to the EEPROM requires a minimum voltage to operate correctly. Secondly, the CPU itself can execute instructions incorrectly, if the supply voltage for executing instructions is too low.

EEPROM data corruption can easily be avoided by following these design recommendations (one is sufficient):

- Keep the AVR RESET active (low) during periods of insufficient power supply voltage. This is best done by an external low V<sub>CC</sub> Reset Protection circuit, often referred to as a Brown-out Detector (BOD). Please refer to application note AVR 180 for design considerations regarding power-on reset and low-voltage detection.
- Keep the AVR core in Power-down Sleep mode during periods of low V<sub>CC</sub>. This will prevent the CPU from attempting to decode and execute instructions, effectively protecting the EEPROM registers from unintentional writes.
- 3. Store constants in Flash memory if the ability to change memory contents from software is not required. Flash memory cannot be updated by the CPU and will not be subject to corruption.



## Port B Input Pins Address – PINB

Bit	7	6	5	4	3	2	1	0	_
\$16 (\$36)	-	-	-	PINB4	PINB3	PINB2	PINB1	PINB0	PINB
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	0	0	0	N/A	N/A	N/A	N/A	N/A	

The Port B Input Pins address (PINB) is not a register and this address enables access to the physical value on each Port B pin. When reading PORTB, the Port B Data Latch is read and when reading PINB, the logical values present on the pins are read.

## **Port B as General Digital** All pins in port B have equal functionality when used as digital I/O pins.

PBn, general I/O pin: The DDBn bit in the DDRB register selects the direction of this pin, if DDBn is set (one), PBn is configured as an output pin. If DDBn is cleared (zero), PBn is configured as an input pin. If PORTBn is set (one) when the pin is configured as an input pin, the MOS pull-up resistor is activated. To switch the pull-up resistor off, the PORTBn has to be cleared (zero) or the pin has to be configured as an output pin. The port pins are tri-stated when a reset condition becomes active, even if the clock is not running.

DDBn	PORTBn	I/O	Pull-up	Comment
0	0	Input	No	Tri-state (high-Z)
0	1	Input	Yes	PBn will source current if ext. pulled low
1	0	Output	No	Push-pull Zero Output
1	1	Output	No	Push-pull One Output

Table 13. DDBn Effects on Port B Pins

Alternate Functions of Port B

**B** The alternate pin functions of Port B are as follows:

## • CLOCK – Port B, Bit 3

Clock input: AT90S/LS2343 only. When the RCEN fuse is programmed and the device runs from the internal RC oscillator, this pin is a general I/O pin. When the RCEN fuse is unprogrammed, an external clock source must be connected to CLOCK.

## • SCK/T0 – Port B, Bit 2

In Serial Programming mode, this bit serves as the serial clock input, SCK.

During normal operation, this pin can serve as the external counter clock input. See the timer/counter description for further details. If external timer/counter clocking is selected, activity on this pin will clock the counter even if it is configured as an output.

## • MISO/INT0 - Port B, Bit 1

In Serial Programming mode, this bit serves as the serial data output, MISO.

During normal operation, this pin can serve as the external interrupt0 input. See the interrupt description for details on how to enable this interrupt. Note that activity on this pin will trigger the interrupt even if the pin is configured as an output.

## • MOSI – Port B, Bit 0

In Serial Programming mode, this pin serves as the serial data input, MOSI.

			Instructio	on Format		
Instruction		Instr.1	Instr.2	Instr.3	Instr.4	<b>Operation Remarks</b>
Read Fuse and Lock Bits (AT90S/ LS2323)	PB0 PB1 PB2	0_0000_0100_00 0_0100_1100_00 x_xxxx_xxx	0_0000_0000_00 0_0111_1000_00 x_xxxx_xxx	0_0000_0000_00 0_0111_1100_00 <i>1_2</i> Sxx_xxRx_xx		Reading <i>1</i> , <i>2</i> , <b>S</b> , <b>R</b> = "0" means the Fuse/Lock bit is programmed.
Read Fuse and Lock Bits (AT90S/ LS2343)	PB0 PB1 PB2	0_0000_0100_00 0_0100_1100_00 x_xxxx_xxx	0_0000_0000_00 0_0111_1000_00 x_xxxx_xxx	0_0000_0000_00 0_0111_1100_00 <i>1_2</i> \$xx_xx <b>R</b> x_xx		Reading <i>1</i> , <i>2</i> , <b>S</b> , <b>R</b> = "0" means the Fuse/Lock bit is programmed.
Read Signature Bytes	PB0 PB1 PB2	0_0000_1000_00 0_0100_1100_00 x_xxxx_xxx	0_0000_00 <b>bb</b> _00 0_0000_1100_00 x_xxxx_xxxx_xx	0_0000_0000_00 0_0110_1000_00 x_xxxx_xxx	0_0000_0000_00 0_0110_1100_00 <b>o_0000_000x_</b> xx	Repeat Instr.2 - Instr.4 for each signature byte address.

Table 16. High-voltage Serial Programming Instruction Set (Continued)	)
---	---

Note: **a** = address high bits

**b** = address low bits

i = data in

**o** = data out

x = don't care

1 = Lock Bit1

2 = Lock Bit2

F = FSTRT Fuse

**R** = RCEN Fuse **S** = SPIEN Fuse

S = SFIEN FUSE



## AT90S/LS2323/2343

## **Electrical Characteristics**

## **Absolute Maximum Ratings\***

with respect to Ground1.0V to V <sub>CC</sub> + 0.5V
Voltage on RESET with Respect to Ground1.0V to +13.0V
Maximum Operating Voltage 6.6V
DC Current per I/O Pin 40.0 mA

\*NOTICE: Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

## **DC Characteristics**

$T_{1} = -40^{\circ}C$ to $85^{\circ}C$ $V_{22} = 2.7V$ to 6.0V	(unless otherwise noted)
$T_A = +0.0100000, V_{CC} = 2.70100.000$	

Symbol	Parameter	Condition	Min	Тур	Мах	Units
V <sub>IL</sub>	Input Low Voltage	(Except XTAL)	-0.5		0.3 V <sub>CC</sub> <sup>(1)</sup>	V
V <sub>IL1</sub>	Input Low Voltage	XTAL	-0.5		0.1 <sup>(1)</sup>	V
V <sub>IH</sub>	Input High Voltage	(Except XTAL, RESET)	0.6 V <sub>CC</sub> <sup>(2)</sup>		V <sub>CC</sub> + 0.5	V
V <sub>IH1</sub>	Input High Voltage	XTAL	0.7 V <sub>CC</sub> <sup>(2)</sup>		V <sub>CC</sub> + 0.5	V
V <sub>IH2</sub>	Input High Voltage	RESET	0.85 V <sub>CC</sub> <sup>(2)</sup>		V <sub>CC</sub> + 0.5	V
V <sub>OL</sub>	Output Low Voltage Ports B	$I_{OL} = 20 \text{ mA}, V_{CC} = 5V$ $I_{OL} = 10 \text{ mA}, V_{CC} = 3V$			0.5 0.4	V V
V <sub>OH</sub>	Output High Voltage Ports B	$I_{OH} = -3 \text{ mA}, V_{CC} = 5V$ $I_{OH} = -1.5 \text{ mA}, V_{CC} = 3V$	4.2 2.4			V V
I <sub>IL</sub>	Input Leakage Current I/O Pin	V <sub>CC</sub> = 6V, Pin Low (absolute value)			8.0	μA
I <sub>IH</sub>	Input Leakage Current I/O Pin	V <sub>CC</sub> = 6V, Pin High (absolute value)			8.0	μA
RRST	Reset Pull-up		100.0		500.0	kΩ
R <sub>I/O</sub>	I/O Pin Pull-up		30.0		150.0	kΩ
	Power Supply Current AT90S2343	Active 4 MHz, $V_{CC}$ = 3V			3.0	mA
I <sub>cc</sub>		Idle 4 MHz, V <sub>CC</sub> = 3V			1.1	mA
		Power-down 4 MHz <sup>(3)</sup> , $V_{CC}$ = 3V WDT Enabled			25.0	μΑ
		Power-down 4 MHz <sup>(3)</sup> , $V_{CC} = 3V$ WDT Disabled			20.0	μA
	Power Supply Current AT90S2323	Active 4 MHz, V <sub>CC</sub> = 3V			4.0	mA
		Idle 4 MHz, V <sub>CC</sub> = 3V		1.0	1.2	mA
		$\begin{tabular}{lllllllllllllllllllllllllllllllllll$		9.0	15.0	μΑ
		$\begin{array}{ c c } \hline Power-down^{(3)}, \\ V_{CC} = 3V \text{ WDT Disabled} \end{array}$		<1.0	2.0	μΑ

1. "Max" means the highest value where the pin is guaranteed to be read as low.

2. "Min" means the lowest value where the pin is guaranteed to be read as high.

3. Minimum  $V_{CC}$  for Power-down is 2V.



## Typical Characteristics

The following charts show typical behavior. These figures are not tested during manufacturing. All current consumption measurements are performed with all I/O pins configured as inputs and with internal pull-ups enabled. A sine wave generator with rail-to-rail output is used as clock source.

The current consumption is a function of several factors such as: operating voltage, operating frequency, loading of I/O pins, switching rate of I/O pins, code executed and ambient temperature. The dominating factors are operating voltage and frequency.

The current drawn from capacitive loaded pins may be estimated (for one pin) as  $C_L \bullet V_{CC} \bullet f$  where  $C_L$  = load capacitance,  $V_{CC}$  = operating voltage and f = average switching frequency of I/O pin.

The parts are characterized at frequencies higher than test limits. Parts are not guaranteed to function properly at frequencies higher than the ordering code indicates.

The difference between current consumption in Power-down mode with Watchdog Timer enabled and Power-down mode with Watchdog Timer disabled represents the differential current drawn by the Watchdog Timer.









Figure 44. Idle Supply Current vs. V<sub>CC</sub>

















## Instruction Set Summary

NHTMERT CARD. Code:     No.       ADD:     R1. fr:     Add hon Seguints     R1 fr1fr2.     Z.G.N.H.     1       ADC     R0. fr:     Add hon Seguints     R1 fr1fr2.     Z.G.N.H.     1       ADC     R0. fr:     Add hon Seguints     R2 R0 Rr.     Z.G.N.H.     1       SUB     R0. fr:     Subtox Containton Register     R2 R0 Rr.     Z.G.N.H.     1       SUB     R0. fr:     Subtox Containton Register     R2 R0 Rr.     Z.G.N.H.     1       SSC     R3. fr:     Subtox Containton Register     R2 R0 Rr.     Z.G.N.H.     1       SSC     R3. fr:     Subtox of Macro To Register     R2 R0 Rr.     Z.G.N.H.     1       SSC     R3. fr:     Subtox of Macro To Register     R2 R0 K C.     Z.G.N.H.     1       SSC     R3. fr:     Subtox Ontainton Register     R2 R0 K C.     Z.G.N.H.     1       SSC     R3. fr:     Subtox Ontainton Register     R2 R0 K C.     Z.G.N.H.     1       ASC     Subtox Ontainton Register     R2 R0 K C.     Z.A.N.H.	Mnemonic	Operands	Description	Operation	Flags	# Clocks
ADC     Rd, Fr.     Add the Registra     Pdx-fd + fr/s     ZCRVH     1       ADC     Rd, Fr.     Add with Cary The Registra     Pdx-fd + Fr + C     ZCRVS     2       ADW     Rd, Fr.     Solitable Too Registra     Pdx-fd - File File K.     ZCRVS     2       SUB     Rd, Fr.     Solitable Too Registra     Pdx-fd - File File K.     ZCRVM     1       SUB     Rd, K.     Solitable Introduction Registra     Pdx-fd - File File K.     ZCRVM     1       SUB     Rd, K.     Solitable Introduction Registra     Pdx-fd - File File File K.     ZCRVM     1       SSI     Rd, K.     Solitable Introduction Registra     Pdx-fd - File - C     ZCRVM     1       ADD     Rd, Fr.     Solitable Registra and contant     Pdx-fd + File File     ZAV     1       ADD     Rd, Fr.     Logical OB Registra and contant     Pdx-fd + File File     ZAV     1       COM     Rd     Trois Complement     Pdx-fd + File File     ZAV     1       COM     Rd + File File     SCRVM     SCRVM     1     1       COM <td>ARITHMETIC AND</td> <td>OGIC INSTRUCTION</td> <td>S</td> <td></td> <td>- <b>3</b> -</td> <td></td>	ARITHMETIC AND	OGIC INSTRUCTION	S		- <b>3</b> -	
AOC     Bit Br     Add sum Gray Tan Registers     Bit Ar. Add Turnediate Word     Derivation     Convertion     Convertion <td>ADD</td> <td>Rd. Br</td> <td>Add Two Begisters</td> <td>Bd ← Bd + Br</td> <td>Z.C.N.V.H</td> <td>1</td>	ADD	Rd. Br	Add Two Begisters	Bd ← Bd + Br	Z.C.N.V.H	1
ADW     Path     Addition Too Segment     Path     ZOAVS     2       SUB     BS, Hr.     Subtract Too Segment     Ret – Ret – R.     ZOAVS     1       SUB     BS, Kr.     Subtract Too Segment     Ret – Ret – R.     ZOAVS     2       SUB     BS, Kr.     Subtract Too Segment     Ret –	ADC	Rd, Rr	Add with Carry Two Registers	$Rd \leftarrow Rd + Rr + C$	Z,C,N,V,H	1
SUB     P6 Ar.     Submot Construction Register     P6 - A     ZCAV H     1       SBW     P6 K     Submat Construction Register     P6 - A     ZCAV S     2       SBW     P6 K     Submat Construction Register     P6 - A     P6 - K     ZCAV S     2       SBC     P6 K     Submat Construction Register     P6 - A     A     CAV H     1       SBC     P6 K     Submat Construction Register     P6 - A     CAV H     1       SBC     P6 K     Submat Construction Register     P6 - A     CAV H     1       SBC     P6 K     Logical AD Register and Construct     P6 - A     ZAV     X     1       ADR     P6 K     Logical AD Register and Construct     P6 - A     ZAV     X     1       COM     P6 K     Logical AD Register and Construct     P6 - A     ZAV     1     1       COM     P6 K     Construct Construct     P6 - A     ZAV     1     1       COM     P6 K     Construct Construct     P6 - A     ZAV     1     1 <t< td=""><td>ADIW</td><td>Rdl, K</td><td>Add Immediate to Word</td><td><math>Rdh:Rdl \leftarrow Rdh:Rdl + K</math></td><td>Z,C,N,V,S</td><td>2</td></t<>	ADIW	Rdl, K	Add Immediate to Word	$Rdh:Rdl \leftarrow Rdh:Rdl + K$	Z,C,N,V,S	2
By Bit     Bat/s     Subtract Internation Program     Pat – Pat – K.     ZOAV JA     1       SBW     Bat/s     Subtract Internation Provided Internation Provided Patholics     ZOAV JA     2       SBC     Bat/s     Subtract Internation Provided Patholics     Rei – Rei – Rei – Rei – C.     ZOAV JA     1       SBC     Bat/s     Subtract Internation Provided Patholics     Rei – Rei – Rei – Rei – C.     ZOAV JA     1       AND     Bat/s     Logical AND Register and Containt     Rei –	SUB	Rd, Rr	Subtract Two Registers	Rd ← Rd – Rr	Z,C,N,V,H	1
SBWRb,RSolution throadiate from WordRb,RHRb,RHL,CN,VSZ,N,VSZSBCRb,RSolution throadiate from Rug.Rd-Rd-R-CZ,N,VH1SBCRb,RLogical AND RegistersRd-Rd-Rd-Rd-CZ,N,VK1ANDRb,RLogical AND Register and ConstantRd-Rd-Rd-Rd-Rd-Rd-Rd-Rd-Rd-Rd-Rd-Rd-Rd-R	SUBI	Rd, K	Subtract Constant from Register	$Rd \leftarrow Rd - K$	Z,C,N,V,H	1
SRCRA,KSolver, Yon Quere into Panel A, P	SBIW	Rdl, K	Subtract Immediate from Word	$Rdh:Rdl \leftarrow Rdh:Rdl - K$	Z,C,N,V,S	2
SBC1Pd.ASederat with Carry Constant term PeanPd.APd.AC. AV.M1ANDPd.ALogical AND Register and ConstantPd.APd.APd.AN.M.M1ANDPd.ALogical CA Register and ConstantPd.APd.APd.AN.M.M1CPIPd.ALogical CA Register and ConstantPd.APd.APd.AN.M.M1CPIPd.ALogical CA Register and ConstantPd.APd.APd.AN.M.M1CDRPd.ADescomperationPd.APd.ASch.N.M11COMPdOris ComplementPd.APd.AC.N.N.M11SBRPd.ASet Biglish RegisterPd.APd.AC.N.N.M11SBRPd.ACore Pd.N.M. PogeterPd.APd.AC.N.N.M11NGCPd.ACore Pd.APd.APd.APd.A2.N.N.M11SBRPd.ACore Pd.APd.APd.APd.A2.N.N.M11SBRPd.ATote Zono A MinasPd.APd.APd.A11 </td <td>SBC</td> <td>Rd, Rr</td> <td>Subtract with Carry Two Registers</td> <td><math display="block">Rd \gets Rd - Rr - C</math></td> <td>Z,C,N,V,H</td> <td>1</td>	SBC	Rd, Rr	Subtract with Carry Two Registers	$Rd \gets Rd - Rr - C$	Z,C,N,V,H	1
ANDPAG.Logent AND RegistersPAG + PA + PAZ.N.V11ORPAG.KLogent AND Registers and ConstantPAG + PA + FAZ.N.V11ORPAG.KLogent OR Registers and ConstantPAG + PA + FAZ.N.V11EORPAG.KLogent OR Registers and ConstantPAG + PAG + FAZ.N.V11EORPAG.KLogent OR Registers and ConstantPAG + PAG + FAZ.N.V11EORPAGConstantPAG + PAG + FAZ.N.V11REGPAGTwo ConstantPAG + PAG + FAZ.N.V11REGPAGTwo ConstantPAG + PAG + FAZ.N.V11REGPAGConstantPAG + PAG + FAZ.N.V11CRAPAGDecomentPAG + PAG + FAZ.N.V11DECPAGDecomentPAG + PAG + PAG + PAGZ.N.V11CRAPAGDecomentPAG + PAG + PAG + PAGZ.N.V11CRAPAGDecomentPAG + PAG +	SBCI	Rd, K	Subtract with Carry Constant from Reg.	$Rd \gets Rd - K - C$	Z,C,N,V,H	1
ANDRd, FLLogial AND Register and ConstantRd - Rd v KZ.N.V1ORRd, RLLogial OR Register and ConstantRd - Rd v KZ.N.V1ORIRd, KLLogial OR Register and ConstantRd - Rd v KZ.N.V1CMRd, NLLogial OR Register and ConstantRd - Rd v KZ.N.V1CMRdDev ComplementRd - Rd v KZ.N.V1CMRdDev ComplementRd - Rd v KZ.N.V1SRRd KSelfkin RegisterRd - Rd v KZ.N.V1SRRd KCanar Biol RegisterRd - Rd v Rd FF - RdZ.N.V1SRRd KCanar Biol RegisterRd - Rd v Rd FF - RdZ.N.V1SRRd KCanar Biol RegisterRd - Rd v Rd FF - RdZ.N.V1INCRd KCanar Biol RegisterRd - Rd v Rd Pd - Rd v Rd vX.N.V1SRRd KCanar Biol RegisterRd - Rd v Rd vRd vRd v1SRRd KCanar Biol RegisterRd v Rd vRd vRd v11SRRd KCanar Biol RegisterRd vRd vRd v11<	AND	Rd, Rr	Logical AND Registers	$Rd \leftarrow Rd ullet Rr$	Z,N,V	1
OR     Rol, Ru     Logical OR Register and Contain     Rid + Ray K     ZNV     1       DOR     Ro, Ru     Logical OR Register and Contain     Rid + Ray An     ZNV     1       EOR     Rod     Toy Complement     Rid + Ray An     ZONV     1       NEG     Rid     Toy Complement     Rid + SPT - Ray An     ZONV     1       SBR     Rid K     Dest Stript In Register     Rid + Rid + Ray     ZNV     1       DEG     Rid     Destembly In Register     Rid + Rid + Rid     ZNV     1       DEG     Rid     Destembly In Register     Rid + Rid + Rid     ZNV     1       DEG     Rid     Destembly In Register     Rid + Rid + Rid     ZNV     1       DEG     Rid     Destembly In Rid Rid     ZNV     1     1       DEG     Rid     Destembly In Rid	ANDI	Rd, K	Logical AND Register and Constant	$Rd \gets Rd \bullet K$	Z,N,V	1
OH     Rd, K     Logial OR Register     Rd - Rd VK     ZNV     1       EOR     Rd, K     Cody Conglement     Rd - Rd Pr.     ZNV     1       CM     Rd     Ons Conglement     Rd + S0 - Rd     ZCAVV     1       SR     Rd     SR Pr. Rd     SR Pr. Rd     ZCAVV     1       SR     Rd     SR Pr. Rd     SR Pr. Rd     ZCAVV     1       SR     SR Pr. Rd     SR Pr. Rd     SR Pr. Rd     ZAVV     1       SR     Case RB(s) Register     Rd + Rd + Rd + Rd - Rd     ZAV     1       DEC     Rd     SR Segeenet     Rd + Rd + Rd + Rd     ZAV     1       DEC     Rd     Set Register     Rd + Rd + Rd + Rd     ZAV     1       DER     Rd     Set Register     Rd + Rd	OR	Rd, Rr	Logical OR Registers	$Rd \leftarrow Rd \lor Rr$	Z,N,V	1
EOR     PA, Pr.     Exclusive OR Registers     Rd - R - R - R - R - S - R - R	ORI	Rd, K	Logical OR Register and Constant	$Rd \gets Rd \lor K$	Z,N,V	1
COM     Pid     Ore's Complement     Pid + SPT - Pid     ZCNV H     1       NEG     Rd     Two's Complement     Rd + SD - Rd / CXNV H     1       SRR     Rd, K     Stability in Register     Rd + Rd + Rd + ST     ZNV     1       SRR     Rd, K     Stability in Register     Rd + Rd + Rd + ST     ZNV     1       INC     Rd / Case Register     Rd + Rd	EOR	Rd, Rr	Exclusive OR Registers	$Rd \gets Rd \oplus Rr$	Z,N,V	1
NEG.     Rd     Two's Complement     Rd - Rd vK     Z/S/W     1       SBR     Rd K     Set B(g) in Register     Rd - Rd vK     ZNV     1       CBR     Rd K     Oser B(g) in Register     Rd - Rd + GFF - K)     ZNV     1       DEC     Rd     Decrement     Rd - Rd + GFF - K)     ZNV     1       DEC     Rd     Decrement     Rd - Rd + GF     ZNV     1       DEC     Rd     Cear Register     Rd - Rd + Rd     ZNV     1       SER     Rd     Cear Register     Rd - Fd + Rd     None     2       BRACH INSTRUTOTOTO     FC - FC + k + 1     None     2     None     3       IDALL     K     Rediave Jamp D(2)     PC - FC + k + 1     None     4       IDALL     Indrace Marking D(2)     PC - FC + k + 1     None     4       IDALL     Indrace Marking D(2)     PC - FC + k + 1     None     4       IDALL     Indrace Marking D(2)     PC - FC + k + 1     None     1       IDALL     Indrace Marking D(2)     PC - FC	СОМ	Rd	One's Complement	$Rd \leftarrow \$FF - Rd$	Z,C,N,V	1
SBR     Pit, K     Set Bit(s) in Register     Pit C = NV K     ZNV     1       CBR     Pit A     Cheer Bit(s) in Register     Pit C = NV K     XNV     1       INC     Pit A     Decrement     Pit C = Pit A     ZNV     1       DEC     Pit A     Decrement     Pit C = Pit A     ZNV     1       DEC     Pit A     Decrement     Pit C = Pit A     ZNV     1       SER     Rd     Set Register     Pit C = Pit A     Rd = Rd	NEG	Rd	Two's Complement	Rd ← \$00 – Rd	Z,C,N,V,H	1
GRN     Bt. K     Course Netgo in Register     Rd + Rd + Rd + FLA,     ZNV     1       INC     Rd     Decrement     Rd + Rd + 1     ZNV     1       DEC     Rd     Decrement     Rd + Rd + Rd - 1     ZNV     1       CLR     Rd     Clear Register     Rd + Rd + Rd - Rd     ZNV     1       StR     Rd     Star Register     Rd + Rd + Rd - Rd     ZNV     1       StR     Rd     Star Register     Rd + StF     None     1       StR     Rd     None     Rd - StF     None     2       RMMP     k     Relative Subroutine Call     PC + PC + k + 1     None     3       IDALL     Indirect Jump to (2)     PC + Z     None     3     1       ICALL     Indirect Jump to (2)     PC + STACK     None     1     4       CPL     Rd, Rr     Compare, Stor Mitional     If (Rd = R) PC + C2 + X + 1     None     1/2/3       CPL     Rd, Rr     Compare with Cary     Rd - Rr - C     ZNV C, H     1       CPL	SBR	Rd, K	Set Bit(s) in Register	$Rd \gets Rd \lor K$	Z,N,V	1
$ \begin{array}{cccccccccccccccccccccccccccccccccccc$	CBR	Rd, K	Clear Bit(s) in Register	$Rd \gets Rd \bullet (\$FF - K)$	Z,N,V	1
DECRdDecommentRd + Rd -1Z,N V1TSTRdTostor or MinusRd + Rel rRdZ,N V1CLRRdClear RegisterRd + Rel rRdZ,N V1SERRdSen RegisterRd + SFNone1SERRdSen RegisterRd + SFNone2RAMPkRelative JumpPC + PC + k + 1None2RAMPkRelative Jump 1C2PC + C-ZNone3RCALLkRelative Subroutine CallPC + PC + k + 1None3ICALLindirect Calls (Z)PC + C-ZNone3ICALLindirect Calls (Z)PC + STACKNone4RETIindirect Calls (Z)PC + STACKNone4RETIcompare, Skip if Equaliff Rd = Rip PC + PC + 2 or 3None1/2/3CPRd, RrCompare Nsip if Equaliff Rd = Rip PC + PC + 2 or 3None1/2/3CPRd, RrCompare Nsip if Equaliff Rhp = DC + PC + 2 or 3None1/2/3CPRd, RrCompare Nsip if Equaliff Rhp = DC + PC + 2 or 3None1/2/3SBSRb bSkip if Bit in Rogister Clearedif (Rhp) = DC + PC + 2 or 3None1/2/3SBSP, bSkip if Bit in Rogister Setif (Rhp) = DC + PC + 2 or 3None1/2/3SBSP, bSkip if Bit in Rogister Setif (Rhp) = DC + PC + 2 or 3None1/2/3SBSP, bSkip if Bit in Rogister Cleared	INC	Rd	Increment	$Rd \leftarrow Rd + 1$	Z,N,V	1
TST     Rd     Clear Degister     Rd - Rd + Rd     Z,NV     1       CLR     Rd     Gear Degister     Rd + Rd Rd     Z,NV     1       SER     Rd     SetRegister     Rd + SFF     None     1       BRANCHINSTRUCTIONE     Indirect Jump to [2]     PC +- PC + k + 1     None     2       LMP     k     Relative Jump (2]     PC +- 2C + k + 1     None     3       ICALL     k     Relative Jump to [2]     PC +- 2C + k + 1     None     3       ICALL     k     Relative Jump to [2]     PC +- 2C + k + 1     None     4       ICALL     k     Relative Jump to [2]     PC +- 2C + k + 1     None     4       ICALL     k     Relative Jump to [2]     PC +- 2C + k + 1     None     4       ICALL     k     Relative Jump to [2]     PC + 2C + 2C + 2C + 1     None     4       ICALL     k     Relative Jump to [2]     PC + 5TACK     I     4       CP     Rd, Rr     Compare Register with Innediater     Rd + Rr     Z,NV,C,H     1  <	DEC	Rd	Decrement	$Rd \leftarrow Rd - 1$	Z,N,V	1
CLR     Rd     Clear Register     Rd – Rd ® Rd     Z,N,V     1       SER     Rd     Set Register     Rd – SFF.     None     1       BRACH INSTRUCTORS       None     2       RAMP     k     Relative Jump to (2)     PC + PC + k + 1     None     2       RCALL     k     Relative Subroutine Call     PC - PC + k + 1     None     3       RET     Indirect Jump to (2)     PC - Z     None     4       CPL     Indirect Jump to (2)     PC - Z     None     4       CPL     Indirect Jump to (2)     PC - Z     None     4       CALL     Indirect Jump telum     PC - STACK     None     4       CPL     Rd, Rr     Compare Register with Carry     Rd - Rr - C     Z.N.V.C.H     1       CPC     Rd, Rr     Compare Register with Carry     Rd - K     Z.N.V.C.H     1       CPC     Rd, K     Compare Register Set     If (Rh(b) = 0) PC - PC + 2 or 3     None     1/2/3       SBRS     Rr, b     Skip if Bin hegester Geard <t< td=""><td>TST</td><td>Rd</td><td>Test for Zero or Minus</td><td><math display="block">Rd \gets Rd \bullet Rd</math></td><td>Z,N,V</td><td>1</td></t<>	TST	Rd	Test for Zero or Minus	$Rd \gets Rd \bullet Rd$	Z,N,V	1
SER     Rd     SetRegister     Rd = SFF     None     1       BRANCH INSTRUCTORS     BRANCH INSTRUCTORS     None     2       LIMP     k     Relative Jump (z)     PC + 2C     None     2       LIMP     indirect Jump to (z)     PC + 2C     None     3       RCALL     k     Relative Jump (Z)     PC + 2C + k + 1     None     3       ICALL     k     Relative Jump to (Z)     PC + 2C + k + 1     None     3       ICALL     k     Relative Jump to (Z)     PC - 2T ACK     None     4       ICAL     indirect Call to (Z)     PC - 2T ACK     None     1     4       CP     Rd, Rr     Compare, Skip IE gual     if (Rd = Ri) PC + PC + 2 or 3     None     1/23       CP     Rd, Rr     Compare Nito Carry     Rd - K     Z_NV, C, H     1       CP     Rd, Rr     Compare Nito Carry     Rd - K     Z_NV, C, H     1       SBR     Rr, b     Skip IB II In Register Clared     if (Rd(b) = 1) PC + PC + 2 or 3     None     1/2/3       SBR	CLR	Rd	Clear Register	$Rd \gets Rd \oplus Rd$	Z,N,V	1
BRANCH INSTRUCTIONS       IMMP     k     Relative Jump to (2)     PC +- PC + k + 1     None     2       LMP     Indirect Jump to (2)     PC +- Z     None     3       RCALL     k     Relative Subroutine Call     PC PC + k + 1     None     3       ICALL     Indirect Call to (2)     PC Z     None     3       ICALL     Indirect Call to (2)     PC Z     None     3       ICALL     Indirect Call to (2)     PC Z     None     4       ICALL     Interrupt Return     PC STACK     None     4       RETI     Subroutine Return     PC STACK     I     4       CPPE     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       SBRC     Rr, b     Skip If Bit in Register Cleared     If (Rh(b) = 1) PC - PC + 2 or 3     None     1/2/3       SBRS     Rr, b     Skip If Bit in NO Register Cleared     If (RD) = 1) PC - PC + 2 or 3     None     1/2/3	SER	Rd	Set Register	Rd ← \$FF	None	1
PLMP     k     Relative Jump     PC - PC + k + 1     None     2       IMP     Indirect Jump to [2]     PC + Z     None     3       ICALL     k     Relative Subroutine Call     PC + -Z     None     3       ICALL     -     Indirect Call to [2]     PC + -Z     None     3       ICALL     -     Subroutine Return     PC - STACK     None     4       RET     Subroutine Return     PC - STACK     I     4       CP     Rd, Rr     Compare Subj IE pual     If (Rd = R) PC - C 2 or 3     None     1/2/3       CPC     Rd, Rr     Compare Magister with Immediate     Rd - K     ZN,VC,H     1       CPC     Rd, Rr     Compare Register with Immediate     Rd - K     ZN,VC,H     1       SBRS     Rr, b     Skip If Bit Rin Register Cleared     If (R(b) = 1) PC + PC + 2 or 3     None     1/2/3       SBRS     P, b     Skip If Bit In IO Register Cleared     If (R(b) = 0) PC + PC + 2 or 3     None     1/2/3       SBRS     P, b     Skip If Bit In IO Register Gleared     If (RD(b) = 0	BRANCH INSTRUC	TIONS				
IMP     Indirect Jump to (2)     PC Z     None     2       RGALL     k     Relative Subroutine Call     PC PC + k + 1     None     3       ICALL     Indirect Call to (2)     PC FC + k + 1     None     3       RET     Subroutine Return     PC STACK     None     4       CPS     Rd, Rr     Compare Skip if Equal     if (Pd - Rn (PC - PC + 2 or 3)     None     1/2/3       CP     Rd, Rr     Compare Skip if Equal     if (Pd - Rn (PC - PC + 2 or 3)     None     1/2/3       CPC     Rd, Rr     Compare Min Carry     Rd - Rr - C     Z/N/C,H     1       CPC     Rd, Rr     Compare Min Carry     Rd - Rr - C     Z/N/C,H     1       CPC     Rd, K     Compare Min Carry     Rd - Rr - C     Z/N/C,H     1       SBRC     Rr, b     Skip if Bit in Register Cleared     if (Rrtip) = 0) PC - PC + 2 or 3     None     1/2/3       SBRS     Rr, b     Skip if Bit in Register is Set     if (R(h) = 1) PC - PC + 2 or 3     None     1/2/3       SBRS     P, b     Skip if Bit in Register is Set	RJMP	k	Relative Jump	$PC \leftarrow PC + k + 1$	None	2
RCALL     k     Relative Subroutine Call     PC C + k + 1     None     3       ICALL     Indired Call to (2)     PC Z     None     3       ICAL     Indired Call to (2)     PC STACK     None     4       RET     Interrupt Return     PC STACK     I     4       CP     Rd, Rr     Compare, Skip if Equal     If (Rd = Rr) PC - PC + 2 or 3     None     1/2/3       CP     Rd, Rr     Compare Might Equal     If (Rd = Rr) PC - PC + 2 or 3     None     1/2/3       CP     Rd, Rr     Compare Mighter with Immediate     Rd - K     Z.N.V.C.H     1       CP     Rd, Rr     Compare Mighter with Immediate     Rd - K     Z.N.V.C.H     1       SBRS     Rr, b     Skip if Bl in Register is Set     If (Rt)b = 1) PC + PC + 2 or 3     None     1/2/3       SBRS     P, b     Skip if Bl in I/O Register Is Set     If (Rb(b) = 1) PC + PC + 2 or 3     None     1/2/3       SBRS     P, b     Skip if Bl in I/O Register Is Set     If (Rb(b) = 1) PC + PC + 2 or 3     None     1/2/3       SBRS     P, b <td< td=""><td>IJMP</td><td></td><td>Indirect Jump to (Z)</td><td><math>PC \leftarrow Z</math></td><td>None</td><td>2</td></td<>	IJMP		Indirect Jump to (Z)	$PC \leftarrow Z$	None	2
ICALLIndirect Call to $(2)$ PC $- Z$ None3RETSubouline ReturnPC $- STACK$ None4RET1Interrupt ReturnPC $- STACK$ I4CPSRd, RrCompare, Skip I Equalif (Rd = R) PC $-$ PC $+ 2 \sigma$ 3None1/23CPRd, RrCompare Skip I Equalif (Rd = R) PC $-$ PC $+ 2 \sigma$ 3None1/23CPRd, RrCompare Negister with CarryRd $-$ Rr $-$ CZ.N.V.C.H1CP1Rd, KCompare Register with ImmediateRd $-$ KrZ.N.V.C.H1CP1Rd, KCompare Register with ImmediateRd $-$ KrZ.N.V.C.H1SBRCRr, bSkip I Bt In Register I Clearedif (Rt(b) = 0) PC $+$ PC $+ 2 \sigma$ 3None1/23SBRSRr, bSkip I Bt In In Register I Setif (Rt(b) = 0) PC $+$ PC $+ 2 \sigma$ 3None1/23SBISP, bSkip I Bt In In Register I Setif (Rt(b) = 1) PC $+$ PC $+ 2 \sigma$ 3None1/23SBRSRr, bStarch I Status Flag Setif (SREG(s) = 1) then PC $+$ PC $+ k + 1$ None1/2BRBSs, kBranch I Status Flag Setif (SREG(s) = 1) then PC $+$ PC $+ k + 1$ None1/2BRECs, kBranch I Kotegualif (Z = 1) then PC $+$ PC $+ k + 1$ None1/2BRECkBranch I floadif (Z = 0) then PC $+$ PC $+ k + 1$ None1/2BRECkBranch I floadif (Z = 0) then PC $+$ PC $+ k + 1$ None1/2BRECkBranch I f	RCALL	k	Relative Subroutine Call	$PC \leftarrow PC + k + 1$	None	3
HETSubroutine ReturnPC - STACKNone4HETIInterrupt ReturnPC - STACKI4CPSEPtd, RrCompare, Skip if Equalif (Rd = Rr) PC + PC + 2 or 3None112/3CPRd, RrCompare Wit CarryRd - RrZ,N,V,C,H1CPCRd, RrCompare Wit CarryRd - Rr - CZ,N,V,C,H1CPLRd, KCompare Negaster with ImmediateRd - KrZ,N,V,C,H1CPLRd, KCompare Negaster with ImmediateRd - KrZ,N,V,C,H1SBRSRr, bSkip if Bit n Register is Setif (R(b) = 1)PC + PC + 2 or 3None11/2/3SBRSP, bSkip if Bit n Register is Setif (R(b) = 1)PC + PC + 2 or 3None11/2/3SBRSP, bSkip if Bit n I/O Register Clearedif (P(b) = 0) PC + PC + 2 or 3None11/2/3SBRSS, kBranch f Status Flag Setif (SREG(s) = 0) then PC - PC + k + 1None11/2BRBCs, kBranch if Status Flag Setif (SREG(s) = 0) then PC - PC + k + 1None11/2BRBCkBranch if Equalif (Z=1) then PC - PC + k + 1None11/2BRCkBranch if Carry Setif (C = 0) then PC + PC + k + 1None11/2BRCkBranch if Carry Clearedif (C = 0) then PC + PC + k + 1None11/2BRCkBranch if Set or Equal, Signedif (N = 0) then PC + PC + k + 1None11/2BRCkBranch if Garry Geard	ICALL		Indirect Call to (Z)	$PC \leftarrow Z$	None	3
HETIInterrupt ReturnPC - STACKI4CPSERd, RrCompare, Skip if Equalif (Rd = Rr) PC + PC + 2 or 3None1/2/3CPRd, RrCompareRd - RrZ,N,V,C,H1CPCRd, RrCompare with CarryRd - Rr - CZ,N,V,C,H1CPIRd, KCompare Register with ImmediateRd - KZ,N,V,C,H1SBRCRr, bSkip if Bit in Register Clearedif (Rr(b) = 0) PC + PC + 2 or 3None1/2/3SBRSRr, bSkip if Bit in Register Clearedif (Rh(b) = 1) PC + PC + 2 or 3None1/2/3SBRSP, bSkip if Bit in 1/O Register Setif (Rh(b) = 1) PC + PC + 2 or 3None1/2/3SBRSS, kBranch if Status Flag Setif (Rb(b) = 1) PC + PC + 2 or 3None1/2/3BRSs, kBranch if Status Flag Setif (Rb(b) = 1) PC + PC + 2 or 3None1/2/3BRSs, kBranch if Status Flag Clearedif (Rb(c) = 1) PC + PC + 2 or 3None1/2/3BRSs, kBranch if Status Flag Clearedif (SREG(s) = 1) then PC + PC + k + 1None1/2BRSs, kBranch if Status Flag Clearedif (SREG(s) = 0) then PC + PC + k + 1None1/2BRSkBranch if Carry Setif (C = 0) then PC + PC + k + 1None1/2BRSkBranch if Carry Clearedif (C = 0) then PC + PC + k + 1None1/2BRCkBranch if Carry Clearedif (C = 0) then PC + PC + k + 1None1/2	RET		Subroutine Return	$PC \leftarrow STACK$	None	4
CPSERd, RrCompare, Skip if Equalif (Rd = Rr) PC $\leftarrow$ PC + 2 or 3None1/2/3CPRd, RrCompare with CarryRd - RrZ,N,V,C,H1CPCRd, RrCompare with CarryRd - Rr - CZ,N,V,C,H1CPIRd, KCompare Register with ImmediateRd - KZ,N,V,C,H1SBRCRr, bSkip if Bit In Register Clearedif (Rr(b) = 0) PC $\leftarrow$ PC + 2 or 3None1/2/3SBRSRr, bSkip if Bit In Register Clearedif (Rr(b) = 1) PC $\leftarrow$ PC + 2 or 3None1/2/3SBRCP, bSkip if Bit In UD Register Clearedif (R(b) = 1) PC $\leftarrow$ PC + 2 or 3None1/2/3SBRSR, kBranch if Status Flag Setif (R(b) = 1) PC $\leftarrow$ PC + 2 or 3None1/2/3BRBSs, kBranch if Status Flag Clearedif (SREG(s) = 0) then PC $\leftarrow$ PC + k + 1None1/2BRBCs, kBranch if Status Flag Clearedif (SREG(s) = 0) then PC $\leftarrow$ PC + k + 1None1/2BREDkBranch if Not Equalif (Z = 0) then PC $\leftarrow$ PC + k + 1None1/2BRNEkBranch if Carry Setif (C = 0) then PC $\leftarrow$ PC + k + 1None1/2BRCkBranch if Carry Setif (C = 0) then PC $\leftarrow$ PC + k + 1None1/2BRCkBranch if Carry Clearedif (C = 0) then PC $\leftarrow$ PC + k + 1None1/2BRCkBranch if Gauser or Equal, Signedif (N = 0) then PC $\leftarrow$ PC + k + 1None1/2BRHkBranch if Gauser or Eq	RETI		Interrupt Return	$PC \leftarrow STACK$	1	4
CPRd, RrCompareRd - RrZ,N,V,C,H1CPCRd, RrCompare Register with ImmediateRd - Rr - CZ,N,V,C,H1CPIRd, KCompare Register with ImmediateRd - KZ,N,V,C,H1SBRCRr, bSkip if Bit in Register Clearedif (Rr(b) = 0) PC + PC + 2 or 3None1/2/3SBRSRr, bSkip if Bit in Negister Clearedif (Rr(b) = 0) PC + PC + 2 or 3None1/2/3SBRSP, bSkip if Bit in IO Register Clearedif (Rr(b) = 1) PC + PC + 2 or 3None1/2/3SBRSP, bSkip if Bit in IO Register Clearedif (RRG(s) = 1) then PC + PC + 2 or 3None1/2/3BRBSs, kBranch if Status Flag Setif (SREG(s) = 0) then PC + PC + k+1None1/2BRBCs, kBranch if Status Flag Clearedif (SREG(s) = 0) then PC + PC + k+1None1/2BREQkBranch if Equalif (Z = 0) then PC + PC + k+1None1/2BRCSkBranch if Garry Setif (C = 0) then PC + PC + k+1None1/2BRCCkBranch if Garry Setif (C = 0) then PC + PC + k+1None1/2BRCLkBranch if Same or Higherif (C = 0) then PC + PC + k+1None1/2BRLkBranch if Algeneif (N = 0) then PC + PC + k+1None1/2BRHkBranch if Gareer or Equal, Signedif (N = 0) then PC + PC + k+1None1/2BRHkBranch if Hallif (N = 0) then PC + PC + k+1 <td< td=""><td>CPSE</td><td>Rd, Rr</td><td>Compare, Skip if Equal</td><td>if (Rd = Rr) PC <math>\leftarrow</math> PC + 2 or 3</td><td>None</td><td>1/2/3</td></td<>	CPSE	Rd, Rr	Compare, Skip if Equal	if (Rd = Rr) PC $\leftarrow$ PC + 2 or 3	None	1/2/3
CPCRd, RrCompare with CarryRd - Rr - CZ,N,V,C,H1CPIRd, KCompare Register with ImmediateRd - KZ,N,V,C,H1SBRCRr, bSkip if Bit in Register Clearedif (Rr(b) = 1) PC $\leftarrow$ PC + 2 or 3None1/2/3SBRSRr, bSkip if Bit in Register Clearedif (Rr(b) = 1) PC $\leftarrow$ PC + 2 or 3None1/2/3SBRCP, bSkip if Bit in /O Register is Setif (Rtb) = 0) PC $\leftarrow$ PC + 2 or 3None1/2/3BRSS, kBranch if Status Flag Setif (Rtb) = 1) PC $\leftarrow$ PC + 2 or 3None1/2/3BRSs, kBranch if Status Flag Clearedif (Rtb) = 1) PC $\leftarrow$ PC + 2 or 3None1/2BRBSs, kBranch if Status Flag Clearedif (SREG(s) = 1) then PC $\leftarrow$ PC + k + 1None1/2BRBCs, kBranch if Status Flag Clearedif (Z = 1) then PC $\leftarrow$ PC + k + 1None1/2BRCkBranch if Carry Setif (Z = 0) then PC $\leftarrow$ PC + k + 1None1/2BRCSkBranch if Carry Setif (C = 0) then PC $\leftarrow$ PC + k + 1None1/2BRCCkBranch if Carry Glearedif (C = 0) then PC $\leftarrow$ PC + k + 1None1/2BRSHkBranch if Carry Setif (C = 0) then PC $\leftarrow$ PC + k + 1None1/2BRCkBranch if Carry Setif (C = 0) then PC $\leftarrow$ PC + k + 1None1/2BRSHkBranch if Carry Setif (N = 0) then PC $\leftarrow$ PC + k + 1None1/2BRSHkBranch if Carry Se	CP	Rd, Rr	Compare	Rd – Rr	Z,N,V,C,H	1
CPIRd, KCompare Register with ImmediateRd - KZ,Ny,C,H1SBRCRr, bSkip if Bit in Register Clearedif (Rr(b) = 0) PC + PC + 2 or 3None11/23SBRSRr, bSkip if Bit in Register Clearedif (Rr(b) = 1) PC + PC + 2 or 3None11/23SBICP, bSkip if Bit in 1/O Register Clearedif (P(b) = 0) PC + PC + 2 or 3None11/23SBISP, bSkip if Bit in 1/O Register Clearedif (P(b) = 0) PC + PC + 2 or 3None11/23BRBSs, kBranch if Status Flag Setif (SREG(s) = 1) then PC - PC + k + 1None11/2BRBCs, kBranch if Status Flag Setif (SREG(s) = 0) then PC - PC + k + 1None11/2BRBCkBranch if Status Flag Clearedif (Z = 1) then PC - PC + k + 1None11/2BRCSkBranch if Status Flag Clearedif (Z = 0) then PC - PC + k + 1None11/2BRCCkBranch if Carry Clearedif (C = 0) then PC - PC + k + 1None11/2BRCCkBranch if Same or Higherif (C = 0) then PC + PC + k + 1None11/2BRLDkBranch if Same or Higherif (N = 0) then PC + PC + k + 1None11/2BRHkBranch if ILowerif (N = 0) then PC + PC + k + 1None11/2BRLkBranch if Greater or Equal, Signedif (N = 0 = 0) then PC + PC + k + 1None11/2BRHkBranch if Greater or Equal, Signedif (N = 0 = 0) then PC + PC + k + 1None11	CPC	Rd, Rr	Compare with Carry	Rd – Rr – C	Z,N,V,C,H	1
SBRCRr, bSkip if Bit in Register Clearedif (Rr(b) = 0) PC $\leftarrow$ PC + 2 or 3None1/2/3SBRSRr, bSkip if Bit in Register is Setif (Rr(b) = 1) PC $\leftarrow$ PC + 2 or 3None1/2/3SBICP, bSkip if Bit in I/O Register Clearedif (P(b) = 0) PC $\leftarrow$ PC + 2 or 3None1/2/3SBISP, bSkip if Bit in I/O Register is Setif (R(b) = 1) PC $\leftarrow$ PC + 2 or 3None1/2/3BRBSs, kBranch if Status Flag Setif (SREG(g) = 1) then PC $\leftarrow$ PC + k + 1None1/2BRBCs, kBranch if Status Flag Clearedif (SREG(g) = 0) then PC $\leftarrow$ PC + k + 1None1/2BREQkBranch if I Equalif (Z = 0) then PC $\leftarrow$ PC + k + 1None1/2BRNEkBranch if I Mot Equalif (Z = 0) then PC $\leftarrow$ PC + k + 1None1/2BRNEkBranch if I derry Setif (C = 0) then PC $\leftarrow$ PC + k + 1None1/2BRCCkBranch if Carry Clearedif (C = 0) then PC $\leftarrow$ PC + k + 1None1/2BRCLkBranch if Garry Clearedif (C = 0) then PC $\leftarrow$ PC + k + 1None1/2BRCLkBranch if Garry Clearedif (N = 0) then PC $\leftarrow$ PC + k + 1None1/2BRSHkBranch if I Garry Clearedif (N = 0) then PC $\leftarrow$ PC + k + 1None1/2BRUDkBranch if Garry Clearedif (N = 0) then PC $\leftarrow$ PC + k + 1None1/2BRLkBranch if Greater or Equal, Signedif (N = 0) then PC $\leftarrow$ PC + k + 1None <td< td=""><td>CPI</td><td>Rd, K</td><td>Compare Register with Immediate</td><td>Rd – K</td><td>Z,N,V,C,H</td><td>1</td></td<>	CPI	Rd, K	Compare Register with Immediate	Rd – K	Z,N,V,C,H	1
SBRSRr, bSkip if Bit in Register is Setif (Rr(b) = 1) PC $\leftarrow$ PC + 2 or 3None12/3SBICP, bSkip if Bit in I/O Register Clearedif (P(b) = 0) PC $\leftarrow$ PC + 2 or 3None11/2/3SBISP, bSkip if Bit in I/O Register is Setif (P(b) = 1) PC $\leftarrow$ PC + 2 or 3None11/2/3BRBSs, kBranch if Status Flag Clearedif (SREG(s) = 1) then PC $\leftarrow$ PC + k + 1None11/2BRBCs, kBranch if Status Flag Clearedif (Z = 0) then PC $\leftarrow$ PC + k + 1None11/2BREQkBranch if Not Equalif (Z = 0) then PC $\leftarrow$ PC + k + 1None11/2BRCSkBranch if Carry Setif (C = 0) then PC $\leftarrow$ PC + k + 1None11/2BRCCkBranch if Carry Clearedif (C = 0) then PC $\leftarrow$ PC + k + 1None11/2BRCLkBranch if Carry Clearedif (C = 0) then PC $\leftarrow$ PC + k + 1None11/2BRCLkBranch if Munusif (N = 0) then PC $\leftarrow$ PC + k + 1None11/2BRCLkBranch if Munusif (N = 0) then PC $\leftarrow$ PC + k + 1None11/2BRHkBranch if Munusif (N = 0) then PC $\leftarrow$ PC + k + 1None11/2BRHkBranch if Greater or Equal, Signedif (N = 0) then PC $\leftarrow$ PC + k + 1None11/2BRHkBranch if Munusif (N = 0) then PC $\leftarrow$ PC + k + 1None11/2BRHkBranch if Greater or Equal, Signedif (N = 0) then PC $\leftarrow$ PC + k + 1None11/2BRH </td <td>SBRC</td> <td>Rr, b</td> <td>Skip if Bit in Register Cleared</td> <td>if <math>(Rr(b) = 0) PC \leftarrow PC + 2 \text{ or } 3</math></td> <td>None</td> <td>1/2/3</td>	SBRC	Rr, b	Skip if Bit in Register Cleared	if $(Rr(b) = 0) PC \leftarrow PC + 2 \text{ or } 3$	None	1/2/3
SBICP, bSkip if Bit in I/O Register Clearedif $(P(b) = 0) PC \leftarrow PC + 2 or 3$ None1/2/3SBISP, bSkip if Bit in I/O Register is Setif $(R(b) = 1) PC \leftarrow PC + 2 or 3$ None1/2/3BRBSs, kBranch if Status Flag Setif $(R(b) = 1) PC \leftarrow PC + k + 1$ None1/2BRBCs, kBranch if Status Flag Setif $(SREG(s) = 1)$ then $PC \leftarrow PC + k + 1$ None1/2BREQkBranch if fot Equalif $(Z = 1)$ then $PC \leftarrow PC + k + 1$ None1/2BRNEkBranch if Ot Equalif $(Z = 0)$ then $PC \leftarrow PC + k + 1$ None1/2BRCCkBranch if Carry Setif $(C = 1)$ then $PC \leftarrow PC + k + 1$ None1/2BRCCkBranch if Garry Clearedif $(C = 0)$ then $PC \leftarrow PC + k + 1$ None1/2BRSHkBranch if Minusif $(C = 1)$ then $PC \leftarrow PC + k + 1$ None1/2BRLOkBranch if Minusif $(C = 1)$ then $PC \leftarrow PC + k + 1$ None1/2BRLDkBranch if Minusif $(N = 1)$ then $PC \leftarrow PC + k + 1$ None1/2BRPLkBranch if Minusif $(N = 0)$ then $PC \leftarrow PC + k + 1$ None1/2BRGEkBranch if Greater or Equal, Signedif $(N \oplus V = 0)$ then $PC \leftarrow PC + k + 1$ None1/2BRHZkBranch if Idees Than Zero, Signedif $(N \oplus V = 1)$ then $PC \leftarrow PC + k + 1$ None1/2BRHCkBranch if Half-carry Flag Setif $(N \oplus V = 1)$ then $PC \leftarrow PC + k + 1$ None1/2B	SBRS	Rr, b	Skip if Bit in Register is Set	if $(Rr(b) = 1) PC \leftarrow PC + 2 \text{ or } 3$	None	1/2/3
SBISP, bSkip if Bit in I/O Register is Setif (R(b) = 1) PC - PC + 2 or 3None1/2/3BRBSs, kBranch if Status Flag Setif (SREG(s) = 1) then PC - PC + k + 1None1/2BRBCs, kBranch if Status Flag Clearedif (SREG(s) = 0) then PC - PC + k + 1None1/2BREQkBranch if Status Flag Clearedif (Z = 1) then PC - PC + k + 1None1/2BRNEkBranch if Not Equalif (Z = 0) then PC - PC + k + 1None1/2BRCSkBranch if Carry Setif (C = 0) then PC - PC + k + 1None1/2BRCCkBranch if Carry Clearedif (C = 0) then PC - PC + k + 1None1/2BRLOkBranch if Carry Clearedif (C = 0) then PC - PC + k + 1None1/2BRLOkBranch if Minusif (N = 1) then PC - PC + k + 1None1/2BRHIkBranch if Graeter or Equal, Signedif (N = 0) then PC - PC + k + 1None1/2BRPLkBranch if Graeter or Equal, Signedif (N = 0) then PC - PC + k + 1None1/2BRQEkBranch if Graeter or Equal, Signedif (N = 0) then PC - PC + k + 1None1/2BRHSkBranch if Half-carry Flag Setif (N = 0) then PC - PC + k + 1None1/2BRHCkBranch if Half-carry Flag Clearedif (N = V = 0) then PC - PC + k + 1None1/2BRHSkBranch if Half-carry Flag Clearedif (N = V = 0) then PC - PC + k + 1None1/2	SBIC	P, b	Skip if Bit in I/O Register Cleared	if $(P(b) = 0) PC \leftarrow PC + 2 \text{ or } 3$	None	1/2/3
BRBSs, kBranch if Status Flag Setif (SREG(s) = 1) then PC $\leftarrow$ PC + k + 1None1/2BRBCs, kBranch if Status Flag Clearedif (SREG(s) = 0) then PC $\leftarrow$ PC + k + 1None1/2BREQkBranch if Equalif (Z = 1) then PC $\leftarrow$ PC + k + 1None1/2BRNEkBranch if Not Equalif (Z = 0) then PC $\leftarrow$ PC + k + 1None1/2BRCSkBranch if Carry Setif (C = 0) then PC $\leftarrow$ PC + k + 1None1/2BRCCkBranch if Carry Clearedif (C = 0) then PC $\leftarrow$ PC + k + 1None1/2BRSHkBranch if Lowerif (C = 0) then PC $\leftarrow$ PC + k + 1None1/2BRLOkBranch if Lowerif (C = 0) then PC $\leftarrow$ PC + k + 1None1/2BRNLkBranch if Minusif (N = 1) then PC $\leftarrow$ PC + k + 1None1/2BRL0kBranch if Minusif (N = 0) then PC $\leftarrow$ PC + k + 1None1/2BRALkBranch if Greater or Equal, Signedif (N = 0) then PC $\leftarrow$ PC + k + 1None1/2BRALkBranch if Greater or Equal, Signedif (N = 0) then PC $\leftarrow$ PC + k + 1None1/2BRGEkBranch if Half-carry Flag Setif (N = 0) then PC $\leftarrow$ PC + k + 1None1/2BRLTkBranch if Half-carry Flag Setif (H = 1) then PC $\leftarrow$ PC + k + 1None1/2BRHSkBranch if T-flag Setif (H = 0) then PC $\leftarrow$ PC + k + 1None1/2BRHCkBranch if T-flag Setif	SBIS	P, b	Skip if Bit in I/O Register is Set	if $(R(b) = 1) PC \leftarrow PC + 2 \text{ or } 3$	None	1/2/3
BRBCs, kBranch if Status Flag Clearedif (SREG(s) = 0) then PC $\leftarrow$ PC + k + 1None1/2BREQkBranch if Equalif (Z = 1) then PC $\leftarrow$ PC + k + 1None1/2BRNEkBranch if Not Equalif (Z = 0) then PC $\leftarrow$ PC + k + 1None1/2BRCSkBranch if Carry Setif (C = 1) then PC $\leftarrow$ PC + k + 1None1/2BRCCkBranch if Carry Clearedif (C = 0) then PC $\leftarrow$ PC + k + 1None1/2BRSHkBranch if Same or Higherif (C = 0) then PC $\leftarrow$ PC + k + 1None1/2BRLOkBranch if Minusif (C = 1) then PC $\leftarrow$ PC + k + 1None1/2BRLUkBranch if Minusif (N = 0) then PC $\leftarrow$ PC + k + 1None1/2BRLUkBranch if Minusif (N = 0) then PC $\leftarrow$ PC + k + 1None1/2BRLUkBranch if Minusif (N = 0) then PC $\leftarrow$ PC + k + 1None1/2BRLTkBranch if Greater or Equal, Signedif (N = 0) then PC $\leftarrow$ PC + k + 1None1/2BRHSkBranch if Half-carry Flag Setif (N = 0) then PC $\leftarrow$ PC + k + 1None1/2BRHSkBranch if T-flag Setif (N = 0) then PC $\leftarrow$ PC + k + 1None1/2BRHSkBranch if T-flag Clearedif (T = 0) then PC $\leftarrow$ PC + k + 1None1/2BRHSkBranch if T-flag Setif (T = 0) then PC $\leftarrow$ PC + k + 1None1/2BRHSkBranch if T-flag Setif (T = 0) then PC $\leftarrow$ PC + k +	BRBS	s, k	Branch if Status Flag Set	if (SREG(s) = 1) then PC $\leftarrow$ PC + k + 1	None	1/2
BREQkBranch if Equalif $(Z = 1)$ then $PC \leftarrow PC + k + 1$ None1/2BRNEkBranch if Not Equalif $(Z = 0)$ then $PC \leftarrow PC + k + 1$ None1/2BRCSkBranch if Carry Setif $(C = 1)$ then $PC \leftarrow PC + k + 1$ None1/2BRCCkBranch if Carry Clearedif $(C = 0)$ then $PC \leftarrow PC + k + 1$ None1/2BRSHkBranch if Same or Higherif $(C = 0)$ then $PC \leftarrow PC + k + 1$ None1/2BRLOkBranch if Lowerif $(C = 1)$ then $PC \leftarrow PC + k + 1$ None1/2BRMIkBranch if Minusif $(N = 1)$ then $PC \leftarrow PC + k + 1$ None1/2BRPLkBranch if Greater or Equal, Signedif $(N = 0)$ then $PC \leftarrow PC + k + 1$ None1/2BRECkBranch if Greater or Equal, Signedif $(N \oplus V = 0)$ then $PC \leftarrow PC + k + 1$ None1/2BRLTkBranch if Half-carry Flag Setif $(N \oplus V = 0)$ then $PC \leftarrow PC + k + 1$ None1/2BRHSkBranch if Half-carry Flag Clearedif $(H = 0)$ then $PC \leftarrow PC + k + 1$ None1/2BRHCkBranch if Half-carry Flag Clearedif $(T = 0)$ then $PC \leftarrow PC + k + 1$ None1/2BRTSkBranch if T-flag Clearedif $(T = 0)$ then $PC \leftarrow PC + k + 1$ None1/2BRVCkBranch if Jeard Setif $(V = 0)$ then $PC \leftarrow PC + k + 1$ None1/2BRVCkBranch if Interrupt Enabledif $(V = 0)$ then $PC \leftarrow PC + k + 1$ None1/2BRID <td>BRBC</td> <td>s, k</td> <td>Branch if Status Flag Cleared</td> <td>if (SREG(s) = 0) then <math>PC \leftarrow PC + k + 1</math></td> <td>None</td> <td>1/2</td>	BRBC	s, k	Branch if Status Flag Cleared	if (SREG(s) = 0) then $PC \leftarrow PC + k + 1$	None	1/2
BRNEkBranch if Not Equalif (Z = 0) then PC $\leftarrow$ PC + k + 1None1/2BRCSkBranch if Carry Setif (C = 1) then PC $\leftarrow$ PC + k + 1None1/2BRCCkBranch if Carry Clearedif (C = 0) then PC $\leftarrow$ PC + k + 1None1/2BRSHkBranch if Same or Higherif (C = 0) then PC $\leftarrow$ PC + k + 1None1/2BRLOkBranch if Lowerif (C = 1) then PC $\leftarrow$ PC + k + 1None1/2BRMIkBranch if Minusif (N = 1) then PC $\leftarrow$ PC + k + 1None1/2BRLDkBranch if Greater or Equal, Signedif (N = 0) then PC $\leftarrow$ PC + k + 1None1/2BREkBranch if Greater or Equal, Signedif (N = 0) then PC $\leftarrow$ PC + k + 1None1/2BRGEkBranch if Half-carry Flag Setif (N = V = 0) then PC $\leftarrow$ PC + k + 1None1/2BRHSkBranch if Half-carry Flag Setif (N = V = 1) then PC $\leftarrow$ PC + k + 1None1/2BRHSkBranch if Half-carry Flag Clearedif (N = V = 1) then PC $\leftarrow$ PC + k + 1None1/2BRHSkBranch if Half-carry Flag Clearedif (H = 0) then PC $\leftarrow$ PC + k + 1None1/2BRTSkBranch if T-flag Setif (T = 0) then PC $\leftarrow$ PC + k + 1None1/2BRTCkBranch if T-flag Setif (V = 1) then PC $\leftarrow$ PC + k + 1None1/2BRVSkBranch if Overflow Flag is Setif (V = 0) then PC $\leftarrow$ PC + k + 1None1/2BRVCk <t< td=""><td>BREQ</td><td>k</td><td>Branch if Equal</td><td>if (Z = 1) then PC <math>\leftarrow</math> PC + k + 1</td><td>None</td><td>1/2</td></t<>	BREQ	k	Branch if Equal	if (Z = 1) then PC $\leftarrow$ PC + k + 1	None	1/2
BRCSkBranch if Carry Setif (C = 1) then PC $\leftarrow$ PC + k + 1None1/2BRCCkBranch if Carry Clearedif (C = 0) then PC $\leftarrow$ PC + k + 1None1/2BRSHkBranch if Same or Higherif (C = 0) then PC $\leftarrow$ PC + k + 1None1/2BRLOkBranch if Lowerif (C = 1) then PC $\leftarrow$ PC + k + 1None1/2BRMIkBranch if Minusif (N = 1) then PC $\leftarrow$ PC + k + 1None1/2BREkBranch if Plusif (N = 0) then PC $\leftarrow$ PC + k + 1None1/2BREkBranch if Greater or Equal, Signedif (N = 0) then PC $\leftarrow$ PC + k + 1None1/2BRLTkBranch if Hufscarry Flag Setif (N $\oplus$ V = 0) then PC $\leftarrow$ PC + k + 1None1/2BRHSkBranch if Half-carry Flag Setif (N $\oplus$ V = 1) then PC $\leftarrow$ PC + k + 1None1/2BRHCkBranch if Half-carry Flag Setif (H = 1) then PC $\leftarrow$ PC + k + 1None1/2BRTSkBranch if T-flag Setif (T = 0) then PC $\leftarrow$ PC + k + 1None1/2BRTCkBranch if T-flag Clearedif (T = 0) then PC $\leftarrow$ PC + k + 1None1/2BRVSkBranch if Overflow Flag is Setif (V = 0) then PC $\leftarrow$ PC + k + 1None1/2BRVCkBranch if Overflow Flag is Clearedif (V = 0) then PC $\leftarrow$ PC + k + 1None1/2BRUCkBranch if Interrupt Enabledif (V = 0) then PC $\leftarrow$ PC + k + 1None1/2BRUCkBranch if I	BRNE	k	Branch if Not Equal	if (Z = 0) then PC $\leftarrow$ PC + k + 1	None	1/2
BRCCkBranch if Carry Clearedif (C = 0) then PC $\leftarrow$ PC + k + 1None1/2BRSHkBranch if Same or Higherif (C = 0) then PC $\leftarrow$ PC + k + 1None1/2BRLOkBranch if Lowerif (C = 1) then PC $\leftarrow$ PC + k + 1None1/2BRMIkBranch if Minusif (N = 1) then PC $\leftarrow$ PC + k + 1None1/2BRPLkBranch if Plusif (N = 0) then PC $\leftarrow$ PC + k + 1None1/2BRGEkBranch if Greater or Equal, Signedif (N $\oplus$ V = 0) then PC $\leftarrow$ PC + k + 1None1/2BRLTkBranch if Less Than Zero, Signedif (N $\oplus$ V = 0) then PC $\leftarrow$ PC + k + 1None1/2BRHSkBranch if Half-carry Flag Setif (H = 1) then PC $\leftarrow$ PC + k + 1None1/2BRHCkBranch if T-flag Setif (T = 1) then PC $\leftarrow$ PC + k + 1None1/2BRTSkBranch if T-flag Setif (T = 0) then PC $\leftarrow$ PC + k + 1None1/2BRTCkBranch if T-flag Setif (T = 0) then PC $\leftarrow$ PC + k + 1None1/2BRVSkBranch if Overflow Flag is Setif (V = 1) then PC $\leftarrow$ PC + k + 1None1/2BRVCkBranch if Interrupt Enabledif (V = 0) then PC $\leftarrow$ PC + k + 1None1/2BRUCkBranch if Interrupt Enabledif (U = 0) then PC $\leftarrow$ PC + k + 1None1/2BRUCkBranch if Interrupt Enabledif (U = 0) then PC $\leftarrow$ PC + k + 1None1/2BRUCkBranch if Interru	BRCS	k	Branch if Carry Set	if (C = 1) then $PC \leftarrow PC + k + 1$	None	1/2
BRSHkBranch if Same or Higherif (C = 0) then PC $\leftarrow$ PC + k + 1None1/2BRLOkBranch if Lowerif (C = 1) then PC $\leftarrow$ PC + k + 1None1/2BRMIkBranch if Minusif (N = 1) then PC $\leftarrow$ PC + k + 1None1/2BRPLkBranch if Greater or Equal, Signedif (N = 0) then PC $\leftarrow$ PC + k + 1None1/2BRLTkBranch if Less Than Zero, Signedif (N $\oplus$ V = 0) then PC $\leftarrow$ PC + k + 1None1/2BRHSkBranch if Half-carry Flag Setif (H = 1) then PC $\leftarrow$ PC + k + 1None1/2BRHCkBranch if Half-carry Flag Clearedif (H = 0) then PC $\leftarrow$ PC + k + 1None1/2BRTSkBranch if T-flag Setif (T = 1) then PC $\leftarrow$ PC + k + 1None1/2BRTCkBranch if T-flag Setif (T = 0) then PC $\leftarrow$ PC + k + 1None1/2BRVSkBranch if Overflow Flag is Setif (Y = 1) then PC $\leftarrow$ PC + k + 1None1/2BRVCkBranch if Overflow Flag is Clearedif (Y = 0) then PC $\leftarrow$ PC + k + 1None1/2BRUCkBranch if Interrupt Enabledif (Y = 0) then PC $\leftarrow$ PC + k + 1None1/2BRUCkBranch if Interrupt Enabledif (U = 0) then PC $\leftarrow$ PC + k + 1None1/2BRUCkBranch if Interrupt Enabledif (U = 0) then PC $\leftarrow$ PC + k + 1None1/2BRUCkBranch if Interrupt Enabledif (U = 0) then PC $\leftarrow$ PC + k + 1None1/2 <tr <td=""></tr>	BRCC	k	Branch if Carry Cleared	if (C = 0) then $PC \leftarrow PC + k + 1$	None	1/2
BRLOkBranch if Lowerif $(C = 1)$ then $PC \leftarrow PC + k + 1$ None1/2BRMIkBranch if Minusif $(N = 1)$ then $PC \leftarrow PC + k + 1$ None1/2BRPLkBranch if Plusif $(N = 0)$ then $PC \leftarrow PC + k + 1$ None1/2BRGEkBranch if Greater or Equal, Signedif $(N \oplus V = 0)$ then $PC \leftarrow PC + k + 1$ None1/2BRLTkBranch if Less Than Zero, Signedif $(N \oplus V = 0)$ then $PC \leftarrow PC + k + 1$ None1/2BRHSkBranch if Half-carry Flag Setif $(H = 1)$ then $PC \leftarrow PC + k + 1$ None1/2BRHCkBranch if Half-carry Flag Clearedif $(H = 0)$ then $PC \leftarrow PC + k + 1$ None1/2BRTSkBranch if T-flag Setif $(T = 1)$ then $PC \leftarrow PC + k + 1$ None1/2BRTCkBranch if T-flag Clearedif $(T = 0)$ then $PC \leftarrow PC + k + 1$ None1/2BRTCkBranch if Overflow Flag is Setif $(V = 1)$ then $PC \leftarrow PC + k + 1$ None1/2BRVSkBranch if Overflow Flag is Setif $(V = 1)$ then $PC \leftarrow PC + k + 1$ None1/2BRVCkBranch if Interrupt Enabledif $(V = 0)$ then $PC \leftarrow PC + k + 1$ None1/2BRIEkBranch if Interrupt Enabledif $(I = 0)$ then $PC \leftarrow PC + k + 1$ None1/2BRIDkBranch if Interrupt Enabledif $(I = 0)$ then $PC \leftarrow PC + k + 1$ None1/2	BRSH	k	Branch if Same or Higher	If $(C = 0)$ then $PC \leftarrow PC + k + 1$	None	1/2
BRMIkBranch if Minusif $(N = 1)$ then $PC \leftarrow PC + k + 1$ None1/2BRPLkBranch if Plusif $(N = 0)$ then $PC \leftarrow PC + k + 1$ None1/2BRGEkBranch if Greater or Equal, Signedif $(N \oplus V = 0)$ then $PC \leftarrow PC + k + 1$ None1/2BRLTkBranch if Less Than Zero, Signedif $(N \oplus V = 1)$ then $PC \leftarrow PC + k + 1$ None1/2BRHSkBranch if Half-carry Flag Setif $(H = 1)$ then $PC \leftarrow PC + k + 1$ None1/2BRHCkBranch if Half-carry Flag Clearedif $(H = 0)$ then $PC \leftarrow PC + k + 1$ None1/2BRTSkBranch if T-flag Setif $(T = 1)$ then $PC \leftarrow PC + k + 1$ None1/2BRTCkBranch if T-flag Setif $(T = 0)$ then $PC \leftarrow PC + k + 1$ None1/2BRVSkBranch if Overflow Flag is Setif $(V = 1)$ then $PC \leftarrow PC + k + 1$ None1/2BRVCkBranch if Overflow Flag is Clearedif $(V = 0)$ then $PC \leftarrow PC + k + 1$ None1/2BRIEkBranch if Interrupt Enabledif $(I = 0)$ then $PC \leftarrow PC + k + 1$ None1/2BRIDkBranch if Interrupt Enabledif $(I = 0)$ then $PC \leftarrow PC + k + 1$ None1/2	BRLO	ĸ	Branch if Lower	If $(C = 1)$ then $PC \leftarrow PC + k + 1$	None	1/2
BRPLkBranch if Plusif (N = 0) then PC $\leftarrow$ PC + k + 1None1/2BRGEkBranch if Greater or Equal, Signedif (N $\oplus$ V = 0) then PC $\leftarrow$ PC + k + 1None1/2BRLTkBranch if Less Than Zero, Signedif (N $\oplus$ V = 1) then PC $\leftarrow$ PC + k + 1None1/2BRHSkBranch if Half-carry Flag Setif (H = 1) then PC $\leftarrow$ PC + k + 1None1/2BRHCkBranch if Half-carry Flag Clearedif (H = 0) then PC $\leftarrow$ PC + k + 1None1/2BRTSkBranch if T-flag Setif (T = 1) then PC $\leftarrow$ PC + k + 1None1/2BRTCkBranch if T-flag Clearedif (T = 0) then PC $\leftarrow$ PC + k + 1None1/2BRVSkBranch if Overflow Flag is Setif (V = 1) then PC $\leftarrow$ PC + k + 1None1/2BRVCkBranch if Overflow Flag is Clearedif (V = 1) then PC $\leftarrow$ PC + k + 1None1/2BRIEkBranch if Interrupt Enabledif (I = 1) then PC $\leftarrow$ PC + k + 1None1/2BRIDkBranch if Interrupt Enabledif (I = 0) then PC $\leftarrow$ PC + k + 1None1/2	BRMI	k	Branch if Minus	If $(N = 1)$ then $PC \leftarrow PC + k + 1$	None	1/2
BRGEkBranch if Greater of Equal, Signedif $(N \oplus V = 0)$ then $PC \leftarrow PC + k + 1$ None1/2BRLTkBranch if Less Than Zero, Signedif $(N \oplus V = 1)$ then $PC \leftarrow PC + k + 1$ None1/2BRHSkBranch if Half-carry Flag Setif $(H = 1)$ then $PC \leftarrow PC + k + 1$ None1/2BRHCkBranch if Half-carry Flag Clearedif $(H = 0)$ then $PC \leftarrow PC + k + 1$ None1/2BRTSkBranch if T-flag Setif $(T = 1)$ then $PC \leftarrow PC + k + 1$ None1/2BRTCkBranch if T-flag Clearedif $(T = 0)$ then $PC \leftarrow PC + k + 1$ None1/2BRVSkBranch if Overflow Flag is Setif $(V = 1)$ then $PC \leftarrow PC + k + 1$ None1/2BRVCkBranch if Overflow Flag is Clearedif $(V = 0)$ then $PC \leftarrow PC + k + 1$ None1/2BRIEkBranch if Interrupt Enabledif $(I = 0)$ then $PC \leftarrow PC + k + 1$ None1/2BRIDkBranch if Interrupt Enabledif $(I = 0)$ then $PC \leftarrow PC + k + 1$ None1/2	BRPL	k	Branch if Plus	If $(N = 0)$ then $PC \leftarrow PC + k + 1$	None	1/2
BRL1kBranch if Less Than Zero, Signedif $(M \oplus V = 1)$ then $PC \leftarrow PC + k + 1$ None1/2BRHSkBranch if Half-carry Flag Setif $(H = 1)$ then $PC \leftarrow PC + k + 1$ None1/2BRHCkBranch if Half-carry Flag Clearedif $(H = 0)$ then $PC \leftarrow PC + k + 1$ None1/2BRTSkBranch if T-flag Setif $(T = 1)$ then $PC \leftarrow PC + k + 1$ None1/2BRTCkBranch if T-flag Clearedif $(T = 0)$ then $PC \leftarrow PC + k + 1$ None1/2BRVSkBranch if Overflow Flag is Setif $(V = 1)$ then $PC \leftarrow PC + k + 1$ None1/2BRVCkBranch if Overflow Flag is Clearedif $(V = 0)$ then $PC \leftarrow PC + k + 1$ None1/2BRIEkBranch if Interrupt Enabledif $(I = 0)$ then $PC \leftarrow PC + k + 1$ None1/2BRIDkBranch if Interrupt Enabledif $(I = 0)$ then $PC \leftarrow PC + k + 1$ None1/2	BRGE	ĸ	Branch if Greater or Equal, Signed	If $(N \oplus V = 0)$ then $PC \leftarrow PC + k + 1$	None	1/2
BRHSkBranch if Half-carry Flag Setif $(H = 1)$ then $PC \leftarrow PC + k + 1$ None1/2BRHCkBranch if Half-carry Flag Clearedif $(H = 0)$ then $PC \leftarrow PC + k + 1$ None1/2BRTSkBranch if T-flag Setif $(T = 1)$ then $PC \leftarrow PC + k + 1$ None1/2BRTCkBranch if T-flag Clearedif $(T = 0)$ then $PC \leftarrow PC + k + 1$ None1/2BRVSkBranch if Overflow Flag is Setif $(V = 1)$ then $PC \leftarrow PC + k + 1$ None1/2BRVCkBranch if Overflow Flag is Clearedif $(V = 0)$ then $PC \leftarrow PC + k + 1$ None1/2BRIEkBranch if Interrupt Enabledif $(I = 0)$ then $PC \leftarrow PC + k + 1$ None1/2BRIDkBranch if Interrupt Enabledif $(I = 0)$ then $PC \leftarrow PC + k + 1$ None1/2	BRLI	ĸ	Branch if Less Than Zero, Signed	If $(N \oplus V = 1)$ then PC $\leftarrow$ PC + k + 1	None	1/2
DRTCKBranch if T-flag Setif (T = 0) then PC $\leftarrow$ PC + K + 1None1/2BRTSkBranch if T-flag Setif (T = 1) then PC $\leftarrow$ PC + K + 1None1/2BRTCkBranch if T-flag Clearedif (T = 0) then PC $\leftarrow$ PC + K + 1None1/2BRVSkBranch if Overflow Flag is Setif (V = 1) then PC $\leftarrow$ PC + K + 1None1/2BRVCkBranch if Overflow Flag is Clearedif (V = 0) then PC $\leftarrow$ PC + K + 1None1/2BRIEkBranch if Interrupt Enabledif (I = 0) then PC $\leftarrow$ PC + K + 1None1/2BRIDkBranch if Interrupt Enabledif (I = 0) then PC $\leftarrow$ PC + K + 1None1/2	BRHS	K	Branch if Half-carry Flag Set	If $(H = 1)$ then $PC \leftarrow PC + k + 1$	None	1/2
BRTSKBranch if 1-flag SetIf $(1 = 1)$ then $PC \leftarrow PC + k + 1$ None1/2BRTCkBranch if T-flag Clearedif $(T = 0)$ then $PC \leftarrow PC + k + 1$ None1/2BRVSkBranch if Overflow Flag is Setif $(V = 1)$ then $PC \leftarrow PC + k + 1$ None1/2BRVCkBranch if Overflow Flag is Clearedif $(V = 0)$ then $PC \leftarrow PC + k + 1$ None1/2BRIEkBranch if Interrupt Enabledif $(I = 0)$ then $PC \leftarrow PC + k + 1$ None1/2BRIDkBranch if Interrupt Enabledif $(I = 0)$ then $PC \leftarrow PC + k + 1$ None1/2	BRHU	ĸ	Dranch If Half-carry Flag Cleared	$ (n = 0) \text{ then } PC \leftarrow PC + K + 1 $	None	1/2
BRIC     K     Branch if 0verflow Flag is Set     if (I = 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       BRIE     k     Branch if Interrupt Enabled     if (I = 0) then PC ← PC + k + 1     None     1/2       BRID     k     Branch if Interrupt Enabled     if (I = 0) then PC ← PC + k + 1     None     1/2	BRIS	K	Branch If I-flag Set	If $(I = 1)$ then PC $\leftarrow$ PC + k + 1	None	1/2
BRVC     k     Branch if Overflow Flag is Cleared     If (V = 1) then PC ← PC + k + 1     None     1/2       BRIE     k     Branch if Interrupt Enabled     if (V = 0) then PC ← PC + k + 1     None     1/2       BRIE     k     Branch if Interrupt Enabled     if (I = 1) then PC ← PC + k + 1     None     1/2	BRIC	ĸ	Branch IT I-Tiag Cleared	If $(1 = 0)$ then $PC \leftarrow PC + K + 1$	None	1/2
BRIE     k     Branch if lnterrupt Enabled     if (l = 0) then PC ← PC + k + 1     None     1/2       BRIE     k     Branch if Interrupt Enabled     if (l = 1) then PC ← PC + k + 1     None     1/2	BRVS	ĸ	Branch If Overflow Flag is Set	$ii (v = 1) \text{ then } PC \leftarrow PC + K + 1$	None	1/2
DFILE 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		ĸ	Dranch if Uvernow Flag is Cleared	If $(v = 0)$ then $PC \leftarrow PC + K + 1$	None	1/2
		r.	Branch if Interrupt Disabled	if $(I = I)$ then PC $\leftarrow$ PC + K + I if $(I = 0)$ then PC $\leftarrow$ PC + k + 1	None	1/2





## Instruction Set Summary (Continued)

Mnemonic	Operands	Description	Operation	Flags	# Clocks
	operando	Description	operation	Tiago	# Olocks
		Meus hetusen Deristere		Nana	4
	Ru, Rr	Nove between Registers		None	1
	Bd X		$Rd \leftarrow K$	None	2
	Bd X+	Load Indirect and Post-inc	$Bd \leftarrow (X) X \leftarrow X + 1$	None	2
LD	Rd, -X	Load Indirect and Pre-dec.	$X \leftarrow X - 1$ , Bd $\leftarrow (X)$	None	2
LD	Rd, Y	Load Indirect	$Rd \leftarrow (Y)$	None	2
LD	Rd, Y+	Load Indirect and Post-inc.	$Rd \leftarrow (Y), Y \leftarrow Y + 1$	None	2
LD	Rd, -Y	Load Indirect and Pre-dec.	$Y \leftarrow Y - 1$ , Rd $\leftarrow$ (Y)	None	2
LDD	Rd,Y+q	Load Indirect with Displacement	$Rd \leftarrow (Y + q)$	None	2
LD	Rd, Z	Load Indirect	$Rd \leftarrow (Z)$	None	2
LD	Rd, Z+	Load Indirect and Post-inc.	$Rd \leftarrow (Z), Z \leftarrow Z + 1$	None	2
LD	Rd, -Z	Load Indirect and Pre-dec.	$Z \leftarrow Z - 1$ , $Rd \leftarrow (Z)$	None	2
LDD	Rd, Z+q	Load Indirect with Displacement	$Rd \leftarrow (Z + q)$	None	2
LDS	Rd, k	Load Direct from SRAM	$Rd \leftarrow (k)$	None	2
ST	X, Rr	Store Indirect	$(X) \leftarrow Rr$	None	2
ST	X+, Rr	Store Indirect and Post-inc.	$(X) \leftarrow \operatorname{Rr}, X \leftarrow X + 1$	None	2
ST	-X, Rr	Store Indirect and Pre-dec.	$X \leftarrow X - 1$ , (X) $\leftarrow Rr$	None	2
ST	Y, Rr	Store Indirect	$(Y) \leftarrow Rr$	None	2
ST	Y+, Rr	Store Indirect and Post-inc.	$(Y) \leftarrow Rr, Y \leftarrow Y + 1$	None	2
ST	-Y, Rr	Store Indirect and Pre-dec.	$Y \leftarrow Y - 1$ , (Y) $\leftarrow Rr$	None	2
STD	Y+q, Rr	Store Indirect with Displacement	$(Y + q) \leftarrow Rr$	None	2
ST	Z, Rr	Store Indirect	$(Z) \leftarrow Rr$	None	2
ST	Z+, Rr	Store Indirect and Post-inc.	$(Z) \leftarrow Rr, Z \leftarrow Z + 1$	None	2
ST	-Z, Rr	Store Indirect and Pre-dec.	$Z \leftarrow Z - 1, (Z) \leftarrow Rr$	None	2
STD	Z+q, Rr	Store Indirect with Displacement	$(Z + q) \leftarrow Rr$	None	2
STS	k, Rr	Store Direct to SRAM	(k) ← Rr	None	2
LPM		Load Program Memory	$R0 \leftarrow (Z)$	None	3
IN	Rd, P	In Port		None	1
DUDU	P, Rr	Out Port	$P \leftarrow Rr$	None	1
PUSH	Rr	Push Register on Stack		None	2
		Pop Register from Stack	Rd ← STACK	None	2
SBI	Ph	Set Bit in I/O Begister	$I/O(P h) \leftarrow 1$	None	2
CBI	P b	Clear Bit in I/O Begister	$I/O(P b) \leftarrow 0$	None	2
	Bd	Logical Shift Left	$Bd(n+1) \leftarrow Bd(n) Bd(0) \leftarrow 0$	ZCNV	1
LSB	Bd	Logical Shift Bight	$Bd(n) \leftarrow Bd(n+1), Bd(7) \leftarrow 0$	Z.C.N.V	1
BOL	Bd	Botate Left through Carry	$Bd(0) \leftarrow C, Bd(n+1) \leftarrow Bd(n), C \leftarrow Bd(7)$	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	$Rd(n) \leftarrow Rd(n+1), n = 06$	Z,C,N,V	1
SWAP	Rd	Swap Nibbles	$Rd(30) \leftarrow Rd(74), Rd(74) \leftarrow Rd(30)$	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	$T \leftarrow Rr(b)$	Т	1
BLD	Rd, b	Bit Load from T to Register	$Rd(b) \leftarrow T$	None	1
SEC		Set Carry	C ← 1	С	1
CLC		Clear Carry	$C \leftarrow 0$	С	1
SEN		Set Negative Flag	N ← 1	N	1
CLN		Clear Negative Flag	$N \leftarrow 0$	N	1
SEZ		Set Zero Flag	Z ← 1	Z	1
CLZ		Clear Zero Flag	Z ← 0	Z	1
SEI		Global Interrupt Enable	l ← 1	1	1
CLI		Global Interrupt Disable	← 0	1	1
SES	+	Set Signed Test Flag	S ← 1	S	1
CLS		Clear Signed Test Flag	<u>S</u> ←0	S	1
SEV		Set Two's Complement Overflow	V ← 1	V	1
CLV		Clear Two's Complement Overflow	V ← 0	V	1
SEI		Set I in SREG			1
		Clear I in SREG			1
SEH	+	Set Half-carry Flag in SREG		н	1
	+	Clear Half-carry Flag in SREG	U → H	H	1
	+		(and appointing descer for Older function)	None	1
WDR		Watchdog Beset	(see specific descr. for Sieep function)	None	1
				INCHE	i I

8S2







## **Atmel Headquarters**

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

## Europe

Atmel SarL Route des Arsenaux 41 Casa Postale 80 CH-1705 Fribourg Switzerland TEL (41) 26-426-5555 FAX (41) 26-426-5500

#### 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 Product Operations**

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

#### Atmel Grenoble

Avenue de Rochepleine BP 123 38521 Saint-Egreve Cedex, France TEL (33) 4-7658-3000 FAX (33) 4-7658-3480

#### Atmel Heilbronn

Theresienstrasse 2 POB 3535 D-74025 Heilbronn, Germany TEL (49) 71 31 67 25 94 FAX (49) 71 31 67 24 23

#### Atmel Nantes

La Chantrerie BP 70602 44306 Nantes Cedex 3, France TEL (33) 0 2 40 18 18 18 FAX (33) 0 2 40 18 19 60

#### Atmel Rousset

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

#### Atmel Smart Card ICs

Scottish Enterprise Technology Park East Kilbride, Scotland G75 0QR TEL (44) 1355-357-000 FAX (44) 1355-242-743

> *e-mail* literature@atmel.com

> Web Site http://www.atmel.com

BBS 1-(408) 436-4309

#### © Atmel Corporation 2001.

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.

ATMEL® and AVR® are the registered trademarks of Atmel.

Other terms and product names may be the trademarks of others.

