# 8-BIT MICROCONTROLLER WITH 2K BYTES BUILD-IN PROGRAMMABLE FLASH

## **Description**

The IN90S2323D and IN90LS2323D is a low-power CMOS 8-bit microcontrollers based on the *AVR* enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the IN90S2323D and IN90LS2323D 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.

#### **Features**

- Utilizes the AVR ® Enhanced RISC Architecture
- AVR High Performance and Low Power RISC Architecture
- 118 Powerful Instructions Most Single Clock Cycle Execution
- · 2K bytes of In-System Programmable ISP Flash
- SPI Serial Interface for In-System Programming
- Endurance: 1,000 Write/Erase Cycles
- 128 bytes EEPROM
- Endurance: 100,000 Write/Erase Cycles
- 128 bytes Internal RAM
- 32 x 8 General Purpose Working Registers
- 3 Programmable I/O Lines
  - V<sub>CC</sub>: 4.0 6.0V IN90S2323D
  - V<sub>CC</sub>: 2.7 6.0V IN90LS2323D
- · Power-On Reset Circuit
- Speed Grades: 0 10 MHz IN90S2323D
- Speed Grades: 0 4 MHz IN90LS2323D
- Up to 10 MIPS Throughput at 10 MHz
- · One 8-Bit Timer/Counter with Separate Prescaler
- · External and Internal Interrupt Sources
- Programmable Watchdog Timer with On-Chip Oscillator
- · Low Power Idle and Power Down Modes
- · Programming Lock for Flash Program and EEPROM Data Security
- · Selectable On-Chip RC Oscillator
  - 8-Pin Device



# **Block Diagram** 8-BIT DATA BUS INTERNAL OSCILLATOR GND PROGRAM COUNTER STACK WATCHDOG TIMING AND RESET POINTER TIMER CONTROL PROGRAM FLASH MCU CONTROL REGISTER SRAM INSTRUCTION REGISTER GENERAL PURPOSE REGISTERS TIMER/ COUNTER INSTRUCTION DECODER INTERRUPT UNIT CONTROL ΑĽU **EEPROM** STATUS REGISTER PROGRAMMING SPI OSCILLATOR LOGIC DATA REGISTER PORTB DATA DIR. REG. PORTB PORTB DRIVERS

PB0 - PB2

# **Pin Descriptions**

#### **VCC**

Supply voltage pin.

#### **GND**

Ground pin.

## Port B (PB2..PB0)

Port B is a 3-bit bi-directional I/O port. Port pins can provide internal pull-up resistors (selected for each bit).

#### **RESET**

Reset input. A low on this pin for two machine cycles while the oscillator is running resets the device.

#### XTAL1

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

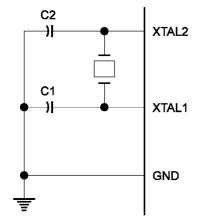
#### XTAL2

Output from the inverting oscillator amplifier.

## **Clock Sources**

The IN90S2313D and IN90LS2313D contains an inverting amplifier which can be configured for use as an on-chip oscillator. XTAL1 and XTAL2 are input and output respectively. Either a quartz crystal or a ceramic resonator may be used.

# **Oscillator Connection**



# **External Clock Drive Configuration**

RESET ☐ 1

XTAL1 ☐ 2

XTAL2 ☐ 3

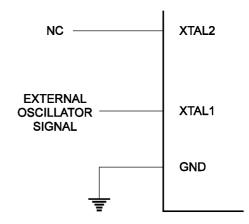
GND ☐ 4

8 🗆 VCC

7 PB2 (SCK/T0)

5 PB0 (MOSI)

6 ☐ PB1 (MISO/INT0)



#### **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.

Six of the 32 registers can be used as three 16-bits indirect address register pointers for Data Space addressing-enabling efficient address calculations. One of the three address pointers is also used as the address pointer for the constant table look up function. These added function registers are the 16-bit X-



register, Y-register and Z-register. 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. In addition to the register operation, the conventional memory addressing modes can be used on the register file as well. This is enabled by the fact that the register file is assigned the 32 lowermost Data Space addresses (\$00 -\$1F), allowing them to be accessed as though they were ordinary memory locations.

The I/O memory space contains 64 addresses for CPU peripheral functions as Control Registers, Timer/Counters, A/D-converters, and other I/O functions. The I/O memory can be accessed directly, or as the Data Space locations following those of the register file, \$20 - \$5F.

The AVR has Harvard architecture - with separate memories and buses for program and data. 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 in every clock cycle. The program memory is in-system downloadable Flash memory.

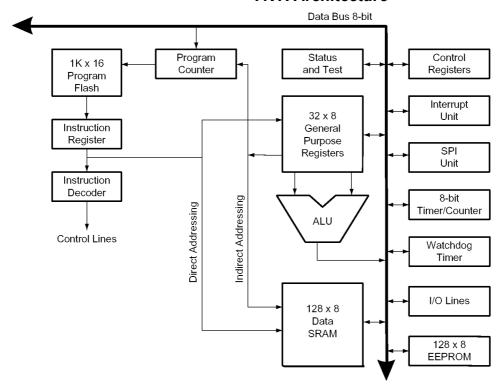
With the relative jump and call instructions, the whole 1K address space is directly accessed. Most *AVR* instructions have a single 16-bit word format. Every program memory address contains a 16- or 32-bit instruction.

During interrupts and subroutine calls, the return address program counter (PC) is stored on the stack. The stack is effectively allocated in the general data SRAM, and consequently the stack size is only limited by the total SRAM size and the usage of the SRAM. All user programs must initialize the SP in the reset routine (before subroutines or interrupts are executed). The 8-bit stack pointer SP is read/write accessible in the I/O space.

The 128 bytes data SRAM + register file and I/O registers can be easily accessed through the five different addressing modes supported in the AVR architecture.

The memory spaces in the AVR architecture are all linear and regular memory maps.

### **AVR Architecture**



# **Memory Maps** Data Memory Program Memory \$000 \$00 \$000 32 Gen. Purpose Working Registers \$1F \$20 64 I/O Registers **EEPROM** Program Flash (1K x 16) (128 x 8) \$5F \$60 \$07F SRAM (128 x 8) \$DF \$3FF

## **REGISTER SUMMARY**

\$3F (\$5F) \$3E (\$5E) R \$3D (\$5D) \$3C (\$5C) R \$3B (\$5B) \$3A (\$5A) \$39 (\$59) \$38 (\$58) \$37 (\$57) R \$36 (\$56) R \$35 (\$55) R \$31 (\$51) R \$33 (\$53) R \$32 (\$52) R \$31 (\$51) R \$30 (\$50) R \$2F (\$4F) R \$2E (\$4E) R \$2D (\$4D) R \$2C (\$4C) R \$2B (\$4B) R \$29 (\$49) R \$28 (\$48) R	Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved	SP7	Bit 6 T SP6 INTO INTFO counter0 (8	Bit 5 H SP5 SE - Bit)	SP4		Bit 2 N SP2 CS02	SP1  TOIE0 TOV0  ISC01 EXTRF	SP0  ISC00 PORF	page 13  page 13  page 17  page 17  page 17  page 15  page 16  page 16
\$3E (\$5E) R \$3D (\$5D) \$3C (\$5C) R \$3B (\$5B) \$3A (\$5A) \$39 (\$59) \$38 (\$58) \$37 (\$57) R \$36 (\$56) R \$35 (\$55) N \$34 (\$54) R \$33 (\$53) \$32 (\$52) \$31 (\$51) R \$30 (\$50) R \$2E (\$4E) R \$2E (\$4E) R \$2D (\$4D) R \$2C (\$4C) R \$2B (\$4B) R \$2A (\$4A) R \$29 (\$49) R	Reserved SPL Reserved GIMSK GIFR TIMSK TIFR Reserved RESERVED MCUCR MCUCR MCUSR TCCR0 TCNT0 RESERVED	- - - - -	SP6 INTO INTFO	SP5 SE				SP1  TOIE0 TOV0  ISC01 EXTRF	SP0 ISC00	page 13  page 17 page 17 page 15 page 16  page 16
\$3D (\$5D) \$3C (\$5C) R \$3B (\$5B) \$3A (\$5A) \$39 (\$59) \$38 (\$58) \$37 (\$57) R \$36 (\$56) R \$35 (\$55) N \$34 (\$54) R \$33 (\$53) \$32 (\$52) \$31 (\$51) R \$30 (\$50) R \$25 (\$4F) R \$22 (\$4E) R	SPL Reserved GIMSK GIFR TIMSK TIFR Reserved Reserved MCUCR MCUSR TCCR0 TCNT0 Reserved	- - - - -	INTO INTFO - - - -	- - - - SE -	- - - - SM	-	-	TOIE0 TOV0	- - - - ISC00	page 17 page 17 page 15 page 16
\$3C (\$5C) R \$3B (\$5B) \$3A (\$5A) \$39 (\$59) \$38 (\$58) \$37 (\$57) R \$36 (\$56) R \$35 (\$55) M \$33 (\$53) \$32 (\$52) \$31 (\$51) R \$30 (\$50) R \$2F (\$4F) R \$2E (\$4E) R \$2D (\$4D) R \$2C (\$4C) R \$2B (\$4B) R \$2A (\$4A) R \$29 (\$49) R \$28 (\$48) R	Reserved GIMSK GIFR TIMSK TIFR Reserved Reserved MCUCR MCUSR TCCR0 TCNT0 Reserved	- - - - -	INTO INTFO - - - -	- - - - SE -	- - - - SM	-		TOIE0 TOV0	- - - - ISC00	page 17 page 17 page 15 page 16
\$3B (\$5B) \$3A (\$5A) \$39 (\$59) \$38 (\$58) \$37 (\$57) \$36 (\$56) \$35 (\$55) \$34 (\$54) \$33 (\$53) \$32 (\$52) \$31 (\$51) \$30 (\$50) \$2F (\$4F) \$22 (\$4E) \$23 (\$4E) \$24 (\$4E) \$25 (\$4E)	GIMSK GIFR TIMSK TIFR Reserved Reserved MCUCR MCUSR TCCR0 TCNT0 Reserved	- - - -	- - - - -	- - SE -	- - - SM			TOIE0 TOV0 ISC01 EXTRF	- - ISC00	page 17 page 15 page 16 page 16
\$3A (\$5A) \$39 (\$59) \$38 (\$58) \$37 (\$57) \$36 (\$56) \$35 (\$55) \$34 (\$54) \$33 (\$53) \$32 (\$52) \$31 (\$51) \$30 (\$50) \$2F (\$4F) \$22 (\$4E) \$22 (\$4E) \$22 (\$4E) \$22 (\$4E) \$22 (\$4E) \$22 (\$4E) \$23 (\$54E) \$24 (\$4E) \$25 (\$4E) \$26 (\$4E) \$27 (\$4E) \$28 (\$4E) \$28 (\$4E) \$28 (\$4E) \$28 (\$4E) \$28 (\$4E) \$38 (\$4E) \$38 (\$4E) \$38 (\$4E) \$39 (\$4E) \$39 (\$4E) \$30 (\$50) \$40 (\$4E) \$40 (	GIFR TIMSK TIFR Reserved Reserved MCUCR MCUSR TCR0 TCNT0 Reserved	- - - -	- - - - -	- - SE -	- - - SM			TOIE0 TOV0 ISC01 EXTRF	- - ISC00	page 17 page 15 page 16 page 16
\$39 (\$59) \$38 (\$58) \$37 (\$57) \$36 (\$56) \$35 (\$55) \$34 (\$54) \$33 (\$53) \$32 (\$52) \$31 (\$51) \$30 (\$50) \$2F (\$4F) \$22 (\$4E) \$22 (\$4C) \$28 (\$4B) \$29 (\$49) \$28 (\$48)	TIMSK TIFR Reserved Reserved MCUCR MCUSR TCR0 TCNT0 Reserved	- - - - -	- - - -	SE - -	SM	-	-	ISC01 EXTRF	ISC00	page 15 page 16 page 16
\$38 (\$58) \$37 (\$57) R \$36 (\$56) R \$35 (\$55) M \$34 (\$54) R \$33 (\$53) \$32 (\$52) \$31 (\$51) R \$30 (\$50) R \$2F (\$4F) R \$2E (\$4E) R \$2D (\$4D) R \$2C (\$4C) R \$2B (\$4B) R \$2A (\$4A) R \$29 (\$49) R	TIFR Reserved Reserved MCUCR MCUSR TCCR0 TCNT0 Reserved	- - -		SE - -	SM	-	-	ISC01 EXTRF	ISC00	page 16 page 16
\$37 (\$57) R \$36 (\$56) R \$35 (\$55) M \$34 (\$54) M \$33 (\$53) S \$32 (\$52) S \$31 (\$51) R \$30 (\$50) R \$2F (\$4F) R \$2E (\$4E) R \$2D (\$4D) R \$2C (\$4C) R \$2B (\$4B) R \$2A (\$4A) R \$29 (\$49) R	Reserved Reserved MCUCR MCUSR TCCR0 TCNT0 Reserved	- - -		SE - -	SM -	-	-	ISC01 EXTRF	ISC00	page 16
\$36 (\$56) R \$35 (\$55) M \$34 (\$54) M \$33 (\$53) \$32 (\$52) \$31 (\$51) R \$30 (\$50) R \$2F (\$4F) R \$2E (\$4E) R \$2D (\$4D) R \$2D (\$4D) R \$22 (\$4C) R \$28 (\$4B) R \$29 (\$4B) R \$28 (\$4B) R	Reserved MCUCR MCUSR TCCR0 TCNT0 Reserved	-	-	-	-	-	- - CS02	EXTRF		
\$35 (\$55) M \$34 (\$54) M \$33 (\$53) \$32 (\$52) \$31 (\$51) R \$30 (\$50) R \$2F (\$4F) R \$2E (\$4E) R \$2D (\$4D) R \$2C (\$4C) R \$2B (\$4B) R \$24 (\$4A) R \$29 (\$49) R \$28 (\$48) R	MCUCR MCUSR TCCR0 TCNT0 Reserved	-	-	-	-	-	- - CS02	EXTRF		
\$34 (\$54) \$33 (\$53) \$32 (\$52) \$31 (\$51) \$30 (\$50) \$2F (\$4F) \$2E (\$4E) \$2D (\$4D) \$2C (\$4C) \$2B (\$4B) \$24 (\$4A) \$29 (\$49) \$28 (\$48)	MCUSR TCCR0 TCNT0 Reserved	-	-	-	-	-	- - CS02	EXTRF		
\$33 (\$53) \$32 (\$52) \$31 (\$51) \$30 (\$50) \$2F (\$4F) \$2E (\$4E) \$2D (\$4D) \$2C (\$4C) \$2B (\$4B) \$24 (\$4A) \$29 (\$49) \$28 (\$48)	TCCR0 TCNT0 Reserved	-	-	- Bit)			- CS02		PORF	
\$32 (\$52) \$31 (\$51) \$30 (\$50) \$2F (\$4F) \$2E (\$4E) \$2D (\$4D) \$2C (\$4C) \$2B (\$4B) \$24 (\$4A) \$29 (\$49) \$28 (\$48)	TCNT0 Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved			Bit)	-	-	CS02			page 14
\$31 (\$51) R \$30 (\$50) R \$2F (\$4F) R \$2E (\$4E) R \$2D (\$4D) R \$2C (\$4C) R \$2B (\$4B) R \$24 (\$4A) R \$29 (\$49) R	Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved	Timer/Co	ounter0 (8	Bit)				CS01	CS00	page 20
\$30 (\$50) R \$2F (\$4F) R \$2E (\$4E) R \$2D (\$4D) R \$2C (\$4C) R \$2B (\$4B) R \$24 (\$4A) R \$29 (\$49) R	Reserved Reserved Reserved Reserved Reserved Reserved									page 20
\$2F (\$4F) R \$2E (\$4E) R \$2D (\$4D) R \$2C (\$4C) R \$2B (\$4B) R \$2A (\$4A) R \$29 (\$49) R \$28 (\$48) R	Reserved Reserved Reserved Reserved Reserved									
\$2F (\$4F) R \$2E (\$4E) R \$2D (\$4D) R \$2C (\$4C) R \$2B (\$4B) R \$2A (\$4A) R \$29 (\$49) R \$28 (\$48) R	Reserved Reserved Reserved Reserved									
\$2E (\$4E) R \$2D (\$4D) R \$2C (\$4C) R \$2B (\$4B) R \$2A (\$4A) R \$29 (\$49) R \$28 (\$48) R	Reserved Reserved Reserved Reserved									
\$2D (\$4D) R \$2C (\$4C) R \$2B (\$4B) R \$2A (\$4A) R \$29 (\$49) R \$28 (\$48) R	Reserved Reserved Reserved									
\$2C (\$4C) R \$2B (\$4B) R \$2A (\$4A) R \$29 (\$49) R \$28 (\$48) R	Reserved Reserved									
\$2B (\$4B) R \$2A (\$4A) R \$29 (\$49) R \$28 (\$48) R	Reserved									
\$2A (\$4A) R \$29 (\$49) R \$28 (\$48) R										
\$29 (\$49) R \$28 (\$48) R	Reserved									
\$28 (\$48) R	Reserved									
	Reserved									
\$27 (\$47) R	Reserved									
	Reserved									
	Reserved									
	Reserved									
	Reserved									
	Reserved									
	WDTCR				WDTO	WDE	WDP2	WDP1	WDP0	2000 21
		-	-	-	WDIO	WDE	WDFZ	WDFI	WDFU	page 21
	Reserved Reserved									
. ( ,			FEDDO	1 1 1 1 1 1 1 1 1 1	Danistan					00
\$1E (\$3E)	EEAR	-		/I Address	Register					page 22
\$1D (\$3D)	EEDR	EEPROM Data							page 22	
#4C (#2C)	FECD	reg	jister						FEDE	00
\$1C (\$3C)	EECR	-	-	-	-	-	EEMW	EEWE	EERE	page 22
	Reserved									
	Reserved									
	Reserved	-	1		D05==	D00==	B05==	D0575	D0575	
	PORTB	-	-	-	PORTB	PORTB	PORTB	PORTB	PORTB	page 23
	DDRB	-	-	-	DDB4	DDB3	DDB2	DDB1	DDB0	page 23
\$16 (\$36)	PINB	-	-	-	PINB4	PINB3	PINB2	PINB1	PINB0	page 23
	Reserved									
	Reserved									
	Reserved									
	Reserved									
	Reserved									
	Reserved									
	Reserved									
	Reserved									
	Reserved									
	Reserved									
	Reserved					_	_			
	Reserved									
	Reserved									
	Reserved									
	Reserved									
	Reserved									



**Instruction Set Summary** 

Instruction Set Summary									
Mnemonics	Operands	Description	Operation	Flags	#Clock				
ARITHMETIC A	ND LOGIC INSTR	RUCTIONS	<u> </u>						
ADD	Rd, Rr	Add two Registers	Rd ← 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				
ADIW	RdI,K	Add Immediate to Word	Rdh:Rdl ← Rdh:Rdl + K	Z,C,N,V,S	2				
SUB	Rd, Rr	Subtract two Registers	Rd ← Rd - Rr	Z,C,N,V,H	1				
SUBI	Rd, K	Subtract Constant from Register	Rd ← Rd - K	Z,C,N,V,H	1				
SBIW	RdI,K	Subtract Immediate from Word	Rdh:Rdl ← Rdh:Rdl - K	Z,C,N,V,S	2				
SBC	Rd, Rr	Subtract with Carry two Registers	Rd ← Rd - Rr – C	Z,C,N,V,H	1				
SBCI	Rd, K	Subtract with Carry Constant from Reg.	Rd ← Rd - K - C	Z,C,N,V,H	1				
AND	Rd, Rr	Logical AND Registers	Rd ← Rd • Rr	Z,N,V	1				
ANDI	Rd, K	Logical AND Register and Constant	Rd ← Rd • K	Z,N,V	1				
OR	Rd, Rr	Logical OR Registers	Rd ← Rd v Rr	Z,N,V	1				
ORI	Rd, K	Logical OR Register and Constant	Rd ← Rd v K	Z.N.V	1				
EOR	Rd, Rr	Exclusive OR Registers	Rd ← Rd ⊕ Rr	Z,N,V	1				
COM	Rd	One's Complement	Rd ← \$FF - Rd	Z,C,N,V	1				
NEG	Rd		·		1				
SBR	Rd,K	Two's Complement	$Rd \leftarrow \$00 - Rd$ $Rd \leftarrow Rd \lor K$	Z,C,N,V,H Z,N,V	1				
CBR	Rd,K Rd,K	Set Bit(s) in Register			1				
	,	Clear Bit(s) in Register	Rd ← Rd • (\$FF - K)	Z,N,V					
INC	Rd	Increment	Rd ← Rd + 1	Z,N,V	1				
DEC	Rd	Decrement	Rd ← Rd - 1	Z,N,V	1				
TST	Rd	Test for Zero or Minus	$Rd \leftarrow Rd \cdot Rd$	Z,N,V	1				
CLR	Rd	Clear Register	$Rd \leftarrow Rd \oplus Rd$	Z,N,V	1				
SER	Rd	Set Register	Rd ← \$FF	None	1				
BRANCH INSTR		ln	DO DO 1 1 1	h.					
RJMP	k	Relative Jump	PC ← PC + k + 1	None	2				
IJMP		Indirect Jump to (Z)	PC ← Z	None	2				
RCALL	k	Relative Subroutine Call	PC ← PC + k + 1	None	3				
ICALL		Indirect Call to (Z)	PC ← Z	None	3				
RET		Subroutine Return	PC ← STACK	None	4				
RETI		Interrupt Return	PC ← STACK	l	4				
CPSE	Rd,Rr	Compare, Skip if Equal	if (Rd = Rr) PC ← 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 ← PC + 2 or 3	None	1/2				
SBRS	Rr, b	Skip if Bit in Register is Set	if $(Rr(b)=1) PC \leftarrow PC + 2 \text{ or } 3$	None	1/2				
SBIC	P, b	Skip if Bit in I/O Register Cleared	if (P(b)=0) PC ← PC + 2 or 3	None	1/2				
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				
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 ← PC + k + 1	None	1/2				
BRNE	k	Branch if Not Equal	if (Z = 0) then PC ← PC + k + 1	None	1/2				
BRCS	k	Branch if Carry Set	if (C = 1) then PC ← PC + k + 1	None	1/2				
BRCC	k	Branch if Carry Cleared	if (C = 0) then PC ← PC + k + 1	None	1/2				
BRSH	k	Branch if Same or Higher	if (C = 0) then PC ← PC + k + 1	None	1/2				
BRLO	k	Branch if Lower	if (C = 1) then PC ← PC + k + 1	None	1/2				
BRMI	k	Branch if Minus	if (N = 1) then PC ← PC + k + 1	None	1/2				
BRPL	k	Branch if Plus	if (N = 0) then PC ← PC + k + 1	None	1/2				
BRGE	k	Branch if Greater or Equal, Signed	if (N ⊕ V= 0) then PC ← PC + k + 1	None	1/2				
BRLT	k	Branch if Less Than Zero. Signed	if (N ⊕ V= 1) then PC ← PC + k + 1	None	1/2				
BRHS	k	Branch if Half Carry Flag Set	if (H = 1) then PC ← PC + k + 1	None	1/2				
BRHC	k	Branch if Half Carry Flag Cleared	if (H = 0) then PC ← PC + k + 1	None	1/2				
BRTS	k	Branch if T Flag Set	if (T = 1) then PC ← PC + k + 1	None	1/2				
BRTC	k	Branch if T Flag Cleared	if (T = 0) then PC ← PC + k + 1	None	1/2				
BRVS	k	Branch if Overflow Flag is Set	if (V = 1) then PC ← PC + k + 1	None	1/2				
BRVC	k	Branch if Overflow Flag is Cleared	if (V = 0) then PC $\leftarrow$ PC + k + 1	None	1/2				
BRIE		Branch if Interrupt Enabled	if ( I = 1) then PC ← PC + k + 1	None	1/2				
BRID	k k	Branch if Interrupt Disabled	if (I = 0) then $PC \leftarrow PC + k + 1$	None	1/2				
סוטס	I.	שומווטו וו ווונכוועףו שוממטופע		INOLIC	1/2				



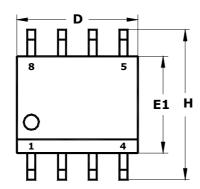
Instruction Set Summary (Continued)

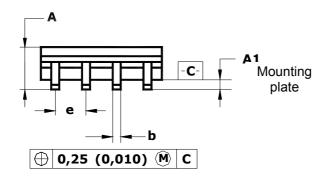
Mnemonics Operands Description #Clock Operation Flags

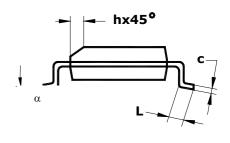
		Description	Operation	riays	#CIOCK
	FER INSTRUC		1		
MOV	Rd, Rr	Move Between Registers	$Rd \leftarrow Rr$	None	1
LDI	Rd, K	Load Immediate	$Rd \leftarrow K$	None	1
LD	Rd, X	Load Indirect	$Rd \leftarrow (X)$	None	2
LD	Rd, X+	Load Indirect and Post-Inc.	$Rd \leftarrow (X), X \leftarrow X + 1$	None	2
LD	Rd, - X	Load Indirect and Pre-Dec.	$X \leftarrow X^{c}$ 1, Rd $\leftarrow$ (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 ← (k)	None	2
ST	X, Rr	Store Indirect	(X) ← Rr	None	2
ST	X+, Rr	Store Indirect and Post-Inc.	$(X) \leftarrow Rr, X \leftarrow X + 1$	None	2
ST	- X, Rr	Store Indirect and Pre-Dec.	X ← X - 1, (X) ← Rr	None	2
ST	Y, Rr	Store Indirect	(Y) ← Rr	None	2
ST	Y+, Rr	Store Indirect and Post-Inc.	$(Y) \leftarrow RR, Y \leftarrow Y + 1$	None	2
ST	- Y, Rr	Store Indirect and Pre-Dec.	Y ← Y - 1, (Y) ← Rr	None	2
STD	Y+g,Rr	Store Indirect with Displacement	(Y + g) ← Rr	None	2
ST	Z, Rr	Store Indirect	(Z) ← 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	K, IXI	Load Program Memory	R0 ← (Z)	None	3
IN	Rd, P	In Port	Rd ← P	None	1
OUT	P, Rr	Out Port	P ← Rr	None	1
PUSH	Rr	Push Register on Stack	STACK ← Rr	None	2
POP	Rd		Rd ← STACK	None	2
		Pop Register from Stack	RU - STACK	None	
BIT AND BIT-T			LO(D b) 4	I Nissa	
SBI CBI	P,b	Set Bit in I/O Register	/O(P,b) ← 1	None	2 2
CBI	P,b	Clear Bit in I/O Register	I/O(P,b) ← 0	None	1
		Lawinal Chiff Laff			1 1
LSL	Rd	Logical Shift Left	$Rd(n+1) \leftarrow Rd(n), Rd(0) \leftarrow 0$	Z,C,N,V	
LSL LSR	Rd Rd	Logical Shift Right	$Rd(n) \leftarrow Rd(n+1), Rd(7) \leftarrow 0$	Z,C,N,V	1
LSL LSR ROL	Rd Rd Rd	Logical Shift Right Rotate Left Through Carry	$\begin{array}{c} Rd(n) \leftarrow Rd(n+1),  Rd(7) \leftarrow 0 \\ Rd(0) \leftarrow C, Rd(n+1) \leftarrow Rd(n), C \leftarrow Rd(7) \end{array}$	Z,C,N,V Z,C,N,V	1
LSL LSR ROL ROR	Rd Rd Rd Rd	Logical Shift Right Rotate Left Through Carry Rotate Right Through Carry	$ \begin{array}{l} Rd(n) \leftarrow Rd(n+1), Rd(7) \leftarrow 0 \\ Rd(0) \leftarrow C, Rd(n+1) \leftarrow Rd(n), C \leftarrow Rd(7) \\ Rd(7) \leftarrow C, Rd(n) \leftarrow Rd(n+1), C \leftarrow Rd(0) \end{array} $	Z,C,N,V Z,C,N,V Z,C,N,V	1 1 1
LSL LSR ROL ROR ASR	Rd Rd Rd Rd Rd	Logical Shift Right Rotate Left Through Carry Rotate Right Through Carry Arithmetic Shift Right	Rd(n) ← Rd(n+1), Rd(7) ← 0 Rd(0)←C,Rd(n+1)← Rd(n),C←Rd(7) Rd(7)←C,Rd(n)← Rd(n+1),C←Rd(0) Rd(n) ← Rd(n+1), n=06	Z,C,N,V Z,C,N,V Z,C,N,V Z,C,N,V	1 1 1
LSL LSR ROL ROR ASR SWAP	Rd Rd Rd Rd Rd Rd	Logical Shift Right Rotate Left Through Carry Rotate Right Through Carry Arithmetic Shift Right Swap Nibbles	Rd(n) ← Rd(n+1), Rd(7) ← 0 Rd(0)←C,Rd(n+1)← Rd(n),C←Rd(7) Rd(7)←C,Rd(n)← Rd(n+1),C←Rd(0) Rd(n) ← Rd(n+1), n=06 Rd(30)←Rd(74),Rd(74)←Rd(30)	Z,C,N,V Z,C,N,V Z,C,N,V Z,C,N,V None	1 1 1 1
LSL LSR ROL ROR ASR SWAP BSET	Rd Rd Rd Rd Rd Rd Rd	Logical Shift Right Rotate Left Through Carry Rotate Right Through Carry Arithmetic Shift Right Swap Nibbles Flag Set	$ \begin{array}{l} Rd(n) \leftarrow Rd(n+1), Rd(7) \leftarrow 0 \\ Rd(0) \leftarrow C, Rd(n+1) \leftarrow Rd(n), C \leftarrow Rd(7) \\ Rd(7) \leftarrow C, Rd(n) \leftarrow Rd(n+1), C \leftarrow Rd(0) \\ Rd(n) \leftarrow Rd(n+1), n=06 \\ Rd(30) \leftarrow Rd(74), Rd(74) \leftarrow Rd(30) \\ SREG(s) \leftarrow 1 \end{array} $	Z,C,N,V Z,C,N,V Z,C,N,V Z,C,N,V None SREG(s)	1 1 1 1 1
LSL LSR ROL ROR ASR SWAP BSET BCLR	Rd Rd Rd Rd Rd Rd S	Logical Shift Right Rotate Left Through Carry Rotate Right Through Carry Arithmetic Shift Right Swap Nibbles Flag Set Flag Clear	$ \begin{array}{c} Rd(n) \leftarrow Rd(n+1), Rd(7) \leftarrow 0 \\ Rd(0) \leftarrow C, Rd(n+1) \leftarrow Rd(n), C \leftarrow Rd(7) \\ Rd(7) \leftarrow C, Rd(n) \leftarrow Rd(n+1), C \leftarrow Rd(0) \\ Rd(n) \leftarrow Rd(n+1), n=06 \\ Rd(30) \leftarrow Rd(74), Rd(74) \leftarrow Rd(30) \\ SREG(s) \leftarrow 1 \\ SREG(s) \leftarrow 0 \end{array} $	Z,C,N,V Z,C,N,V Z,C,N,V Z,C,N,V None SREG(s) SREG(s)	1 1 1 1 1 1
LSL LSR ROL ROR ASR SWAP BSET BCLR BST	Rd Rf	Logical Shift Right Rotate Left Through Carry Rotate Right Through Carry Arithmetic Shift Right Swap Nibbles Flag Set Flag Clear Bit Store from Register to T	$ \begin{array}{c} Rd(n) \leftarrow Rd(n+1), Rd(7) \leftarrow 0 \\ Rd(0) \leftarrow C, Rd(n+1) \leftarrow Rd(n), C \leftarrow Rd(7) \\ Rd(7) \leftarrow C, Rd(n) \leftarrow Rd(n+1), C \leftarrow Rd(0) \\ Rd(n) \leftarrow Rd(n+1), n=06 \\ Rd(30) \leftarrow Rd(74), Rd(74) \leftarrow Rd(30) \\ SREG(s) \leftarrow 1 \\ SREG(s) \leftarrow 0 \\ T \leftarrow Rr(b) \end{array} $	Z,C,N,V Z,C,N,V Z,C,N,V Z,C,N,V None SREG(s) SREG(s)	1 1 1 1 1 1 1 1
LSL LSR ROL ROR ASR SWAP BSET BCLR BST BLD	Rd Rd Rd Rd Rd Rd S	Logical Shift Right Rotate Left Through Carry Rotate Right Through Carry Arithmetic Shift Right Swap Nibbles Flag Set Flag Clear Bit Store from Register to T Bit load from T to Register	$ \begin{array}{c} Rd(n) \leftarrow Rd(n+1), Rd(7) \leftarrow 0 \\ Rd(0) \leftarrow C, Rd(n+1) \leftarrow Rd(n), C \leftarrow Rd(7) \\ Rd(7) \leftarrow C, Rd(n) \leftarrow Rd(n+1), C \leftarrow Rd(0) \\ Rd(n) \leftarrow Rd(n+1), n=06 \\ Rd(30) \leftarrow Rd(74), Rd(74) \leftarrow Rd(30) \\ SREG(s) \leftarrow 1 \\ SREG(s) \leftarrow 0 \\ T \leftarrow Rr(b) \\ Rd(b) \leftarrow T \end{array} $	Z,C,N,V Z,C,N,V Z,C,N,V Z,C,N,V None SREG(s) SREG(s) T None	1 1 1 1 1 1 1 1 1
LSL LSR ROL ROR ASR SWAP BSET BCLR BST BLD SEC	Rd Rf	Logical Shift Right Rotate Left Through Carry Rotate Right Through Carry Arithmetic Shift Right Swap Nibbles Flag Set Flag Clear Bit Store from Register to T Bit load from T to Register Set Carry	$ \begin{array}{l} Rd(n) \leftarrow Rd(n+1), Rd(7) \leftarrow 0 \\ Rd(0) \leftarrow C, Rd(n+1) \leftarrow Rd(n), C \leftarrow Rd(7) \\ Rd(7) \leftarrow C, Rd(n) \leftarrow Rd(n+1), C \leftarrow Rd(0) \\ Rd(n) \leftarrow Rd(n+1), n=06 \\ Rd(30) \leftarrow Rd(74), Rd(74) \leftarrow Rd(30) \\ SREG(s) \leftarrow 1 \\ SREG(s) \leftarrow 0 \\ T \leftarrow Rr(b) \\ Rd(b) \leftarrow T \\ C \leftarrow 1 \end{array} $	Z,C,N,V Z,C,N,V Z,C,N,V Z,C,N,V None SREG(s) SREG(s) T None C	1 1 1 1 1 1 1 1 1 1
LSL LSR ROL ROR ASR SWAP BSET BCLR BST BLD SEC CLC	Rd Rf	Logical Shift Right Rotate Left Through Carry Rotate Right Through Carry Arithmetic Shift Right Swap Nibbles Flag Set Flag Clear Bit Store from Register to T Bit load from T to Register Set Carry Clear Carry	Rd(n) ← Rd(n+1), Rd(7) ← 0 Rd(0)←C,Rd(n+1)← Rd(n),C←Rd(7) Rd(7)←C,Rd(n)← Rd(n+1),C←Rd(0) Rd(n) ← Rd(n+1), n=06 Rd(30)←Rd(74),Rd(74)←Rd(30) SREG(s) ← 1 SREG(s) ← 0 $T \leftarrow Rr(b)$ Rd(b) ← $T$ $C \leftarrow 1$ $C \leftarrow 0$	Z,C,N,V Z,C,N,V Z,C,N,V Z,C,N,V None SREG(s) SREG(s) T None C C	1 1 1 1 1 1 1 1 1 1 1
LSL LSR ROL ROR ASR SWAP BSET BCLR BST BLD SEC CLC SEN	Rd Rf	Logical Shift Right Rotate Left Through Carry Rotate Right Through Carry Arithmetic Shift Right Swap Nibbles Flag Set Flag Clear Bit Store from Register to T Bit load from T to Register Set Carry Clear Carry Set Negative Flag	$ \begin{array}{l} Rd(n) \leftarrow Rd(n+1), Rd(7) \leftarrow 0 \\ Rd(0) \leftarrow C, Rd(n+1) \leftarrow Rd(n), C \leftarrow Rd(7) \\ Rd(7) \leftarrow C, Rd(n) \leftarrow Rd(n+1), C \leftarrow Rd(0) \\ Rd(n) \leftarrow Rd(n+1), n=06 \\ Rd(30) \leftarrow Rd(74), Rd(74) \leftarrow Rd(30) \\ SREG(s) \leftarrow 1 \\ SREG(s) \leftarrow 0 \\ T \leftarrow Rr(b) \\ Rd(b) \leftarrow T \\ C \leftarrow 1 \\ C \leftarrow 0 \\ N \leftarrow 1 \end{array} $	Z,C,N,V Z,C,N,V Z,C,N,V Z,C,N,V None SREG(s) SREG(s) T None C C N	1 1 1 1 1 1 1 1 1 1 1 1 1
LSL LSR ROL ROR ASR SWAP BSET BCLR BST BLD SEC CLC SEN CLN	Rd Rf	Logical Shift Right Rotate Left Through Carry Rotate Right Through Carry Arithmetic Shift Right Swap Nibbles Flag Set Flag Clear Bit Store from Register to T Bit load from T to Register Set Carry Clear Carry Set Negative Flag Clear Negative Flag	$ \begin{array}{l} Rd(n) \leftarrow Rd(n+1), Rd(7) \leftarrow 0 \\ Rd(0) \leftarrow C, Rd(n+1) \leftarrow Rd(n), C \leftarrow Rd(7) \\ Rd(7) \leftarrow C, Rd(n) \leftarrow Rd(n+1), C \leftarrow Rd(0) \\ Rd(n) \leftarrow Rd(n+1), n=06 \\ Rd(30) \leftarrow Rd(74), Rd(74) \leftarrow Rd(30) \\ SREG(s) \leftarrow 1 \\ SREG(s) \leftarrow 0 \\ T \leftarrow Rr(b) \\ Rd(b) \leftarrow T \\ C \leftarrow 1 \\ C \leftarrow 0 \\ N \leftarrow 1 \\ N \leftarrow 0 \end{array} $	Z,C,N,V Z,C,N,V Z,C,N,V Z,C,N,V None SREG(s) SREG(s) T None C C N	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
LSL LSR ROL ROR ASR SWAP BSET BCLR BST BLD SEC CLC SEN CLN SEZ	Rd Rf	Logical Shift Right Rotate Left Through Carry Rotate Right Through Carry Arithmetic Shift Right Swap Nibbles Flag Set Flag Clear Bit Store from Register to T Bit load from T to Register Set Carry Clear Carry Set Negative Flag Clear Negative Flag Set Zero Flag	$ \begin{array}{l} Rd(n) \leftarrow Rd(n+1), Rd(7) \leftarrow 0 \\ Rd(0) \leftarrow C, Rd(n+1) \leftarrow Rd(n), C \leftarrow Rd(7) \\ Rd(7) \leftarrow C, Rd(n) \leftarrow Rd(n+1), C \leftarrow Rd(0) \\ Rd(n) \leftarrow Rd(n+1), n=06 \\ Rd(30) \leftarrow Rd(74), Rd(74) \leftarrow Rd(30) \\ SREG(s) \leftarrow 1 \\ SREG(s) \leftarrow 0 \\ T \leftarrow Rr(b) \\ Rd(b) \leftarrow T \\ C \leftarrow 1 \\ C \leftarrow 0 \\ N \leftarrow 1 \\ N \leftarrow 0 \\ Z \leftarrow 1 \end{array} $	Z,C,N,V Z,C,N,V Z,C,N,V Z,C,N,V None SREG(s) T None C C N N X	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
LSL LSR ROL ROR ASR SWAP BSET BCLR BST BLD SEC CLC SEN CLN SEZ CLZ	Rd Rf	Logical Shift Right Rotate Left Through Carry Rotate Right Through Carry Arithmetic Shift Right Swap Nibbles Flag Set Flag Clear Bit Store from Register to T Bit load from T to Register Set Carry Clear Carry Set Negative Flag Clear Negative Flag Set Zero Flag Clear Zero Flag	$ \begin{array}{l} Rd(n) \leftarrow Rd(n+1), Rd(7) \leftarrow 0 \\ Rd(0) \leftarrow C, Rd(n+1) \leftarrow Rd(n), C \leftarrow Rd(7) \\ Rd(7) \leftarrow C, Rd(n) \leftarrow Rd(n+1), C \leftarrow Rd(0) \\ Rd(n) \leftarrow Rd(n+1), n=06 \\ Rd(30) \leftarrow Rd(74), Rd(74) \leftarrow Rd(30) \\ SREG(s) \leftarrow 1 \\ SREG(s) \leftarrow 0 \\ T \leftarrow Rr(b) \\ Rd(b) \leftarrow T \\ C \leftarrow 1 \\ C \leftarrow 0 \\ N \leftarrow 1 \\ N \leftarrow 0 \\ Z \leftarrow 1 \\ Z \leftarrow 0 \end{array} $	Z,C,N,V Z,C,N,V Z,C,N,V Z,C,N,V None SREG(s) SREG(s) T None C C N	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
LSL LSR ROL ROR ASR SWAP BSET BCLR BST BLD SEC CLC SEN CLN SEZ CLZ SEI	Rd Rf	Logical Shift Right Rotate Left Through Carry Rotate Right Through Carry Arithmetic Shift Right Swap Nibbles Flag Set Flag Clear Bit Store from Register to T Bit load from T to Register Set Carry Clear Carry Set Negative Flag Clear Negative Flag Set Zero Flag Clear Zero Flag Global Interrupt Enable	$ \begin{array}{c} Rd(n) \leftarrow Rd(n+1), Rd(7) \leftarrow 0 \\ Rd(0) \leftarrow C, Rd(n+1) \leftarrow Rd(n), C \leftarrow Rd(7) \\ Rd(7) \leftarrow C, Rd(n) \leftarrow Rd(n+1), C \leftarrow Rd(0) \\ Rd(n) \leftarrow Rd(n+1), n=06 \\ Rd(30) \leftarrow Rd(74), Rd(74) \leftarrow Rd(30) \\ SREG(s) \leftarrow 1 \\ SREG(s) \leftarrow 0 \\ T \leftarrow Rr(b) \\ Rd(b) \leftarrow T \\ C \leftarrow 1 \\ C \leftarrow 0 \\ N \leftarrow 1 \\ N \leftarrow 0 \\ Z \leftarrow 1 \\ Z \leftarrow 0 \\ I \leftarrow 1 \end{array} $	Z,C,N,V Z,C,N,V Z,C,N,V Z,C,N,V None SREG(s) T None C C N N X	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
LSL LSR ROL ROR ASR SWAP BSET BCLR BST BLD SEC CLC SEN CLN SEZ CLZ SEI CLI	Rd Rf	Logical Shift Right Rotate Left Through Carry Rotate Right Through Carry Arithmetic Shift Right Swap Nibbles Flag Set Flag Clear Bit Store from Register to T Bit load from T to Register Set Carry Clear Carry Set Negative Flag Clear Negative Flag Set Zero Flag Global Interrupt Enable Global Interrupt Disable	$ \begin{array}{c} Rd(n) \leftarrow Rd(n+1), Rd(7) \leftarrow 0 \\ Rd(0) \leftarrow C, Rd(n+1) \leftarrow Rd(n), C \leftarrow Rd(7) \\ Rd(7) \leftarrow C, Rd(n) \leftarrow Rd(n+1), C \leftarrow Rd(0) \\ Rd(n) \leftarrow Rd(n+1), n=06 \\ Rd(30) \leftarrow Rd(74), Rd(74) \leftarrow Rd(30) \\ SREG(s) \leftarrow 1 \\ SREG(s) \leftarrow 0 \\ T \leftarrow Rr(b) \\ Rd(b) \leftarrow T \\ C \leftarrow 1 \\ C \leftarrow 0 \\ N \leftarrow 1 \\ N \leftarrow 0 \\ Z \leftarrow 1 \\ Z \leftarrow 0 \\ I \leftarrow 1 \\ I \leftarrow 0 \\ \end{array} $	Z,C,N,V Z,C,N,V Z,C,N,V Z,C,N,V None SREG(s) T None C C N N X Z I I	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
LSL LSR ROL ROR ASR SWAP BSET BCLR BST BLD SEC CLC SEN CLN SEZ CLZ SEI CLI SES	Rd Rf	Logical Shift Right Rotate Left Through Carry Rotate Right Through Carry Arithmetic Shift Right Swap Nibbles Flag Set Flag Clear Bit Store from Register to T Bit load from T to Register Set Carry Clear Carry Set Negative Flag Clear Negative Flag Set Zero Flag Global Interrupt Enable Global Interrupt Disable Set Signed Test Flag	Rd(n) ← Rd(n+1), Rd(7) ← 0  Rd(0)←C,Rd(n+1)← Rd(n),C←Rd(7)  Rd(7)←C,Rd(n)← Rd(n+1),C←Rd(0)  Rd(n) ← Rd(n+1), n=06  Rd(30)←Rd(74),Rd(74)←Rd(30)  SREG(s) ← 1  SREG(s) ← 0 $T \leftarrow Rr(b)$ Rd(b) ← $T$ $C \leftarrow 1$ $C \leftarrow 0$ $N \leftarrow 1$ $N \leftarrow 0$ $Z \leftarrow 1$ $Z \leftarrow 0$ $I \leftarrow 1$ $I \leftarrow 0$ $S \leftarrow 1$	Z,C,N,V Z,C,N,V Z,C,N,V Z,C,N,V None SREG(s) T None C C N N X Z I I S	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
LSL LSR ROL ROR ASR SWAP BSET BCLR BST BLD SEC CLC SEN CLN SEZ CLZ SEI CLI SES CLS	Rd Rf	Logical Shift Right Rotate Left Through Carry Rotate Right Through Carry Arithmetic Shift Right Swap Nibbles Flag Set Flag Clear Bit Store from Register to T Bit load from T to Register Set Carry Clear Carry Set Negative Flag Clear Negative Flag Clear Zero Flag Global Interrupt Enable Global Interrupt Disable Set Signed Test Flag Clear Signed Test Flag	$ \begin{array}{c} Rd(n) \leftarrow Rd(n+1), Rd(7) \leftarrow 0 \\ Rd(0) \leftarrow C, Rd(n+1) \leftarrow Rd(n), C \leftarrow Rd(7) \\ Rd(7) \leftarrow C, Rd(n) \leftarrow Rd(n+1), C \leftarrow Rd(0) \\ Rd(n) \leftarrow Rd(n+1), n=06 \\ Rd(30) \leftarrow Rd(74), Rd(74) \leftarrow Rd(30) \\ SREG(s) \leftarrow 1 \\ SREG(s) \leftarrow 0 \\ T \leftarrow Rr(b) \\ Rd(b) \leftarrow T \\ C \leftarrow 1 \\ C \leftarrow 0 \\ N \leftarrow 1 \\ N \leftarrow 0 \\ Z \leftarrow 1 \\ Z \leftarrow 0 \\ I \leftarrow 1 \\ I \leftarrow 0 \\ S \leftarrow 1 \\ S \leftarrow 0 \\ \end{array} $	Z,C,N,V Z,C,N,V Z,C,N,V Z,C,N,V None SREG(s) T None C C N N X Z I I S S S	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
LSL LSR ROL ROR ASR SWAP BSET BCLR BST BLD SEC CLC SEN CLN SEZ CLZ SEI CLI SES CLS SES	Rd Rf	Logical Shift Right Rotate Left Through Carry Rotate Right Through Carry Arithmetic Shift Right Swap Nibbles Flag Set Flag Clear Bit Store from Register to T Bit load from T to Register Set Carry Clear Carry Set Negative Flag Clear Negative Flag Clear Zero Flag Global Interrupt Enable Global Interrupt Disable Set Signed Test Flag Clear Signed Test Flag Set Twos Complement Overflow	Rd(n) ← Rd(n+1), Rd(7) ← 0  Rd(0)←C,Rd(n+1)← Rd(n),C←Rd(7)  Rd(7)←C,Rd(n)← Rd(n+1),C←Rd(0)  Rd(n) ← Rd(n+1), n=06  Rd(30)←Rd(74),Rd(74)←Rd(30)  SREG(s) ← 1  SREG(s) ← 0 $T \leftarrow Rr(b)$ Rd(b) ← $T$ $C \leftarrow 1$ $C \leftarrow 0$ $N \leftarrow 1$ $N \leftarrow 0$ $Z \leftarrow 1$ $Z \leftarrow 0$ $  \leftarrow 1$ $  \leftarrow 0$ $  \leftarrow 1$ $  \leftarrow 0$ $  \leftarrow 1$ $  \leftarrow 0$ $  \leftarrow 0$ $  \leftarrow 1$ $  \leftarrow 0$ $  \leftarrow 1$ $  \leftarrow 0$ $  \leftarrow 0$ $  \leftarrow 1$ $  \leftarrow 0$ $  \leftarrow 1$ $  \leftarrow 0$ $  \leftarrow 0$ $  \leftarrow 1$ $  \leftarrow 0$	Z,C,N,V Z,C,N,V Z,C,N,V Z,C,N,V None SREG(s) T None C N N Z Z I I S S V	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
LSL LSR ROL ROR ASR SWAP BSET BCLR BST BLD SEC CLC SEN CLN SEZ CLZ SEI CLI SES CLS SES CLS	Rd Rf	Logical Shift Right Rotate Left Through Carry Rotate Right Through Carry Arithmetic Shift Right Swap Nibbles Flag Set Flag Clear Bit Store from Register to T Bit load from T to Register Set Carry Clear Carry Set Negative Flag Clear Negative Flag Set Zero Flag Global Interrupt Enable Global Interrupt Disable Set Signed Test Flag Clear Signed Test Flag Set Twos Complement Overflow Clear Twos Complement Overflow	$ \begin{array}{c} Rd(n) \leftarrow Rd(n+1), Rd(7) \leftarrow 0 \\ Rd(0) \leftarrow C, Rd(n+1) \leftarrow Rd(n), C \leftarrow Rd(7) \\ Rd(7) \leftarrow C, Rd(n) \leftarrow Rd(n+1), C \leftarrow Rd(0) \\ Rd(n) \leftarrow Rd(n+1), n=06 \\ Rd(30) \leftarrow Rd(74), Rd(74) \leftarrow Rd(30) \\ SREG(s) \leftarrow 1 \\ SREG(s) \leftarrow 0 \\ T \leftarrow Rr(b) \\ Rd(b) \leftarrow T \\ C \leftarrow 1 \\ C \leftarrow 0 \\ N \leftarrow 1 \\ N \leftarrow 0 \\ Z \leftarrow 1 \\ Z \leftarrow 0 \\ I \leftarrow 1 \\ I \leftarrow 0 \\ S \leftarrow 1 \\ S \leftarrow 0 \\ V \leftarrow 1 \\ V \leftarrow 0 \\ \end{array} $	Z,C,N,V Z,C,N,V Z,C,N,V Z,C,N,V None SREG(s) T None C C N N Z I I S S S V V	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
LSL LSR ROL ROR ASR SWAP BSET BCLR BST BLD SEC CLC SEN CLN SEZ CLZ SEI CLI SES CLS SES CLS SES CLS SES CLS SEV CLV SET	Rd Rf	Logical Shift Right Rotate Left Through Carry Rotate Right Through Carry Arithmetic Shift Right Swap Nibbles Flag Set Flag Clear Bit Store from Register to T Bit load from T to Register Set Carry Clear Carry Clear Carry Set Negative Flag Clear Negative Flag Set Zero Flag Global Interrupt Enable Global Interrupt Disable Set Signed Test Flag Clear Signed Test Flag Set Twos Complement Overflow Clear Twos Complement Overflow Set T in SREG	$ \begin{array}{c} Rd(n) \leftarrow Rd(n+1), Rd(7) \leftarrow 0 \\ Rd(0) \leftarrow C, Rd(n+1) \leftarrow Rd(n), C \leftarrow Rd(7) \\ Rd(7) \leftarrow C, Rd(n) \leftarrow Rd(n+1), C \leftarrow Rd(0) \\ Rd(n) \leftarrow Rd(n+1), n=06 \\ Rd(30) \leftarrow Rd(74), Rd(74) \leftarrow Rd(30) \\ SREG(s) \leftarrow 1 \\ SREG(s) \leftarrow 0 \\ T \leftarrow Rr(b) \\ Rd(b) \leftarrow T \\ C \leftarrow 1 \\ C \leftarrow 0 \\ N \leftarrow 1 \\ N \leftarrow 0 \\ Z \leftarrow 1 \\ Z \leftarrow 0 \\ I \leftarrow 1 \\ I \leftarrow 0 \\ S \leftarrow 1 \\ S \leftarrow 0 \\ V \leftarrow 1 \\ V \leftarrow 0 \\ T \leftarrow 1 \end{array} $	Z,C,N,V Z,C,N,V Z,C,N,V Z,C,N,V None SREG(s) T None C C N N Z I I S S V V T	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
LSL LSR ROL ROR ASR SWAP BSET BCLR BST BLD SEC CLC SEN CLN SEZ CLZ SEI CLI SES CLS SES CLS SES CLS SEV CLV SET CLT	Rd Rf	Logical Shift Right Rotate Left Through Carry Rotate Right Through Carry Arithmetic Shift Right Swap Nibbles Flag Set Flag Clear Bit Store from Register to T Bit load from T to Register Set Carry Clear Carry Clear Carry Set Negative Flag Clear Negative Flag Set Zero Flag Global Interrupt Enable Global Interrupt Disable Set Signed Test Flag Clear Signed Test Flag Set Twos Complement Overflow Clear Twos Complement Overflow Set T in SREG Clear T in SREG	$ \begin{array}{l} Rd(n) \leftarrow Rd(n+1), Rd(7) \leftarrow 0 \\ Rd(0) \leftarrow C, Rd(n+1) \leftarrow Rd(n), C \leftarrow Rd(7) \\ Rd(7) \leftarrow C, Rd(n) \leftarrow Rd(n+1), C \leftarrow Rd(0) \\ Rd(n) \leftarrow Rd(n+1), n=06 \\ Rd(30) \leftarrow Rd(74), Rd(74) \leftarrow Rd(30) \\ SREG(s) \leftarrow 1 \\ SREG(s) \leftarrow 0 \\ T \leftarrow Rr(b) \\ Rd(b) \leftarrow T \\ C \leftarrow 1 \\ C \leftarrow 0 \\ N \leftarrow 1 \\ N \leftarrow 0 \\ Z \leftarrow 1 \\ Z \leftarrow 0 \\ I \leftarrow 1 \\ I \leftarrow 0 \\ S \leftarrow 1 \\ C \leftarrow 0 \\ V \leftarrow 1 \\ V \leftarrow 0 \\ T \leftarrow 1 \\ T \leftarrow 0 \end{array} $	Z,C,N,V Z,C,N,V Z,C,N,V Z,C,N,V None SREG(s) T None C N N X Z I I I S S V V V T T	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
LSL LSR ROL ROR ASR SWAP BSET BCLR BST BLD SEC CLC SEN CLN SEZ CLZ SEI CLI SES CLS SES CLS SES CLS SEV CLV SET CLT SEH	Rd Rf	Logical Shift Right Rotate Left Through Carry Rotate Right Through Carry Arithmetic Shift Right Swap Nibbles Flag Set Flag Clear Bit Store from Register to T Bit load from T to Register Set Carry Clear Carry Clear Carry Set Negative Flag Clear Negative Flag Set Zero Flag Clear Zero Flag Global Interrupt Enable Global Interrupt Disable Set Signed Test Flag Set Twos Complement Overflow Clear Twos Complement Overflow Set T in SREG Clear T in SREG Set Half Carry Flag in SREG	$ \begin{array}{c} Rd(n) \leftarrow Rd(n+1), Rd(7) \leftarrow 0 \\ Rd(0) \leftarrow C, Rd(n+1) \leftarrow Rd(n), C \leftarrow Rd(7) \\ Rd(7) \leftarrow C, Rd(n) \leftarrow Rd(n+1), C \leftarrow Rd(0) \\ Rd(n) \leftarrow Rd(n+1), n=06 \\ Rd(30) \leftarrow Rd(74), Rd(74) \leftarrow Rd(30) \\ SREG(s) \leftarrow 1 \\ SREG(s) \leftarrow 0 \\ T \leftarrow Rr(b) \\ Rd(b) \leftarrow T \\ C \leftarrow 1 \\ C \leftarrow 0 \\ N \leftarrow 1 \\ N \leftarrow 0 \\ Z \leftarrow 1 \\ Z \leftarrow 0 \\ I \leftarrow 1 \\ I \leftarrow 0 \\ S \leftarrow 1 \\ S \leftarrow 0 \\ V \leftarrow 1 \\ V \leftarrow 0 \\ T \leftarrow 1 \\ T \leftarrow 0 \\ H \leftarrow 1 \end{array} $	Z,C,N,V Z,C,N,V Z,C,N,V Z,C,N,V None SREG(s) T None C C N N Z Z I I S S V V V T T H	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
LSL LSR ROL ROR ASR SWAP BSET BCLR BST BLD SEC CLC SEN CLN SEZ CLZ SEI CLI SES CLS SEV CLV SET CLT SEH CLH	Rd Rf	Logical Shift Right Rotate Left Through Carry Rotate Right Through Carry Arithmetic Shift Right Swap Nibbles Flag Set Flag Clear Bit Store from Register to T Bit load from T to Register Set Carry Clear Carry Clear Carry Set Negative Flag Clear Negative Flag Set Zero Flag Clear Zero Flag Global Interrupt Enable Global Interrupt Disable Set Signed Test Flag Set Twos Complement Overflow Clear Twos Complement Overflow Set T in SREG Clear T in SREG Set Half Carry Flag in SREG Clear Half Carry Flag in SREG	$ \begin{array}{l} Rd(n) \leftarrow Rd(n+1), Rd(7) \leftarrow 0 \\ Rd(0) \leftarrow C, Rd(n+1) \leftarrow Rd(n), C \leftarrow Rd(7) \\ Rd(7) \leftarrow C, Rd(n) \leftarrow Rd(n+1), C \leftarrow Rd(0) \\ Rd(n) \leftarrow Rd(n+1), n=06 \\ Rd(30) \leftarrow Rd(74), Rd(74) \leftarrow Rd(30) \\ SREG(s) \leftarrow 1 \\ SREG(s) \leftarrow 0 \\ T \leftarrow Rr(b) \\ Rd(b) \leftarrow T \\ C \leftarrow 1 \\ C \leftarrow 0 \\ N \leftarrow 1 \\ N \leftarrow 0 \\ Z \leftarrow 1 \\ Z \leftarrow 0 \\ I \leftarrow 1 \\ I \leftarrow 0 \\ S \leftarrow 1 \\ C \leftarrow 0 \\ V \leftarrow 1 \\ V \leftarrow 0 \\ T \leftarrow 1 \\ T \leftarrow 0 \end{array} $	Z,C,N,V Z,C,N,V Z,C,N,V Z,C,N,V None SREG(s) T None C N N Z Z I I S S V V V T T H H	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
LSL LSR ROL ROR ASR SWAP BSET BCLR BST BLD SEC CLC SEN CLN SEZ CLZ SEI CLI SES CLS SEV CLV SET CLT SEH CLH NOP	Rd Rf	Logical Shift Right Rotate Left Through Carry Rotate Right Through Carry Arithmetic Shift Right Swap Nibbles Flag Set Flag Clear Bit Store from Register to T Bit load from T to Register Set Carry Clear Carry Clear Carry Set Negative Flag Clear Negative Flag Set Zero Flag Global Interrupt Enable Global Interrupt Disable Set Signed Test Flag Set Twos Complement Overflow Clear Twos Complement Overflow Set T in SREG Clear Half Carry Flag in SREG Clear Half Carry Flag in SREG No Operation	$ \begin{array}{l} Rd(n) \leftarrow Rd(n+1), Rd(7) \leftarrow 0 \\ Rd(0) \leftarrow C, Rd(n+1) \leftarrow Rd(n), C \leftarrow Rd(7) \\ Rd(7) \leftarrow C, Rd(n) \leftarrow Rd(n+1), C \leftarrow Rd(0) \\ Rd(n) \leftarrow Rd(n+1), n=06 \\ Rd(30) \leftarrow Rd(74), Rd(74) \leftarrow Rd(30) \\ SREG(s) \leftarrow 1 \\ SREG(s) \leftarrow 0 \\ T \leftarrow Rr(b) \\ Rd(b) \leftarrow T \\ C \leftarrow 1 \\ C \leftarrow 0 \\ N \leftarrow 1 \\ N \leftarrow 0 \\ Z \leftarrow 1 \\ Z \leftarrow 0 \\ I \leftarrow 1 \\ I \leftarrow 0 \\ S \leftarrow 1 \\ S \leftarrow 0 \\ V \leftarrow 1 \\ V \leftarrow 0 \\ T \leftarrow 1 \\ T \leftarrow 0 \\ H \leftarrow 1 \\ H \leftarrow 0 \\ \end{array} $	Z,C,N,V Z,C,N,V Z,C,N,V Z,C,N,V None SREG(s) T None C C N N Z Z I I S S V V V T T H H None	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
LSL LSR ROL ROR ASR SWAP BSET BCLR BST BLD SEC CLC SEN CLN SEZ CLZ SEI CLI SES CLS SEV CLV SET CLT SEH CLH	Rd Rf	Logical Shift Right Rotate Left Through Carry Rotate Right Through Carry Arithmetic Shift Right Swap Nibbles Flag Set Flag Clear Bit Store from Register to T Bit load from T to Register Set Carry Clear Carry Clear Carry Set Negative Flag Clear Negative Flag Set Zero Flag Clear Zero Flag Global Interrupt Enable Global Interrupt Disable Set Signed Test Flag Set Twos Complement Overflow Clear Twos Complement Overflow Set T in SREG Clear T in SREG Set Half Carry Flag in SREG Clear Half Carry Flag in SREG	$ \begin{array}{c} Rd(n) \leftarrow Rd(n+1), Rd(7) \leftarrow 0 \\ Rd(0) \leftarrow C, Rd(n+1) \leftarrow Rd(n), C \leftarrow Rd(7) \\ Rd(7) \leftarrow C, Rd(n) \leftarrow Rd(n+1), C \leftarrow Rd(0) \\ Rd(n) \leftarrow Rd(n+1), n=06 \\ Rd(30) \leftarrow Rd(74), Rd(74) \leftarrow Rd(30) \\ SREG(s) \leftarrow 1 \\ SREG(s) \leftarrow 0 \\ T \leftarrow Rr(b) \\ Rd(b) \leftarrow T \\ C \leftarrow 1 \\ C \leftarrow 0 \\ N \leftarrow 1 \\ N \leftarrow 0 \\ Z \leftarrow 1 \\ Z \leftarrow 0 \\ I \leftarrow 1 \\ I \leftarrow 0 \\ S \leftarrow 1 \\ S \leftarrow 0 \\ V \leftarrow 1 \\ V \leftarrow 0 \\ T \leftarrow 1 \\ T \leftarrow 0 \\ H \leftarrow 1 \end{array} $	Z,C,N,V Z,C,N,V Z,C,N,V Z,C,N,V None SREG(s) T None C N N Z Z I I S S V V V T T H H	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1



# MS-012AA Package dimensions







	D	E1	Н	b	е	α	Α	A1	С	L	h
mm											
min	4.80	3.80	5.80	0.33		0°	1.35	0.10	0.19	0.41	0.25
max	5.00	4.00	6.20	0.51	1.27	8°	1.75	0.25	0.25	1.27	0.50
inches											
min	0.1890	0.1497	0.2284	0.013		0°	0.0532	0.0040	0.0075	0.016	0.0099
max	0.1968	0.1574	0.2440	0.020	0.100	8°	0.0688	0.0090	0.0098	0.050	0.0196