3. Conditional Branch Summary

Test	Boolean	Mnemonic	Complementary	Boolean	Mnemonic	Comment
Rd > Rr	Z•(N ⊕ V) = 0	BRLT(1)	Rd ≤ Rr	Z+(N ⊕ V) = 1	BRGE*	Signed
Rd ≥ Rr	(N ⊕ V) = 0	BRGE	Rd < Rr	(N ⊕ V) = 1	BRLT	Signed
Rd = Rr	Z = 1	BREQ	Rd ≠ Rr	Z = 0	BRNE	Signed
Rd ≤ Rr	Z+(N ⊕ V) = 1	BRGE(1)	Rd > Rr	Z•(N ⊕ V) = 0	BRLT*	Signed
Rd < Rr	(N ⊕ V) = 1	BRLT	Rd ≥ Rr	(N ⊕ V) = 0	BRGE	Signed
Rd > Rr	C + Z = 0	BRLO(1)	Rd ≤ Rr	C + Z = 1	BRSH*	Unsigned
Rd ≥ Rr	C = 0	BRSH/ BRCC	Rd < Rr	C = 1	BRLO/BRCS	Unsigned
Rd = Rr	Z = 1	BREQ	Rd ≠ Rr	Z = 0	BRNE	Unsigned
Rd ≤ Rr	C + Z = 1	BRSH(1)	Rd > Rr	C + Z = 0	BRLO*	Unsigned
Rd < Rr	C = 1	BRLO/BRCS	Rd ≥ Rr	C = 0	BRSH/BRCC	Unsigned
Carry	C = 1	BRCS	No carry	C = 0	BRCC	Simple
Negative	N = 1	BRMI	Positive	N = 0	BRPL	Simple
Overflow	V = 1	BRVS	No overflow	V = 0	BRVC	Simple
Zero	Z = 1	BREQ	Not zero	Z = 0	BRNE	Simple

Note: Interchange Rd and Rr in the operation before the test, i.e., CP Rd,Rr → CP Rr,Rd.

Table 4-2. Arithmetic and Logic Instructions

Mnemonic	Operands	Description		Op		Flags	#Clocks AVR	#Clocks AVRxm	#Clocks AVRxt	#Clocks AVRrc
ADD	Rd, Rr	Add without Carry	Rd	←	Rd + Rr	Z,C,N,V,S,H	1	1	1	1
ADC	Rd, Rr	Add with Carry	Rd	←	Rd + Rr + C	Z,C,N,V,S,H	1	1	1	1
ADIW	Rd, K	Add Immediate to Word	Rd	←	Rd + 1:Rd + K	Z,C,N,V,S	2	2	2	N/A
SUB	Rd, Rr	Subtract without Carry	Rd	←	Rd - Rr	Z,C,N,V,S,H	1	1	1	1
SUBI	Rd, K	Subtract Immediate	Rd	←	Rd - K	Z,C,N,V,S,H	1	1	1	1
SBC	Rd, Rr	Subtract with Carry	Rd	←	Rd - Rr - C	Z,C,N,V,S,H	1	1	1	1
SBCI	Rd, K	Subtract Immediate with Carry	Rd	←	Rd - K - C	Z,C,N,V,S,H	1	1	1	1
SBIW	Rd, K	Subtract Immediate from Word	Rd + 1:Rd	←	Rd + 1:Rd - K	Z,C,N,V,S	2	2	2	N/A
AND	Rd, Rr	Logical AND	Rd	←	Rd • Rr	Z,N,V,S	1	1	1	1

Mnemonic	Operands	Description		Op		Flags	#Clocks AVR	#Clocks AVRxm	#Clocks AVRxt	#Clocks AVRrc
ANDI	Rd, K	Logical AND with Immediate	Rd	←	Rd • K	Z,N,V,S	1	1	1	1
OR	Rd, Rr	Logical OR	Rd	←	Rd v Rr	Z,N,V,S	1	1	1	1
ORI	Rd, K	Logical OR with Immediate	Rd	←	Rd v K	Z,N,V,S	1	1	1	1
EOR	Rd, Rr	Exclusive OR	Rd	←	Rd ⊕ Rr	Z,N,V,S	1	1	1	1
COM	Rd	One's Complement	Rd	←	$FF - Rd	Z,C,N,V,S	1	1	1	1
NEG	Rd	Two's Complement	Rd	←	$00 - Rd	Z,C,N,V,S,H	1	1	1	1
SBR	Rd,K	Set Bit(s) in Register	Rd	←	Rd v K	Z,N,V,S	1	1	1	1
CBR	Rd,K	Clear Bit(s) in Register	Rd	←	Rd • ($FFh - K)	Z,N,V,S	1	1	1	1
INC	Rd	Increment	Rd	←	Rd + 1	Z,N,V,S	1	1	1	1
DEC	Rd	Decrement	Rd	←	Rd - 1	Z,N,V,S	1	1	1	1
TST	Rd	Test for Zero or Minus	Rd	←	Rd • Rd	Z,N,V,S	1	1	1	1
CLR	Rd	Clear Register	Rd	←	Rd ⊕ Rd	Z,N,V,S	1	1	1	1
SER	Rd	Set Register	Rd	←	$FF	None	1	1	1	1
MUL	Rd,Rr	Multiply Unsigned	R1:R0	←	Rd x Rr (UU)	Z,C	2	2	2	N/A
MULS	Rd,Rr	Multiply Signed	R1:R0	←	Rd x Rr (SS)	Z,C	2	2	2	N/A
MULSU	Rd,Rr	Multiply Signed with Unsigned	R1:R0	←	Rd x Rr (SU)	Z,C	2	2	2	N/A

Table 4-4. Data Transfer Instructions

Mnemonic	Operands	Description		Op		Flags	#Clocks AVR	#Clocks AVRxm	#Clocks AVRxt	#Clocks AVRrc
MOV	Rd, Rr	Copy Register	Rd	←	Rr	None	1	1	1	1
MOVW	Rd, Rr	Copy Register Pair	Rd+1:Rd	←	Rr+1:Rr	None	1	1	1	N/A
LDI	Rd, K	Load Immediate	Rd	←	K	None	1	1	1	1
LDS	Rd, k	Load Direct from data space	Rd	←	(k)	None	2[1]	2[1]	3[1]	2
LD	Rd, X	Load Indirect	Rd	←	(X)	None	2[1]	1[1]	2[1]	1 / 2
LD	Rd, X+	Load Indirect and Post-Increment	Rd X	← ←	(X) X + 1	None	2[1]	1[1]	2[1]	2 / 3
LD	Rd, -X	Load Indirect and Pre-Decrement	X Rd	← ←	X - 1 (X)	None	2[1]	2[1]	2[1]	2 / 3
LD	Rd, Y	Load Indirect	Rd	←	(Y)	None	2[1]	1[1]	2[1]	1 / 2

Table 4-3. Branch Instructions

Mnemonic	Operands	Description		Op		Flags	#Clocks AVR	#Clocks AVRxm	#Clocks AVRxt	#Clocks AVRrc
RJMP	k	Relative Jump	PC	←	PC + k + 1	None	2	2	2	2
IJMP		Indirect Jump to (Z)	PC(15:0) PC(21:16)	← ←	Z 0	None	2	2	2	2
EIJMP		Extended Indirect Jump to (Z)	PC(15:0) PC(21:16)	← ←	Z EIND	None	2	2	2	N/A
JMP	k	Jump	PC	←	k	None	3	3	3	N/A
RCALL	k	Relative Call Subroutine	PC	←	PC + k + 1	None	3 / 4(1)	2 / 3(1)	2 / 3	3(1)
ICALL		Indirect Call to (Z)	PC(15:0)	←	Z	None	3 / 4(1)	2 / 3(1)	2 / 3	3(1)
EICALL		Extended Indirect Call to (Z)	PC(15:0) PC(21:16)	← ←	Z EIND	None	4(1)	3(1)	2 / 3	N/A
CALL	k	Call Subroutine	PC	←	k	None	4 / 5(1)	3 / 4(1)	3 / 4	N/A
RET		Subroutine Return	PC	←	STACK	None	4 / 5(1)	4 / 5(1)	4 / 5	6(1)
RETI		Interrupt Return	PC	←	STACK	I	4 / 5(1)	4 / 5(1)	4 / 5	6(1)
CPSE	Rd,Rr	Compare, skip if Equal	if (Rd = Rr) PC	←	PC + 2 or 3	None	1 / 2 / 3	1 / 2 / 3	1 / 2 / 3	1 / 2
CP	Rd,Rr	Compare	Rd - Rr			Z,C,N,V,S,H	1	1	1	1
CPC	Rd,Rr	Compare with Carry	Rd - Rr - C			Z,C,N,V,S,H	1	1	1	1
CPI	Rd,K	Compare with Immediate	Rd - K			Z,C,N,V,S,H	1	1	1	1
SBRC	Rr, b	Skip if Bit in Register Cleared	if (Rr(b) = 0) PC	←	PC + 2 or 3	None	1 / 2 / 3	1 / 2 / 3	1 / 2 / 3	1 / 2
SBRS	Rr, b	Skip if Bit in Register Set	if (Rr(b) = 1) PC	←	PC + 2 or 3	None	1 / 2 / 3	1 / 2 / 3	1 / 2 / 3	1 / 2
SBIC	A, b	Skip if Bit in I/O Register Cleared	if (I/O(A,b) = 0) PC	←	PC + 2 or 3	None	1 / 2 / 3	2 / 3 / 4	1 / 2 / 3	1 / 2
SBIS	A, b	Skip if Bit in I/O Register Set	If (I/O(A,b) =1) PC	←	PC + 2 or 3	None	1 / 2 / 3	2 / 3 / 4	1 / 2 / 3	1 / 2
BRBS	s, k	Branch if Status Flag Set	if (SREG(s) = 1) then PC	←	PC + k + 1	None	1 / 2	1 / 2	1 / 2	1 / 2

BREQ	k	Branch if Equal	if (Z = 1) then PC	←	PC + k + 1	None	1 / 2	1 / 2	1 / 2	1 / 2
BRNE	k	Branch if Not Equal	if (Z = 0) then PC	←	PC + k + 1	None	1 / 2	1 / 2	1 / 2	1 / 2
BRCS	k	Branch if Carry Set	if (C = 1) then PC	←	PC + k + 1	None	1 / 2	1 / 2	1 / 2	1 / 2
BRCC	k	Branch if Carry Cleared	if (C = 0) then PC	←	PC + k + 1	None	1 / 2	1 / 2	1 / 2	1 / 2
BRSH	k	Branch if Same or Higher	if (C = 0) then PC	←	PC + k + 1	None	1 / 2	1 / 2	1 / 2	1 / 2

Mnemonic	Operands	Description		Op		Flags	#Clocks AVR	#Clocks AVRxm	#Clocks AVRxt	#Clocks AVRrc
BRLO	k	Branch if Lower	if (C = 1) then PC	←	PC + k + 1	None	1 / 2	1 / 2	1 / 2	1 / 2
BRMI	k	Branch if Minus	if (N = 1) then PC	←	PC + k + 1	None	1 / 2	1 / 2	1 / 2	1 / 2
BRPL	k	Branch if Plus	if (N = 0) then PC	←	PC + k + 1	None	1 / 2	1 / 2	1 / 2	1 / 2
BRGE	k	Branch if Greater or Equal, Signed	if (N ⊕ V= 0) then PC	←	PC + k + 1	None	1 / 2	1 / 2	1 / 2	1 / 2
BRLT	k	Branch if Less Than, Signed	if (N ⊕ V= 1) then PC	←	PC + k + 1	None	1 / 2	1 / 2	1 / 2	1 / 2
BRHS	k	Branch if Half Carry Flag Set	if (H = 1) then PC	←	PC + k + 1	None	1 / 2	1 / 2	1 / 2	1 / 2
BRHC	k	Branch if Half Carry Flag Cleared	if (H = 0) then PC	←	PC + k + 1	None	1 / 2	1 / 2	1 / 2	1 / 2
BRTS	k	Branch if T Flag Set	if (T = 1) then PC	←	PC + k + 1	None	1 / 2	1 / 2	1 / 2	1 / 2
BRTC	k	Branch if T Flag Cleared	if (T = 0) then PC	←	PC + k + 1	None	1 / 2	1 / 2	1 / 2	1 / 2
BRVS	k	Branch if Overflow Flag is Set	if (V = 1) then PC	←	PC + k + 1	None	1 / 2	1 / 2	1 / 2	1 / 2
BRVC	k	Branch if Overflow Flag is Cleared	if (V = 0) then PC	←	PC + k + 1	None	1 / 2	1 / 2	1 / 2	1 / 2
BRIE	k	Branch if Interrupt Enabled	if (I = 1) then PC	←	PC + k + 1	None	1 / 2	1 / 2	1 / 2	1 / 2
BRID	k	Branch if Interrupt Disabled	if (I = 0) then PC	←	PC + k + 1	None	1 / 2	1 / 2	1 / 2	1 / 2