

5. AVR Assembly Instruction Set

Operand Notation

Before diving into the instructions, here are the common operand symbols used:

Symbol	Description
Rd	Destination register (R0-R31). The result of the operation is stored here.
Rr	Source register (R0-R31). Used as input for the operation.
K	Constant/Immediate value (8-bit: 0-255 or 0x00-0xFF).
k	Address constant for SRAM or program memory.
A	I/O register address (0-63 for IN/OUT, 0-31 for SBI/CBI).
b	Bit number (0-7) within a register or I/O address.
X, Y, Z	Pointer registers (X=R27:R26, Y=R29:R28, Z=R31:R30).

Note: Some instructions only work with upper registers (R16-R31), such as `LDI`, `ANDI`, `ORI`, `SUBI`, `SBCI`, and `CPI`.

A. Data Transfer Instructions

Used to move data between registers or between registers and memory/I/O.

Mnemonic	Operand	Description	Example	Notes
LDI	Rd, K	Load Immediate	<code>LDI R16, 0xFF</code>	Loads 8-bit constant K into register Rd (R16-R31 only)
MOV	Rd, Rr	Move/Copy Register	<code>MOV R0, R1</code>	Copies contents of register Rr to Rd
IN	Rd, A	Input from I/O	<code>IN R16, PINB</code>	Reads data from I/O port A to register Rd
OUT	A, Rr	Output to I/O	<code>OUT PORTB, R16</code>	Sends data from register Rr to I/O port A
LDS	Rd, k	Load from SRAM	<code>LDS R16, 0x0100</code>	Loads data from SRAM address k to register Rd

Mnemonic	Operand	Description	Example	Notes
STS	k, Rr	Store to SRAM	STS 0x0100, R16	Stores register Rr contents to SRAM address k
LD	Rd, X/Y/Z	Load Indirect	LD R16, X	Loads data from address pointed by pointer X/Y/Z
ST	X/Y/Z, Rr	Store Indirect	ST X, R16	Stores data to address pointed by pointer X/Y/Z
PUSH	Rr	Push to Stack	PUSH R16	Saves register to stack
POP	Rd	Pop from Stack	POP R16	Retrieves data from stack to register

B. Bit Manipulation Instructions (I/O Specific)

These instructions operate on the lower 32 I/O addresses (\$00-\$1F). Very efficient for changing one bit without affecting other bits.

Mnemonic	Operand	Description	Example	Notes
SBI	A, b	Set Bit in I/O	SBI DDRB, 5	Sets bit b in I/O register A to 1
CBI	A, b	Clear Bit in I/O	CBI PORTB, 5	Clears bit b in I/O register A to 0
BST	Rr, b	Bit Store to T	BST R16, 3	Copies bit b from register Rr to T flag
BLD	Rd, b	Bit Load from T	BLD R17, 5	Copies T flag to bit b of register Rd

C. Arithmetic Instructions

Mnemonic	Operand	Description	Example	Notes
ADD	Rd, Rr	Add	ADD R1, R2	$Rd = Rd + Rr$
ADC	Rd, Rr	Add with Carry	ADC R1, R2	$Rd = Rd + Rr + C$ (Carry flag)
SUB	Rd, Rr	Subtract	SUB R16, R17	$Rd = Rd - Rr$
SBC	Rd, Rr	Subtract with Carry	SBC R16, R17	$Rd = Rd - Rr - C$
SUBI	Rd, K	Subtract Immediate	SUBI R16, 10	$Rd = Rd - K$ (R16-R31 only)

Mnemonic	Operand	Description	Example	Notes
SBCI	Rd, K	Subtract Immediate with Carry	SBCI R17, 0	$Rd = Rd - K - C$
INC	Rd	Increment	INC R16	$Rd = Rd + 1$
DEC	Rd	Decrement	DEC R16	$Rd = Rd - 1$
MUL	Rd, Rr	Multiply Unsigned	MUL R16, R17	$R1:R0 = Rd \times Rr$ (16-bit result)
MULS	Rd, Rr	Multiply Signed	MULS R16, R17	$R1:R0 = Rd \times Rr$ (signed)
NEG	Rd	Negate (Two's Complement)	NEG R16	$Rd = 0x00 - Rd$

D. Logic Instructions

Mnemonic	Operand	Description	Example	Notes
AND	Rd, Rr	Logical AND	AND R1, R2	$Rd = Rd \text{ AND } Rr$
ANDI	Rd, K	AND Immediate	ANDI R16, 0x0F	$Rd = Rd \text{ AND } K$ (masking)
OR	Rd, Rr	Logical OR	OR R1, R2	$Rd = Rd \text{ OR } Rr$
ORI	Rd, K	OR Immediate	ORI R16, 0x80	$Rd = Rd \text{ OR } K$
EOR	Rd, Rr	Exclusive OR	EOR R16, R17	$Rd = Rd \text{ XOR } Rr$
COM	Rd	One's Complement	COM R16	$Rd = 0xFF - Rd$ (inverts all bits)
CLR	Rd	Clear Register	CLR R16	$Rd = 0$ (same as EOR Rd, Rd)
SER	Rd	Set Register	SER R16	$Rd = 0xFF$ (R16-R31 only)

E. Shift & Rotate Instructions

Mnemonic	Operand	Description	Example	Notes
LSL	Rd	Logical Shift Left	LSL R16	Shift left, bit 0 = 0, bit 7 → Carry
LSR	Rd	Logical Shift Right	LSR R16	Shift right, bit 7 = 0, bit 0 → Carry
ROL	Rd	Rotate Left through Carry	ROL R16	Rotate left through Carry flag

Mnemonic	Operand	Description	Example	Notes
ROR	Rd	Rotate Right through Carry	<code>ROR R16</code>	Rotate right through Carry flag
ASR	Rd	Arithmetic Shift Right	<code>ASR R16</code>	Shift right, bit 7 remains (preserve sign)
SWAP	Rd	Swap Nibbles	<code>SWAP R16</code>	Swaps upper and lower 4-bits in register

F. Branch & Control Flow Instructions

Mnemonic	Operand	Description	Example	Notes
RJMP	k	Relative Jump	<code>RJMP loop</code>	Jump to label k ($\pm 2K$ words)
JMP	k	Jump	<code>JMP far_label</code>	Jump to 22-bit address (all memory)
RCALL	k	Relative Call	<code>RCALL delay</code>	Call subroutine relative to PC
CALL	k	Call	<code>CALL far_sub</code>	Call subroutine at 22-bit address
RET	-	Return	<code>RET</code>	Return from subroutine
RETI	-	Return from Interrupt	<code>RETI</code>	Return from interrupt handler
CP	Rd, Rr	Compare	<code>CP R16, R17</code>	Compare Rd with Rr (updates flags)
CPI	Rd, K	Compare Immediate	<code>CPI R16, 5</code>	Compare Rd with constant K
CPC	Rd, Rr	Compare with Carry	<code>CPC R17, R19</code>	For multi-byte comparison
BREQ	k	Branch if Equal	<code>BREQ target</code>	Jump if Z flag = 1 (result equal)
BRNE	k	Branch if Not Equal	<code>BRNE loop</code>	Jump if Z flag = 0 (result not equal)
BRLO	k	Branch if Lower	<code>BRLO less</code>	Jump if C flag = 1 (unsigned <)
BRSH	k	Branch if Same or Higher	<code>BRSH greater</code>	Jump if C flag = 0 (unsigned \geq)
BRLT	k	Branch if Less Than	<code>BRLT neg</code>	Jump if S flag = 1 (signed <)

Mnemonic	Operand	Description	Example	Notes
BRGE	k	Branch if Greater or Equal	<code>BRGE pos</code>	Jump if S flag = 0 (signed \geq)

G. Skip Instructions

Mnemonic	Operand	Description	Example	Notes
SBIS	A, b	Skip if Bit in I/O Set	<code>SBIS PINB, 0</code>	Skip next instruction if bit = 1
SBIC	A, b	Skip if Bit in I/O Cleared	<code>SBIC PIND, 2</code>	Skip next instruction if bit = 0
SBRS	Rr, b	Skip if Bit in Register Set	<code>SBRS R16, 7</code>	Skip if bit b in register = 1
SBRC	Rr, b	Skip if Bit in Register Cleared	<code>SBRC R16, 0</code>	Skip if bit b in register = 0

H. Other Instructions

Mnemonic	Operand	Description	Example	Notes
NOP	-	No Operation	<code>NOP</code>	Does nothing (1 clock cycle)
SLEEP	-	Sleep	<code>SLEEP</code>	Enters sleep mode (power saving)
WDR	-	Watchdog Reset	<code>WDR</code>	Resets watchdog timer
SBIW	Rd, K	Subtract Immediate from Word	<code>SBIW R24, 1</code>	Subtract K from 16-bit value (R25:R24)
ADIW	Rd, K	Add Immediate to Word	<code>ADIW R24, 1</code>	Add K to 16-bit value

Revision #1

Created 2026-02-04 06:54:58 UTC by AX

Updated 2026-02-04 06:55:11 UTC by AX