Список команд

Mnemonics	Operands	Description	Operation	Flags	Clocks
ARITHMETIC AND LOGIC INSTRUCTIONS					
ADD	Rd, Rr	Add two Registers (all registers)	Rd ← Rd + Rr	Z,C,N,V,H,S	1
ADC	Rd, Rr	Add with Carry two Registers (all registers)	Rd ← Rd + Rr + C	Z,C,N,V,H,S	1
ADIW	Rdl, K	Add Immediate to Word (dl = 24, 26, 28, 30)	Rdh:Rdl ← Rdh:Rdl + K (K ≤ 63)	Z,C,N,V,S	2
SUB	Rd, Rr	Subtract two Registers (all registers)	Rd ← Rd – Rr	Z,C,N,V,H,S	1
SUBI	Rd, K	Subtract Constant from Register (d ≥16)	Rd ← Rd – K (K ≤ 255)	Z,C,N,V,H,S	1
SBC	Rd, Rr	Subtract with Carry two Registers (all registers)	Rd ← Rd – Rr – C	Z,C,N,V,H,S	1
SBCI	Rd, K	Subtract with Carry Constant from Reg. (d ≥16)	Rd ← Rd – K – C (K ≤ 255)	Z,C,N,V,H,S	1
SBIW	Rdl, K	Subtract Immediate from Word (dl = 24, 26, 28, 30)	Rdh:Rdl ← Rdh:Rdl – K (K ≤ 63)	Z,C,N,V,S	2
AND	Rd, Rr	Logical AND Registers (all registers)	Rd ← Rd • Rr	Z,N,V,S	1
ANDI	Rd, K	Logical AND Register and Constant (d ≥16)	Rd ← Rd • K (K ≤ 255)	Z,N,V,S	1
OR	Rd, Rr	Logical OR Registers (all registers)	Rd ← Rd ∨ Rr	Z,N,V,S	1
ORI	Rd, K	Logical OR Register and Constant (d ≥16)	Rd ← Rd ∨ K (K ≤ 255)	Z,N,V,S	1
EOR	Rd, Rr	Exclusive OR Registers (all registers)	Rd ← Rd ⊕ Rr	Z,N,V,S	1
COM	Rd	One's Complement (all registers)	Rd ← $FF – Rd	Z,C,N,V,S	1
NEG	Rd	Two's Complement (all registers)	Rd ← $00 – Rd	Z,C,N,V,H,S	1
SBR	Rd, K	Set Bit(s) in Register (d ≥16)	Rd ← Rd ∨ K (K ≤ 255)	Z,N,V,S	1
CBR	Rd, K	Clear Bit(s) in Register (d ≥16)	Rd ← Rd • ($FF – K) (K ≤ 255)	Z,N,V,S	1
INC	Rd	Increment (all registers)	Rd ← Rd + 1	Z,N,V,S	1
DEC	Rd	Decrement (all registers)	Rd ← Rd – 1	Z,N,V,S	1
TST	Rd	Test for Zero or Minus (all registers)	Rd ← Rd • Rd	Z,N,V,S	1
CLR	Rd	Clear Register (all registers)	Rd ← Rd ⊕ Rd	Z,N,V,S	1
SER	Rd	Set Register (d ≥16)	Rd ← $FF	None	1
MUL	Rd, Rr	Multiply Unsigned	R1 : R0 ← Rd × Rr	Z,C	2
MULS	Rd, Rr	Multiply Signed	R1 : R0 ← Rd × Rr	Z,C	2
MULSU	Rd, Rr	Multiply Signed with Unsigned	R1 : R0 ← Rd × Rr	Z,C	2
FMUL	Rd, Rr	Fractional Multiply Unsigned	R1 : R0 ← (Rd × Rr) << 1	Z,C	2
FMULS	Rd, Rr	Fractional Multiply Signed	R1 : R0 ← (Rd × Rr) << 1	Z,C	2
FMULSU	Rd, Rr	Fractional Multiply Signed with Unsigned	R1 : R0 ← (Rd × Rr) << 1	Z,C	2
BRANCH INSTRUCTIONS					
Mnemonics	Operands	Description	Operation	Flags	Clocks
RJMP	k	Relative Jump (–2K ≤ k ≤ 2K)	PC ← PC + k + 1	None	2
IJMP		Indirect Jump to (Z)	PC ← Z	None	2
RCALL	k	Relative Subroutine Call (–2K ≤ k ≤ 2K)	STACK ← PC + 1, PC ← PC + k + 1	None	3
ICALL		Indirect Call to (Z)	STACK ← PC + 1, PC ← Z	None	3
RET		Subroutine Return	PC ← STACK	None	4
RETI		Interrupt Return	PC ← STACK, I = 1	I	4
CPSE	Rd, Rr	Compare, Skip if Equal (all registers)	if (Rd = Rr) PC ← PC + 2 or 3	None	1 or 2 or 3
CP	Rd, Rr	Compare (all registers)	Rd – Rr	Z,N,V,C,H,S	1
CPC	Rd, Rr	Compare with Carry (all registers)	Rd – Rr – C	Z,N,V,C,H,S	1
CPI	Rd, K	Compare Register with Immediate (d ≥ 16)	Rd – K (K ≤ 255)	Z,N,V,C,H,S	1
SBRC	Rr, b	Skip if Bit in Register Cleared (r ≤ 31, b ≤ 7)	if (Rr(b) = 0) PC ← PC + 2 or 3	None	1 or 2 or 3
SBRS	Rr, b	Skip if Bit in Register Set (r ≤ 31, b ≤ 7)	if (Rr(b) = 1) PC ← PC + 2 or 3	None	1 or 2 or 3
SBIC	P, b	Skip if Bit in I/O Register Clear (P ≤ $1F, b ≤ 7)	if (P(b) = 0) PC ← PC + 2 or 3	None	1 or 2 or 3
SBIS	P, b	Skip if Bit in I/O Register Set (P ≤ $1F, b ≤ 7)	if (P(b) = 1) PC ← PC + 2 or 3	None	1 or 2 or 3
BRBS	b, k	Branch if Status Flag Set (–64 ≤ k ≤ 63, b ≤ 7)	if (SREG(b) = 1) then PC ← PC + k + 1	None	1 or 2
BRBC	b, k	Branch if Status Flag Cleared (–64 ≤ k ≤ 63, b ≤ 7)	if (SREG(b) = 0) then PC ← PC + k + 1	None	1 or 2
BREQ	k	Branch if Equal (–64 ≤ k ≤ 63)	if (Z = 1) then PC ← PC + k + 1	None	1 or 2
BRNE	k	Branch if Not Equal (–64 ≤ k ≤ 63)	if (Z = 0) then PC ← PC + k + 1	None	1 or 2
BRCS	k	Branch if Carry Set (–64 ≤ k ≤ 63)	if (C = 1) then PC ← PC + k + 1	None	1 or 2
BRCC	k	Branch if Carry Cleared (–64 ≤ k ≤ 63)	if (C = 0) then PC ← PC + k + 1	None	1 or 2
BRSH	k	Branch if Same or Higher (–64 ≤ k ≤ 63)	if (C = 0) then PC ← PC + k + 1	None	1 or 2
BRLO	k	Branch if Lower (–64 ≤ k ≤ 63)	if (C = 1) then PC ← PC + k + 1	None	1 or 2
BRMI	k	Branch if Minus (–64 ≤ k ≤ 63)	if (N = 1) then PC ← PC + k + 1	None	1 or 2
BRPL	k	Branch if Plus (–64 ≤ k ≤ 63)	if (N = 0) then PC ← PC + k + 1	None	1 or 2
BRGE	k	Branch if Greater or Equal, Signed (–64 ≤ k ≤ 63)	if (S = 1) then PC ← PC + k + 1	None	1 or 2
BRLT	k	Branch if Less Than Zero, Signed (–64 ≤ k ≤ 63)	if (S = 0) then PC ← PC + k + 1	None	1 or 2
BRHS	k	Branch if Half Carry Flag Set (–64 ≤ k ≤ 63)	if (H = 1) then PC ← PC + k + 1	None	1 or 2
BRHC	k	Branch if Half Carry Flag Cleared (–64 ≤ k ≤ 63)	if (H = 0) then PC ← PC + k + 1	None	1 or 2
BRTS	k	Branch if T Flag Set (–64 ≤ k ≤ 63)	if (T = 1) then PC ← PC + k + 1	None	1 or 2
BRTC	k	Branch if T Flag Cleared (–64 ≤ k ≤ 63)	if (T = 0) then PC ← PC + k + 1	None	1 or 2
BRVS	k	Branch if Overflow Flag Set (–64 ≤ k ≤ 63)	if (V = 1) then PC ← PC + k + 1	None	1 or 2
BRVC	k	Branch if Overflow Flag Cleared (–64 ≤ k ≤ 63)	if (V = 0) then PC ← PC + k + 1	None	1 or 2
BRIE	k	Branch if Interrupt Enable (–64 ≤ k ≤ 63)	if (I = 1) then PC ← PC + k + 1	None	1 or 2
BRID	k	Branch if Interrupt Disabled (–64 ≤ k ≤ 63)	if (I = 0) then PC ← PC + k + 1	None	1 or 2

DATA TRANSFER INSTRUCTIONS

MOV	Rd, Rr	Move Between Registers (all registers)	Rd ← Rr	None	1
MOVW	Rd, Rr	Copy Register Word	Rd + 1 : Rd ← Rr + 1 : Rr	None	1
LDI	Rd, K	Load Immediate (d ≥16)	Rd ← K (K ≤ 255)	None	1
LD	Rd, X	Load Indirect (all registers)	Rd ← (X)	None	2
LD	Rd, X+	Load Indirect and Post-Inc. (all registers)	Rd ← (X), X ← X + 1	None	2
LD	Rd, –X	Load Indirect and Pre-Dec. (all registers)	X ← X – 1, Rd ← (X)	None	2
LD	Rd, Y	Load Indirect (all registers)	Rd ← (Y)	None	2
LD	Rd, Y+	Load Indirect and Post-Inc. (all registers)	Rd ← (Y), Y ← Y + 1	None	2
LD	Rd, –Y	Load Indirect and Pre-Dec. (all registers)	Y ← Y – 1, Rd ← (Y)	None	2
LDD	Rd, Y+q	Load Indirect with Displacement (all regs)	Rd ← (Y + q) (q ≤ 63)	None	2
LD	Rd, Z	Load Indirect (all registers)	Rd ← (Z)	None	2
LD	Rd, Z+	Load Indirect and Post-Inc. (all registers)	Rd ← (Z), Z ← Z + 1	None	2
LD	Rd, –Z	Load Indirect and Pre-Dec. (all registers)	Z ← Z – 1, Rd ← (Z)	None	2
LDD	Rd, Z+q	Load Indirect with Displacement (all regs)	Rd ← (Z+q) (q ≤ 63)	None	2
LDS	Rd, k	Load Direct from SRAM (all registers)	Rd ← (k) (k ≤ 65535)	None	3
ST	X, Rr	Store Indirect (all registers)	(X) ← Rr	None	2
ST	X+, Rr	Store Indirect and Post-Inc. (all registers)	(X) ← Rr, X ← X + 1	None	2
ST	X–, Rr	Store Indirect and Pre-Dec. (all registers)	X ← X – 1, (X) ← Rr	None	2
ST	Y, Rr	Store Indirect (all registers)	(Y) ← Rr	None	2
ST	Y+, Rr	Store Indirect and Post-Inc. (all registers)	(Y) ← Rr, Y ← Y + 1	None	2
ST	Y–, Rr	Store Indirect and Pre-Dec. (all registers)	Y ← Y – 1, (Y) ← Rr	None	2
STD	Y+q, Rr	Store Indirect with Displacement (all regs)	(Y + q) ← Rr (q ≤ 63)	None	2
ST	Z, Rr	Store Indirect (all registers)	(Z) ← Rr	None	2
ST	Z+, Rr	Store Indirect and Post-Inc. (all registers)	(Z) ← Rr, Z ← Z+1	None	2
ST	Z–, Rr	Store Indirect and Pre-Dec. (all registers)	Z ← Z – 1, (Z) ← Rr	None	2
STD	Z+q, Rr	Store Indirect with Displacement (all regs)	(Y + q) ← Rr (q ≤ 63)	None	2
STS	k, Rr	Store Direct to SRAM (all registers)	(k) ← Rr, (k ≤ 65535)	None	2
LPM		Load Program Memory	R0 ← (Z)	None	3
LPM	Rd, Z	Load Program Memory	Rd ← (Z)		
LPM	Rd, Z+	Load Program Memory and Post-Inc	Rd ← (Z), Z ← Z + 1		
SPM		Store Program Memory	(Z) ← R1 : R0		
IN	Rd, P	In Port (all registers and all I/O registers)	Rd ← P	None	1
OUT	P, Rr	Out Port (all registers and all I/O registers)	P ← Rr	None	1
PUSH	Rr	Push Register on (all registers)	STACK ← Rr	None	2
POP	Rd	Pop Register from Stack (all registers)	Rd ← STACK	None	2

BIT AND BIT-TEST INSTRUCTIONS

SBI	P, b	Set Bit in I/O Register (P ≤ $1F)	I/O(P,b) ← 1 (b ≤ 7)	None	2
CBI	P, b	Clear Bit in I/O Register (P ≤ $1F)	I/O(P,b) ← 0 (b ≤ 7)	None	2
LSL	Rd	Logical Shift Left (all registers)	Rd(n + 1) ← Rd(n), Rd(0) ← 0 (n = 0..6)	Z,C,N,V,H	1
LSR	Rd	Logical Shift Right (all registers)	Rd(n) ← Rd(n + 1), Rd(7) ← 0 (n = 0..6)	Z,C,N,V,H	1
ROL	Rd	Rotate Left Through Carry (all registers)	Rd(0) ← C, Rd(n + 1) ← Rd(n), C ← Rd(7) (n = 0..6)	Z,C,N,V	1
ROR	Rd	Rotate Right Through Carry (all registers)	Rd(7) ← C, Rd(n) ← Rd(n + 1), C ← Rd(0) (n = 0..6)	Z,C,N,V	1
ASR	Rd	Arithmetic Shift Right (all registers)	Rd(n) ← Rd(n + 1) (n = 0..6)	Z,C,N,V	1
SWAP	Rd	Swap Nibbles (all registers)	Rd(3..0) ← Rd(7..4), Rd(7..4) ← Rd(3..0)	None	1
BSET	s	Flag Set	SREG(s) ← 1 (s ≤ 7)	SREG(s)	1
BCLR	s	Flag Clear	SREG(s) ← 0 (s ≤ 7)	SREG(s)	1
BST	Rr, b	Bit Store from Register to T (all registers)	T ← Rr(b) (b ≤ 7)	T	1
BLD	Rd, b	Bit Load from T to Register (all registers)	Rd(b) ← T (b ≤ 7)	None	1
SEC		Set Carry	C ← 1	C	1
CLC		Clear Carry	C ← 0	C	1
SEN		Set Negative Flag	N ← 1	N	1
CLN		Clear Negative Flag	N ← 0	N	1
SEZ		Set Zero Flag	Z ← 1	Z	1
CLZ		Clear Zero Flag	Z ← 0	Z	1
SEI		Global Interrupt Enable	I ← 1	I	1
CLI		Global Interrupt Disable	I ← 0	I	1
SES		Set Signed Test Flag	S ← 1	S	1
CLS		Clear Signed Test Flag	S ← 0	S	1
SEV		Set Twos Complement Overflow	V ← 1	V	1
CLV		Clear Twos Complement Overflow	V ← 0	V	1
SET		Set T in SREG	T ← 1	T	1
CLT		Clear T in SREG	T ← 0	T	1
SEN		Set Half Carry Flag in SREG	H ← 1	H	1
CLH		Clear Half Carry Flag in SREG	H ← 0	H	1

MCU CONTROL INSTRUCTIONS

NOP		No Operation		None	1
SLEEP		Sleep	(see specific description for Sleep function)	None	3
WDR		Watchdog timer Reset	(see specific description for WDR/timer)	None	1
BREAK		Break	For On-chip Debug Only	None	N/A