instruction Operand		description	operating	Flag cycle	
Jump instructions (cont.)					
CPSE	R_d, R_r	That jump is equal to	If (R_d = R_r) PC ← PC + 2 or 3	None	1/2
CP	R_d, R_r	Compare	R_d - R_r	Z, N, V, C, H	1
CPC	R_d, R_r	Carry compare	R_d - R_r - C	Z, N, V, C, H	1
CPI	R_d, K	Compared with the immediate	R_d - K	Z, N, V, C, H	1
SBRC	R_r, b	Bit 0 Skip next instruction	If (R_r (b) = 0) PC ← PC + 2 or 3	None	1/2
SBRS	R_r, b	Bit 1 Skip next instruction	If (R_r (b) = 1) PC ← PC + 2 or 3	None	1/2
SBIC	P, b	I / O Bit 0 Skip next instruction	If (P (b) = 0) PC ← PC + 2 or 3	None	1/2
SBIS	P, b	I / O Bit 1 Skip next instruction	If (P (b) = 1) PC ← PC + 2 or 3	None	1/2
BRBS	s, k	State marked 1 That jump	If (SREG (S) = 1) PC ← PC + K + 1	None	1/2
BRBC	s, k	State marked 0 That jump	If (SREG (S) = 0) PC ← PC + K + 1	None	1/2
BREQ	k	That jump is equal to	if (Z = 1) then PC ← PC + k + 1	None	1/2
BRNE	k	Range will jump	if (Z = 0) then PC ← PC + k + 1	None	1/2
BRCS	k	Carry Jump	if (C = 1) then PC ← PC + k + 1	None	1/2
BRCC	k	Not Carry Jump	if (C = 0) then PC ← PC + k + 1	None	1/2
BRSH	k	Not less than jump	if (C = 0) then PC ← PC + k + 1	None	1/2
BRLO	k	Less than jump	if (C = 1) then PC ← PC + k + 1	None	1/2
BRMI	k	Negative jump	if (N = 1) then PC ← PC + k + 1	None	1/2
BRPL	k	As a regular jump	if (N = 0) then PC ← PC + k + 1	None	1/2
BRGE	k	Signed i.e. jump is not less than	if (N ⊕ V = 0) then PC ← PC + k + 1	None	1/2
BRLT	k	Signed less than 0 That jump	if (N ⊕ V = 1) then PC ← PC + k + 1	None	1/2
BRHS	k	Half-carry is 1 Jump	if (H = 1) then PC ← PC + k + 1	None	1/2
BRHC	k	Half-carry is 0 Jump	if (H = 0) then PC ← PC + k + 1	None	1/2
BRTS	k	T Set Jump	if (T = 1) then PC ← PC + k + 1	None	1/2
BRTC	k	T Clear Jump	if (T = 0) then PC ← PC + k + 1	None	1/2
BRVS	k	Overflow jump	f (V = 1) then PC ← PC + k + 1	None	1/2
BRVC	k	Does not overflow jump	f (V = 0) then PC ← PC + k + 1	None	1/2
BRIE	k	Global Interrupt Enable jump	f (I = 1) then PC ← PC + k + 1	None	1/2
BRID	k	Global Interrupt Disable Jump	f (I = 0) then PC ← PC + k + 1	None	1/2
Data Transfer Instructions					
MOV	Rd, Rr	Move data between registers	Rd ← Rr	None	1
MOVW	Rd, Rr	Moving a data word	Rd + 1: Rd ← Rr + 1: Rr	None	1
LDI	Rd, K	Immediate loading	Rd ← K	None	1
LD	Rd, X	Indirect load	Rd ← (X)	None	1/2
LD	Rd, X +	Indirect load, the address is incremented	Rd ← (X), X ← X + 1	None	1/2
LD	Rd, -X	Address decrement, indirect load	X ← X - 1, Rd ← (X)	None	1/2
LD	Rd, Y	Indirect load	Rd ← (Y)	None	1/2
LD	Rd, Y +	Indirect load, the address is incremented	Rd ← (Y), Y ← Y + 1	None	1/2
LD	Rd, -Y	Address decrement, indirect load	Y ← Y - 1, Rd ← (Y)	None	1/2
LDD	Rd, Y + q	Indirect with offset loading	Rd ← (Y + q)	None	1/2
LD	Rd, Z	Indirect load	Rd ← (Z)	None	1/2