

ATmega128 instruction set

1. Instruction Set

1.1 Instruction Set Nomenclature

Assembler is case insensitive.

Status Register (SREG) bits

- C: Carry Flag
- Z: Zero Flag
- N: Negative Flag
- V: Two's complement overflow indicator
- S: N xor V, For signed tests
- H: Half Carry Flag
- T: Transfer bit used by BLD and BST instructions
- I: Global Interrupt Enable/Disable Flag

Registers and Operands

- Rd: Destination (and source) register, R0-R31 or R16-R31 (depending on instruction)
- Rr: Source register in the Register File; R0-R31
- P: I/O register, constant (0-31/63), can be a constant expression
- R: Result after instruction is executed
- K: Constant data, (0-255), can be a constant expression
- k: Constant address, value range depending on instruction, can be a constant expression.
- b: Bit in the Register (3-bit), constant (0-7), can be a constant expression
- s: Bit in the Status Register (3-bit), constant (0-7), can be a constant expression
- X,Y,Z: Indirect Address Register, (X=R27:R26, Y=R29:R28 and Z=R31:R30)
- q: Displacement for direct addressing (6-bit)

1.2 ARITHMETIC AND LOGIC INSTRUCTIONS

Mnemo.	Operands	Description	Operation	Flags	#Clk
ADD	Rd, Rr	Add without Carry	$Rd \leftarrow Rd + Rr$	Z,C,N,V,H	1
ADC	Rd, Rr	Add with Carry	$Rd \leftarrow Rd + Rr + C$	Z,C,N,V,H	1
ADIW	Rd, K	Add Immediate to Word	$Rd+1:Rd \leftarrow Rd+1:Rd + K$	Z,C,N,V	2
SUB	Rd, Rr	Subtract without Carry	$Rd \leftarrow Rd - Rr$	Z,C,N,V,H	1
SUBI	Rd, K	Subtract Immediate	$Rd \leftarrow Rd - K$	Z,C,N,V,H	1
SBC	Rd, Rr	Subtract with Carry	$Rd \leftarrow Rd - Rr - C$	Z,C,N,V,H	1
SBCI	Rd, K	Subtract Immediate with Carry	$Rd \leftarrow Rd - K - C$	Z,C,N,V,H	1
SBIW	Rd, K	Subtract Immediate from Word	$Rd+1:Rd \leftarrow Rd+1:Rd - K$	Z,C,N,V	2
AND	Rd, Rr	Logical AND	$Rd \leftarrow Rd \bullet Rr$	Z,N,V	1
ANDI	Rd, K	Logical AND with Immediate	$Rd \leftarrow Rd \bullet K$	Z,N,V	1
OR	Rd, Rr	Logical OR	$Rd \leftarrow Rd \vee Rr$	Z,N,V	1
ORI	Rd, K	Logical OR with Immediate	$Rd \leftarrow Rd \vee K$	Z,N,V	1
EOR	Rd, Rr	Exclusive OR	$Rd \leftarrow Rd \oplus Rr$	Z,N,V	1
COM	Rd	One's Complement	$Rd \leftarrow \$FF - Rd$	Z,C,N,V	1
NEG	Rd	Two's Complement	$Rd \leftarrow \$00 - Rd$	Z,C,N,V,H	1
SBR	Rd,K	Set Bit(s) in Register	$Rd \leftarrow Rd \vee K$	Z,N,V	1
CBR	Rd,K	Clear Bit(s) in Register	$Rd \leftarrow Rd \bullet (\$FFh - K)$	Z,N,V	1
INC	Rd	Increment	$Rd \leftarrow Rd + 1$	Z,N,V	1
DEC	Rd	Decrement	$Rd \leftarrow Rd - 1$	Z,N,V	1
TST	Rd	Test for Zero or Minus	$Rd \leftarrow Rd \bullet Rd$	Z,N,V	1
CLR	Rd	Clear Register	$Rd \leftarrow Rd \oplus Rd$	Z,N,V	1
SER	Rd	Set Register	$Rd \leftarrow \$FF$	None	1
MUL (1)	Rd,Rr	Multiply Unsigned	$R1, R0 \leftarrow Rd \times Rr$	C	2

Note (1). Not available in base-line microcontrollers

1.3 BRANCH INSTRUCTIONS

Mnemo.	Operands	Description	Operation	Flags	#Clk
RJMP	k	Relative Jump	$PC \leftarrow PC + k + 1$	None	2
IJMP		Indirect Jump to (Z)	$PC \leftarrow Z$	None	2
JMP	k	Jump	$PC \leftarrow k$	None	3
RCALL	k	Relative Call Subroutine	$PC \leftarrow PC + k + 1$	None	3
ICALL		Indirect Call to (Z)	$PC \leftarrow Z$	None	3
CALL	k	Call Subroutine	$PC \leftarrow k$	None	4
RET		Subroutine Return	$PC \leftarrow STACK$	None	4
RETI		Interrupt Return	$PC \leftarrow STACK$	I	4
CPSE	Rd,Rr	Compare, Skip if Equal	if $(Rd = Rr)$ $PC \leftarrow PC + 2$ or 3	None	1/2/3
CP	Rd,Rr	Compare	$Rd - Rr$	Z,C,N,V,H	1
CPC	Rd,Rr	Compare with Carry	$Rd - Rr - C$	Z,C,N,V,H	1
CPI	Rd,K	Compare with Immediate	$Rd - K$	Z,C,N,V,H	1
SBRC	Rr, b	Skip if Bit in Reg. Cleared	if $(Rr(b)=0)$ $PC \leftarrow PC + 2$ or 3	None	1/2/3
SBRs	Rr, b	Skip if Bit in Register Set	if $(Rr(b)=1)$ $PC \leftarrow PC + 2$ or 3	None	1/2/3
SBIC	P, b	Skip if Bit in I/O Reg. Cleared	if $(I/O(P,b)=0)$ $PC \leftarrow PC + 2$ or 3	None	1/2/3
SBIS	P, b	Skip if Bit in I/O Register Set	if $(I/O(P,b)=1)$ $PC \leftarrow PC + 2$ or 3	None	1/2/3
BRBS	s, k	Branch if Status Flag Set	if $(SREG(s) = 1)$ then $PC \leftarrow PC+k + 1$	None	1/2
BRBC	s, k	Branch if Status Flag Cleared	if $(SREG(s) = 0)$ then $PC \leftarrow PC+k + 1$	None	1/2

BRANCH INSTRUCTIONS (Continued)

Mnemo.	Operands	Description	Operation	Flags	#Clk
BREQ	k	Branch if Equal	if ($Z = 1$) then $PC \leftarrow PC + k + 1$	None	1/2
BRNE	k	Branch if Not Equal	if ($Z = 0$) then $PC \leftarrow PC + k + 1$	None	1/2
BRCS	k	Branch if Carry Set	if ($C = 1$) then $PC \leftarrow PC + k + 1$	None	1/2
BRCC	k	Branch if Carry Cleared	if ($C = 0$) then $PC \leftarrow PC + k + 1$	None	1/2
BRSH	k	Branch if Same or Higher	if ($C = 0$) then $PC \leftarrow PC + k + 1$	None	1/2
BRLO	k	Branch if Lower	if ($C = 1$) then $PC \leftarrow PC + k + 1$	None	1/2
BRMI	k	Branch if Minus	if ($N = 1$) then $PC \leftarrow PC + k + 1$	None	1/2
BRPL	k	Branch if Plus	if ($N = 0$) then $PC \leftarrow PC + k + 1$	None	1/2
BRGE	k	Branch if Greater or Equal, Signed	if ($N \oplus V = 0$) then $PC \leftarrow PC + k + 1$	None	1/2
BRLT	k	Branch if Less Than, Signed	if ($N \oplus V = 1$) then $PC \leftarrow PC + k + 1$	None	1/2
BRHS	k	Branch if Half Carry Flag Set	if ($H = 1$) then $PC \leftarrow PC + k + 1$	None	1/2
BRHC	k	Branch if Half Carry Flag Cleared	if ($H = 0$) then $PC \leftarrow PC + k + 1$	None	1/2
BRTS	k	Branch if T Flag Set	if ($T = 1$) then $PC \leftarrow PC + k + 1$	None	1/2
BRTC	k	Branch if T Flag Cleared	if ($T = 0$) then $PC \leftarrow PC + k + 1$	None	1/2
BRVS	k	Branch if Overflow Flag is Set	if ($V = 1$) then $PC \leftarrow PC + k + 1$	None	1/2
BRVC	k	Branch if Overflow Fl. Cleared	if ($V = 0$) then $PC \leftarrow PC + k + 1$	None	1/2
BRIE	k	Branch if Interrupt Enabled	if ($I = 1$) then $PC \leftarrow PC + k + 1$	None	1/2
BRID	k	Branch if Interrupt Disabled	if ($I = 0$) then $PC \leftarrow PC + k + 1$	None	1/2

1.4 DATA TRANSFER INSTRUCTIONS

Mnemo.	Operands	Description	Operation	Flags	#Clk
MOV	Rd, Rr	Copy Register	$Rd \leftarrow Rr$	None	1
LDI	Rd, K	Load Immediate	$Rd \leftarrow K$	None	1
LDS	Rd, k	Load Direct from SRAM	$Rd \leftarrow (k)$	None	3
LD	Rd, X	Load Indirect	$Rd \leftarrow (X)$	None	2
LD	Rd, X+	Load Indirect and Post-Increment	$Rd \leftarrow (X), X \leftarrow X + 1$	None	2
LD	Rd, -X	Load Indirect and Pre-Decrement	$X \leftarrow X - 1, Rd \leftarrow (X)$	None	2
LD	Rd, Y	Load Indirect	$Rd \leftarrow (Y)$	None	2
LD	Rd, Y+	Load Indirect and Post-Increment	$Rd \leftarrow (Y), Y \leftarrow Y + 1$	None	2
LD	Rd, -Y	Load Indirect and Pre-Decrement	$Y \leftarrow Y - 1, Rd \leftarrow (Y)$	None	2
LDD	Rd, Y+q	Load Indirect with Displacement	$Rd \leftarrow (Y + q)$	None	2
LD	Rd, Z	Load Indirect	$Rd \leftarrow (Z)$	None	2
LD	Rd, Z+	Load Indirect and Post-Increment	$Rd \leftarrow (Z), Z \leftarrow Z + 1$	None	2
LD	Rd, -Z	Load Indirect and Pre-Decrement	$Z \leftarrow Z - 1, Rd \leftarrow (Z)$	None	2
LDD	Rd, Z+q	Load Indirect with Displacement	$Rd \leftarrow (Z + q)$	None	2
STS	k, Rr	Store Direct to SRAM	$(k) \leftarrow Rr$	None	3
ST	X, Rr	Store Indirect	$(X) \leftarrow Rr$	None	2
ST	X+, Rr	Store Indirect and Post-Increment	$(X) \leftarrow Rr, X \leftarrow X + 1$	None	2
ST	-X, Rr	Store Indirect and Pre-Decrement	$X \leftarrow X - 1, (X) \leftarrow Rr$	None	2
ST	Y, Rr	Store Indirect	$(Y) \leftarrow Rr$	None	2
ST	Y+, Rr	Store Indirect and Post-Increment	$(Y) \leftarrow Rr, Y \leftarrow Y + 1$	None	2
ST	-Y, Rr	Store Indirect and Pre-Decrement	$Y \leftarrow Y - 1, (Y) \leftarrow Rr$	None	2
STD	Y+q, Rr	Store Indirect with Displacement	$(Y + q) \leftarrow Rr$	None	2
ST	Z, Rr	Store Indirect	$(Z) \leftarrow Rr$	None	2
ST	Z+, Rr	Store Indirect and Post-Increment	$(Z) \leftarrow Rr, Z \leftarrow Z + 1$	None	2

DATA TRANSFER INSTRUCTIONS (Continued)

Mnemo.	Operands	Description	Operation	Flags	#Clk
ST	-Z, Rr	Store Indirect and Pre-Decrement	$Z \leftarrow Z - 1, (Z) \leftarrow Rr$	None	2
STD	Z+q,Rr	Store Indirect with Displacement	$(Z + q) \leftarrow Rr$	None	2
LPM		Load Program Memory	$R0 \leftarrow (Z)$	None	3
IN	Rd, P	In Port	$Rd \leftarrow P$	None	1
OUT	P, Rr	Out Port	$P \leftarrow Rr$	None	1
PUSH	Rr	Push Register on Stack	$STACK \leftarrow Rr$	None	2
POP	Rd	Pop Register from Stack	$Rd \leftarrow STACK$	None	2

1.5 BIT AND BIT-TEST INSTRUCTIONS

Mnemo.	Operands	Description	Operation	Flags	#Clk
LSL	Rd	Logical Shift Left	$Rd(n+1) \leftarrow Rd(n),$ $Rd(0) \leftarrow 0, C \leftarrow Rd(7)$	Z,C,N,V,H	1
LSR	Rd	Logical Shift Right	$Rd(n) \leftarrow Rd(n+1),$ $Rd(7) \leftarrow 0, C \leftarrow Rd(0)$	Z,C,N,V	1
ROL	Rd	Rotate Left Through Carry	$Rd(0) \leftarrow C, Rd(n+1) \leftarrow Rd(n), C \leftarrow Rd(7)$	Z,C,N,V,H	1
ROR	Rd	Rotate Right Through Carry	$Rd(7) \leftarrow C, Rd(n) \leftarrow Rd(n+1), C \leftarrow Rd(0)$	Z,C,N,V	1
ASR	Rd	Arithmetic Shift Right	$Rd(n) \leftarrow Rd(n+1), n=0..6$	Z,C,N,V	1
SWAP		Rd Swap Nibbles	$Rd(3..0) \leftrightarrow Rd(7..4)$	None	1
BSET	s	Flag Set	$SREG(s) \leftarrow 1$	SREG(s)	1
BCLR	s	Flag Clear	$SREG(s) \leftarrow 0$	SREG(s)	1
SBI	P, b	Set Bit in I/O Register	$I/O(P, b) \leftarrow 1$	None	2
CBI	P, b	Clear Bit in I/O Register	$I/O(P, b) \leftarrow 0$	None	2
BST	Rr, b	Bit Store from Register to T	$T \leftarrow Rr(b)$	T	1
BLD	Rd, b	Bit load from T to Register	$Rd(b) \leftarrow T$	None	1
SEC		Set Carry	$C \leftarrow 1$	C	1
CLC		Clear Carry	$C \leftarrow 0$	C	1
SEN		Set Negative Flag	$N \leftarrow 1$	N	1
CLN		Clear Negative Flag	$N \leftarrow 0$	N	1
SEZ		Set Zero Flag	$Z \leftarrow 1$	Z	1
CLZ		Clear Zero Flag	$Z \leftarrow 0$	Z	1
SEI		Global Interrupt Enable	$I \leftarrow 1$	I	1
CLI		Global Interrupt Disable	$I \leftarrow 0$	I	1
SES		Set Signed Test Flag	$S \leftarrow 1$	S	1
CLS		Clear Signed Test Flag	$S \leftarrow 0$	S	1
SEV		Set Two's Complement Overflow	$V \leftarrow 1$	V	1
CLV		Clear Two's Complement Overflow	$V \leftarrow 0$	V	1
SET		Set T in SREG	$T \leftarrow 1$	T	1
CLT		Clear T in SREG	$T \leftarrow 0$	T	1
SEH		Set Half Carry Flag in SREG	$H \leftarrow 1$	H	1
CLH		Clear Half Carry Flag in SREG	$H \leftarrow 0$	H	1
NOP		No Operation		None	1
SLEEP		Sleep		None	1
WDR		Watchdog Reset		None	1