Instruction Set Quick Reference

instruction	Operand	description	operating	Flag	cycle
Arithmetic and logic instructions					
ADD	Rd, Rr	Adding register	Rd ← Rd + Rr	Z, C, N, V, H	1
ADC	Rd, Rr	Adding the carry bit register	Rd ← Rd + Rr + C	Z, C, N, V, H	1
ADIW	Rdl, K	Now the number is added to the word	Rdh: Rdl ← Rdh: Rdl + K	Z, C, N, V, S	1
SUB	Rd, Rr	Add and subtract registers	Rd ← Rd - Rr	Z, C, N, V, H	1
SUBI	Rd, K	Constant Register Save	Rd ← Rd - K	Z, C, N, V, H	1
SBC	Rd, Rr	Register of addition and subtraction with borrow	Rd ← Rd - Rr - C	Z, C, N, V, H	1
SBCI	Rd, K	Save Register is constant with borrow	Rd ← Rd - K - C	Z, C, N, V, H	1
SBIW	Rdl, K	Subtract immediate word	Rdh: Rdl ← Rdh: Rdl - K	Z, C, N, V, S	1
AND	Rd, Rr	Logic and	Rd ← Rd & Rr	Z, N, V	1
ANDI	Rd, K	And a constant register logic	Rd ← Rd & K	Z, N, V	1
OR	Rd, Rr	Logical or	Rd ← Rd | Rr	Z, N, V	1
ORI	Rd, K	Or constant register logic	Rd ← Rd | K	Z, N, V	1
EOR	Rd, Rr	XOR register	Rd ← Rd ⊕ Rr	Z, N, V	1
COM	Rd	Inverted	Rd ← $ FF - Rd	Z, C, N, V	1
NEG	Rd	2 Ban complement	Rd ← $ 00 - Rd	Z, C, N, V, H	1
SBR	Rd, K	Setting register bit	Rd ← Rd v K	Z, N, V	1
CBR	Rd, K	Register bit clear	Rd ← Rd v ($ FF - K)	Z, N, V	1
INC	Rd	Increment	Rd ← Rd + 1	Z, N, V	1
DEC	Rd	Decreasing	Rd ← Rd - 1	Z, N, V	1
TST	Rd	Tests for 0 Or negative	Rd ← Rd & Rd	Z, N, V	1
CLR	Rd	Clear register	Rd ← Rd ⊕ Rd	Z, N, V	1
SER	Rd	Register are set to 1	Rd ← $ FF	None	1
MUL	Rd, Rr	Unsigned multiply	R1: R0 ← Rd x Rr	Z, C	1
MULS	Rd, Rr	Signed multiply	R1: R0 ← Rd x Rr	Z, C	1
MULSU	Rd, Rr	Signed unsigned multiplication	R1: R0 ← Rd x Rr	Z, C	1
FMUL	Rd, Rr	Unsigned multiplication, shift	R1: R0 ← (Rd x Rr) << 1	Z, C	1
FMULS	Rd, Rr	Signed multiply, shift	R1: R0 ← (Rd x Rr) << 1	Z, C	1
FMULSU	Rd, Rr	Signed unsigned multiplication, shift	R1: R0 ← (Rd x Rr) << 1	Z, C	1
Jump instructions					
RJMP	K	Relative jump	PC ← PC + K + 1	None	1
IJMP		Indirect jump (to Z At the address)	PC ← Z	None	2
JMP	K	Jump directly	PC ← K	None	2
RCALL	K	Relative subroutine call address	PC ← PC + K + 1	None	1
ICALL		Indirect subroutine call (Z At the address) PC ← Z		None	2
CALL	K	Direct subroutine call	PC ← K	None	2
RET		Subroutine returns	PC ← Stack	None	2
RETI		Interrupt return	PC ← Stack	I	2