Appendix A: Summary of Atmel AVR Instruction Set

Mnemonics	Operands	Operation	Affected flags	#Clocks	#Clocks XMega
Arithmetic and Logic Instructions					
ADD	Rd, Rr	Rd ← Rd + Rr	Z,C,N,V,S,H	1	
ADC	Rd, Rr	Rd ← Rd + Rr + C	Z,C,N,V,S,H	1	
ADIW[1]	Rd, K	$R_{d+1}:R_d \leftarrow R_{d+1}:R_d + k$	Z,C,N,V,S	2	
SUB	Rd, Rr	Rd ← Rd – Rr	Z,C,N,V,S,H	1	
SUBI	Rd, K	Rd ← Rd – K	Z,C,N,V,S,H	1	
SBC	Rd, Rr	Rd ← Rd – Rr – C	Z,C,N,V,S,H	1	
SBCI	Rd, K	Rd ← Rd – K – C	Z,C,N,V,S,H	1	
SBIW[1]	Rd, K	$R_{d+1}:R_d \leftarrow R_{d+1}:R_d - K$	Z,C,N,V,S	2	
AND	Rd, Rr	Rd ← Rd & Rr	Z,N,V,S	1	
ANDI	Rd, K	Rd ← Rd & K	Z,N,V,S	1	
OR	Rd, Rr	Rd ← Rd | Rr	Z,N,V,S	1	
ORI	Rd, K	Rd ← Rd | K	Z,N,V,S	1	
EOR	Rd, Rr	Rd ← Rd ⊕ Rr	Z,N,V,S	1	
COM	Rd	Rd ← 0 × FF – Rd	Z,C,N,V,S	1	
NEG	Rd	Rd ← 0 × 00 – Rd	Z,C,N,V,S,H	1	
SBR	Rd, K	Rd ← Rd | K	Z,N,V,S	1	
CBR	Rd, K	Rd ← Rd & (0 × FF – K)	Z,N,V,S	1	
INC	Rd	Rd ← Rd + 1	Z,N,V,S	1	
DEC	Rd	Rd ← Rd – 1	Z,N,V,S	1	
TST	Rd	Rd ← Rd & Rd	Z,N,V,S	1	
CLR	Rd	Rd ← Rd ⊕ Rd	Z,N,V,S	1	
SER	Rd	Rd ← 0 × FF	None	1	
MUL[1]	Rd, Rr	R1:R0 ← Rd × Rr (UU)	Z,C	2	
MULS[1]	Rd, Rr	R1:R0 ← Rd × Rr (SS)	Z,C	2	
MULSU[1]	Rd, Rr	R1:R0 ← Rd × Rr (SU)	Z,C	2	
FMUL[1]	Rd, Rr	R1:R0 ← Rd × Rr << 1 (UU)	Z,C	2	
FMULS[1]	Rd, Rr	R1:R0 ← Rd × Rr << 1 (SS)	Z,C	2	
FMULSU[1]	Rd, Rr	R1:R0 ← Rd × Rr << 1 (SU)	Z,C	2	
DES	K	if (H = 0) then R15:R0 ← Encrypt (R15:R0, K) else R15:R0 Decrypt (R15:R0, K)			1/2
Branch Instructions					
RJMP	K	PC ← PC + k +1	none	2	
IJMP[1]		PC(15:0) ← Z, PC(21:16) ← 0	none	2	
EIJMP[1]		PC(15:0) ← Z, PC(21:16) ← EIND	none	2	
JMP[1]	K	PC ← k	none	3	

Table A.1 ■ AVR Instruction Set Summary

(*Continued*)

Table A.1 ■ (Continued)

Mnemonics	Operands	Operation	Affected flags	#Clocks	#Clocks XMega
RCALL	K	PC ← PC + k + 1	none	3/4[3][5]	2/3[3]
ICALL[1]		PC(15:0) ← Z, PC(21:16) ← 0	none	3/4[3][5]	2/3[3]
EICALL[1]		PC(15:0) ← Z, PC(21:16) ← EIND	none	4[3]	3[3]
CALL[1]	K	PC ← k	none	4/5[3]	3/4[3]
RET		PC ← STACK	none	4/5[3]	
RETI		PC ← STACK	I	4/5[3]	
CPSE	Rd, Rr	if (Rd = Rr) PC ← PC + 2 or 3	none	1/2/3	
CP	Rd, Rr	Rd – Rr	Z,C,N,V,S,H	1	
CPC	Rd, Rr	Rd – Rr – C	Z,C,N,V,S,H	1	
CPI	Rd, K	Rd – K	Z,C,N,V,S,H	1	
SBRC	Rr, b	if (Rr(b) = 0) then PC ← PC + 2 or 3	none	1/2/3	
SBRS	Rr, b	if (Rr(b) = 1) then PC ← PC + 2 or 3	none	1/2/3	
SBIC	A, b	if (I/O(A, b) = 0) then PC ← PC + 2 or 3	none	1/2/3	2/3/4
SBIS	A, b	if (I/O(A, b) = 1) then PC ← PC + 2 or 3	none	1/2/3	2/3/4
BRBS	s, k	if (SREG(s) = 1) then PC ← PC + k + 1	none	1/2	
BRBC	s, k	if (SREG(s) = 0) then PC ← PC + k + 1	none	1/2	
BREQ	K	if (Z = 1) then PC ← PC + k + 1	none	1/2	
BRNE	k	if (Z = 0) then PC ← PC + k + 1	none	1/2	
BRCS	k	if (C = 1) then PC ← PC + k + 1	none	1/2	
BRCC	k	if (C = 0) then PC ← PC + k + 1	none	1/2	
BRSH	k	if (C = 0) then PC ← PC + k + 1	none	1/2	
BRLO	k	if (C = 1) then PC ← PC + k + 1	none	1/2	
BRMI	k	if (N = 1) then PC ← PC + k + 1	none	1/2	
BRPL	k	if (N = 0) then PC ← PC + k + 1	none	1/2	
BRGE	k	if (N ⊕ V = 0) then PC ← PC + k + 1	none	1/2	
BRLT	k	if (N ⊕ V = 1) then PC ← PC + k + 1	none	1/2	
BRHS	k	if (H = 1) then PC ← PC + k+ 1	none	1/2	
BRHC	k	if (H = 0) then PC ← PC + k + 1	none	1/2	
BRTS	k	if (T = 1) then PC ← PC + k + 1	none	1/2	
BRTC	k	if (T = 0) then PC ← PC + k + 1	none	1/2	
BRVS	k	if (V = 1) then PC ← PC + k + 1	none	1/2	
BRVC	k	if (V = 0) then PC ← PC + k + 1	none	1/2	
BRIE	k	if (I = 1) then PC ← PC + k + 1	none	1/2	
BRID	k	if (I = D) then PC ← PC + k + 1	none	1/2	
Data Transfer Instructions					
MOV	Rd, Rr	Rd ← Rd	none	1	
MOVW	Rd, Rr	R_{d+1}:R_d ← R_{r+1}:R_r	none	1	
LDI	Rd, K	Rd ← K	none	1	
LDS[1]	Rd, k	Rd ← (k)	none	1[5]2[3]	2[3][4]
LD[2]	Rd, X	Rd ← (X)	none	1[5]2[3]	1[3][4]
LD[2]	Rd, X+	Rd ← (X), X ← X + 1	none	2[3]	1[3][4]

Table A.1 ■ (Continued)

Mnemonics	Operands	Operation	Affected flags	#Clocks	#Clocks XMega
LD[2]	Rd, –X	X ← X − 1, Rd ← (X)	none	2[3]3[5]	2[3][4]
LD[2]	Rd, Y	Rd ← (Y)	none	1[5]2[3]	1[3][4]
LD[2]	Rd, Y+	Rd ← (Y), Y ← Y + 1	none	2[3]	1[3][4]
LD[2]	Rd, –Y	Y ← Y − 1, Rd ← (Y)	none	2[3]3[5]	2[3][4]
LDD[1]	Rd, Y + q	Rd ← (Y + q)	none	2[3]	2[3][4]
LD[2]	Rd, Z	Rd ← (Z)	none	1[5]2[3]	1[3][4]
LD[2]	Rd, Z+	Rd ← (Z), Z ← Z + 1	none	2[3]	1[3][4]
LD[2]	Rd, –Z	Z ← Z − 1, Rd ← (Z)	none	2[3]3[5]	2[3][4]
LDD[1]	Rd, Z + q	Rd ← (Z + q)	none	2[3]	2[3][4]
STS[1]	k, Rr	(k) ← Rr	none	1[5]2[3]	2[3]
ST[2]	X, Rr	(X) ← Rr	none	1[5]2[3]	1[3]
ST[2]	X+, Rr	(X) ← Rr, X ← X + 1	none	1[5]2[3]	1[3]
ST[2]	–X, Rr	X ← X – 1, (X) ← Rr	none	2[3]	2[3]
ST[2]	Y, Rr	(Y) ← Rr	none	1[5]2[3]	1[3]
ST[2]	Y+ , Rr	(Y) ← Rr, (Y) ← Y + 1	none	1[5]2[3]	1[3]
ST[2]	–Y, Rr	Y ← Y – 1, (Y) ← Rr	none	2[3]	2[3]
STD[1]	Y + q, Rr	(Y + q) ← Rr	none	2[3]	2[3]
ST[2]	Z, Rr	(Z) ← Rr	none	1[5]2[3]	1[3]
ST[2]	Z+, Rr	(Z) ← Rr, Z ← Z + 1	none	1[5]2[3]	1[3]
ST[2]	–Z, Rr	Z ← Z – 1, (Z) ← Rr	none	2[3]	2[3]
STD[1]	Z + q, Rr	(Z + q) ← Rr	none	2[3]	2[3]
LPM[1][2]		R0 ← (Z); load program memory	none	3	3
LPM[1][2]	Rd, Z	Rd ← (Z); load program memory	none	3	3
LPM[1][2]	Rd, Z+	Rd ← (Z), Z ← Z + 1; load program memory	none	3	3
ELPM[1]		R0 ← (RAMPZ:Z); load program memory	none	3	
ELPM[1]	Rd, Z	Rd ← (RAMPZ:Z); load program memory	none	3	
ELPM[1]	Rd, Z+	Rd ← (RAMPZ:Z), Z ← Z + 1	none	3	
SPM[1]		(RAMPZ:Z) ← R1:R0; store program mem.	none	—	—
SPM[1]	Z+	(RAMPZ:Z) ← R1:R0, Z ← Z + 2	none	—	—
IN	Rd, A	Rd ← I/O(A)	none	1	
OUT	A, Rr	I/O(A) ← Rr	none	1	
PUSH[1]	Rr	STACK ← Rr	none	2	1[3]
POP[1]	Rd	Rd ← STACK	none	2	2[3]
XCH	Z, Rd	(Z) ← Rd, Rd ← (Z)	none	1	
LAS	Z, Rd	(Z) ← Rd | (Z), Rd ← (Z)	none	1	
LAC	Z, Rd	(Z) ← (0 × FF − Rd) & (Z), Rd ← (Z)	none	1	
LAT	Z, Rd	(Z) ← Rd ⊕ (Z), Rd ← (Z)	none	1	
Bit and Bit-test Instructions					
LSL	Rd	Rd(n + 1) ← Rd(n), Rd(0) ← 0, C ← Rd(7)	Z,C,N,V,H	1	
LSR	Rd	Rd(n) ← Rd(n + 1), Rd(7) ← 0, C ← Rd(0)	Z,C,N,V	1	
ROL	Rd	Rd(0) ← C, Rd(n + 1) ← Rd(n), C ← Rd(7)	Z,C,N,V,H	1	

(*Continued*)

Table A.1 ■ (Continued)

Mnemonics	Operands	Operation	Affected flags	#Clocks	#Clocks XMega
ROR	Rd	Rd(7) ← C, Rd(n) ← Rd(n + 1), C ← Rd(0)	Z,C,N,V	1	
ASR	Rd	Rd(n) ← Rd(n + 1), n = 0, …, 6, Rd(7) ← Rd(7)	Z,C,N,V	1	
SWAP	Rd	Rd(3..0) ↔ Rd(7..4)	none	1	
BSET	s	SREG(s) ← 1	SREG(s)	1	
BCLR	s	SREG(s) ← 0	SREG(s)	1	
SBI	A, b	I/O(A,b) ← 1	none	1[5]2	1
CBI	A, b	I/O(A,b) ← 0	none	1[5]2	1
BST	Rr, b	T ← Rr(b)	T	1	
BLD	Rd, b	Rd(b) ← T	none	1	
SEC		C ← 1	C	1	
CLC		C ← 0	C	1	
SEN		N ← 1	N	1	
CLN		N ← 0	N	1	
SEZ		Z ← 1	Z	1	
CLZ		Z ← 0	Z	1	
SEI		I ← 1	I	1	
CLI		I ← 0	I	1	
SES		S ← 1	S	1	
CLS		S ← 0	S	1	
SEV		V ← 1	V	1	
CLV		V ← 0	V	1	
SET		T ← 1	T	1	
CLT		T ← 0	T	1	
SEH		H ← 1	H	1	
CLH		H ← 0	H	1	
MCU Control Instructions					
BREAK		See specific description for BREAK	none	1	
NOP			none	1	
SLEEP		See specific description about SLEEP	none	1	
WDR		See specific description about WDR	none	1	

Note: 1. This instruction is not available in all devices. Refer to the device specific instruction set summary.
2. Not all variants of this instruction are available in all devices. Refer to the device specific set summary.
3. Cycle times for data memory accesses assume internal memory accesses, and are not valid for accesses via the external RAM interface.
4. One extra cycle must be added when accessing internal SRAM.
5. Number of clock cycles for reduced core tinyAVR.

Source: Atmel Corporation, The Atmel AVR Microcontroller