ATMEL INFO PAGE

Mnemonics	Operands	Description
ARITHMETIC AND LOGIC INSTRUCTIONS		
ADD	Rd, Rr	Add two Registers
ADC	Rd, Rr	Add with Carry two Registers
ADIW	Rdl,K	Add Immediate to Word
SUB	Rd, Rr	Subtract two Registers
SUBI	Rd, K	Subtract Constant from Register
SBC	Rd, Rr	Subtract with Carry two Registers
SBCI	Rd, K	Subtract with Carry Constant from Reg.
SBIW	Rdl,K	Subtract Immediate from Word
AND	Rd, Rr	Logical AND Registers
ANDI	Rd, K	Logical AND Register and Constant
OR	Rd, Rr	Logical OR Registers
ORI	Rd, K	Logical OR Register and Constant
EOR	Rd, Rr	Exclusive OR Registers
COM	Rd	One's Complement
NEG	Rd	Two's Complement
SBR	Rd,K	Set Bit(s) in Register
CBR	Rd,K	Clear Bit(s) in Register
INC	Rd	Increment
DEC	Rd	Decrement
TST	Rd	Test for Zero or Minus
CLR	Rd	Clear Register
SER	Rd	Set Register
MUL	Rd, Rr	Multiply Unsigned
MULS	Rd, Rr	Multiply Signed
MULSU	Rd, Rr	Multiply Signed with Unsigned
FMUL	Rd, Rr	Fractional Multiply Unsigned
FMULS	Rd, Rr	Fractional Multiply Signed
FMULSU	Rd, Rr	Fractional Multiply Signed with Unsigned

BIT AND BIT-TEST INSTRUCTIONS		
SBI	P,b	Set Bit in I/O Register
CBI	P,b	Clear Bit in I/O Register
LSL	Rd	Logical Shift Left
LSR	Rd	Logical Shift Right
ROL	Rd	Rotate Left Through Carry
ROR	Rd	Rotate Right Through Carry
ASR	Rd	Arithmetic Shift Right
SWAP	Rd	Swap Nibbles
BSET	s	Flag Set
BCLR	s	Flag Clear
BST	Rr, b	Bit Store from Register to T
BLD	Rd, b	Bit load from T to Register
SEC		Set Carry
CLC		Clear Carry
SEN		Set Negative Flag
CLN		Clear Negative Flag
SEZ		Set Zero Flag
CLZ		Clear Zero Flag
SEI		Global Interrupt Enable
CLI		Global Interrupt Disable
SES		Set Signed Test Flag
CLS		Clear Signed Test Flag
SEV		Set Twos Complement Overflow.
CLV		Clear Twos Complement Overflow
SET		Set T in SREG
CLT		Clear T in SREG
SEH		Set Half Carry Flag in SREG
CLH		Clear Half Carry Flag in SREG

DATA TRANSFER INSTRUCTIONS		
MOV	Rd, Rr	Move Between Registers
MOVW	Rd, Rr	Copy Register Word
LDI	Rd, K	Load Immediate
LD	Rd, X	Load Indirect
LD	Rd, X+	Load Indirect and Post-Inc.
LD	Rd, - X	Load Indirect and Pre-Dec.
LD	Rd, Y	Load Indirect
LD	Rd, Y+	Load Indirect and Post-Inc.
LD	Rd, - Y	Load Indirect and Pre-Dec.
LDD	Rd,Y+q	Load Indirect with Displacement
LD	Rd, Z	Load Indirect
LD	Rd, Z+	Load Indirect and Post-Inc.
LD	Rd, -Z	Load Indirect and Pre-Dec.
LDD	Rd, Z+q	Load Indirect with Displacement
LDS	Rd, k	Load Direct from SRAM
ST	X, Rr	Store Indirect
ST	X+, Rr	Store Indirect and Post-Inc.
ST	- X, Rr	Store Indirect and Pre-Dec.
ST	Y, Rr	Store Indirect
ST	Y+, Rr	Store Indirect and Post-Inc.
ST	- Y, Rr	Store Indirect and Pre-Dec.
STD	Y+q,Rr	Store Indirect with Displacement
ST	Z, Rr	Store Indirect
ST	Z+, Rr	Store Indirect and Post-Inc.
ST	-Z, Rr	Store Indirect and Pre-Dec.
STD	Z+q,Rr	Store Indirect with Displacement
STS	k, Rr	Store Direct to SRAM
LPM		Load Program Memory
LPM	Rd, Z	Load Program Memory
LPM	Rd, Z+	Load Program Memory and Post-Inc
SPM		Store Program Memory
IN	Rd, P	In Port
OUT	P, Rr	Out Port
PUSH	Rr	Push Register on Stack
POP	Rd	Pop Register from Stack

BRANCH INSTRUCTIONS		
RJMP	k	Relative Jump
IJMP		Indirect Jump to (Z)
JMP[1]	k	Direct Jump
RCALL	k	Relative Subroutine Call
ICALL		Indirect Call to (Z)
CALL[1]	k	Direct Subroutine Call
RET		Subroutine Return
RETI		Interrupt Return
CPSE	Rd,Rr	Compare, Skip if Equal
CP	Rd,Rr	Compare
CPC	Rd,Rr	Compare with Carry
CPI	Rd,K	Compare Register with Immediate
SBRC	Rr, b	Skip if Bit in Register Cleared
SBRS	Rr, b	Skip if Bit in Register is Set
SBIC	P, b	Skip if Bit in I/O Register Cleared
SBIS	P, b	Skip if Bit in I/O Register is Set
BRBS	s, k	Branch if Status Flag Set
BRBC	s, k	Branch if Status Flag Cleared
BREQ	k	Branch if Equal
BRNE	k	Branch if Not Equal
BRCS	k	Branch if Carry Set
BRCC	k	Branch if Carry Cleared
BRSH	k	Branch if Same or Higher
BRLO	k	Branch if Lower
BRMI	k	Branch if Minus
BRPL	k	Branch if Plus
BRGE	k	Branch if Greater or Equal, Signed
BRLT	k	Branch if Less Than Zero, Signed