Arithmetic and Logic Instructions

Mnem.	*Ops.*	*Range*	*Description*	*Operation*	*Flags*	*Cycle*
ADD	Rd,Rr	d,r ∈ [0,31]	Add without Carry	Rd ← Rd + Rr	Z,C,N,V,H,S	1
ADC	Rd,Rr	d,r ∈ [0,31]	Add with Carry	Rd ← Rd + Rr + C	Z,C,N,V,H,S	1
ADIW	Rd, K	d ∈ {24,26,28,30}, K ∈ [0,63]	Add Immediate To Word	Rd+1:Rd←Rd+1:Rd+K	Z,C,N,V,S	2
SUB	Rd,Rr	d,r ∈ [0,31]	Subtract without Carry	Rd ← Rd - Rr	Z,C,N,V,H,S	1
SUBI	Rd,K	d ∈ [16,31], K ∈ [0,255]	Subtract Immediate	Rd ← Rd - K	Z,C,N,V,H,S	1
SBC	Rd,Rr	d,r ∈ [0,31]	Subtract with Carry	Rd ← Rd - Rr - C	Z,C,N,V,H,S	1
SBCI	Rd,K	d ∈ [16,31], K ∈ [0,255]	Subtract with Carry Immedtiate	Rd ← Rd - K - C	Z,C,N,V,H,S	1
AND	Rd,Rr	d,r ∈ [0,31]	Logical AND	Rd ← Rd ∧ Rr	Z,N,V,S	1
ANDI	Rd,K	d ∈ [16,31], K ∈ [0,255]	Logical AND with Immediate	Rd ← Rd ∧ K	Z,N,V,S	1
OR	Rd,Rr	d,r ∈ [0,31]	Logical OR	Rd ← Rd ∨ Rr	Z,N,V,S	1
ORI	Rd,K	d ∈ [16,31], K ∈ [0,255]	Logical OR with Immediate	Rd ← Rd ∨ K	Z,N,V,S	1
EOR	Rd,Rr	d,r ∈ [0,31]	Logical Exclusive OR	Rd ← Rd ⊕ Rr	Z,N,V,S	1
COM	Rd	d ∈ [0,31]	One's Complement	Rd ← $FF - Rd	Z,C,N,V,S	1
NEG	Rd	d,r ∈ [0,31]	Two's Complement	Rd ← $00 - Rd	Z,C,N,V,H,S	1
SBR	Rd,K	d ∈ [16,31], K ∈ [0,255]	Set Bit(s) in Register	Rd ← Rd ∨ K	Z,C,N,V,S	1
CBR	Rd,K	d ∈ [16,31], K ∈ [0,255]	Clear Bit(s) in Register	Rd ← Rd ∧ ($FF - K)	Z,C,N,V,S	1
INC	Rd	d ∈ [0,31]	Increment Register	Rd ← Rd + 1	Z,N,V,S	1
DEC	Rd	d ∈ [0,31]	Decrement Register	Rd ← Rd - 1	Z,N,V,S	1
TST	Rd	d,r ∈ [0,31]	Test for Zero or Negative	Rd ← Rd ∧ Rd	Z,C,N,V,S	1
CLR	Rd	d ∈ [0,31]	Clear Register	Rd ← 0	Z,C,N,V,S	1
SER	Rd	d ∈ [16,31]	Set Register	Rd ← $FF	None	1
SBIW	Rd,K	d ∈ {24,26,28,30}, K ∈ [0,63]	Subtract Immediate from Word	Rd+1:Rd←Rd+1:Rd-K	Z,C,N,V,S	2
MUL	Rd,Rr	d,r ∈ [0,31]	Multiply Unsigned	R1:R0 ← Rd * Rr	Z,C	2
MULS	Rd,Rr	d,r ∈ [0,31]	Multiply Signed	R1:R0 ← Rd * Rr	Z,C	2
MULSU	Rd,Rr	d,r ∈ [0,31]	Multiply Signed with Unsigned	R1:R0 ← Rd * Rr	Z,C	2
FMUL	Rd,Rr	d,r ∈ [16,23]	Fractional Multiply Unsigned	R1:R0 ← (Rd * Rr) << 1	Z,C	2
FMULS	Rd,Rr	d,r ∈ [16,23]	Fractional Multiply Signed	R1:R0 ← (Rd *Rr) << 1	Z,C	2
FMULSU	Rd,Rr	d,r ∈ [16,23]	Fractional Multiply Signed with Unsigned	R1:R0 ← (Rd * Rr) << 1	Z,C	2

Branch Instructions

Mnem.	*Ops.*	*Range*	*Description*	*Operation*	*Flags*	*Cycle*
RJMP	k	k ∈ [-2K,2K]	Relative Jump	PC ← PC + k +1	None	2
IJMP	None		Indirect Jump to (Z)	PC ← Z	None	2
JMP	k	k ∈ [0,4M]	Jump	PC ← k	None	3
RCALL	k	k ∈ [-2K,2K]	Relative Call Subroutine	STACK ← PC+1, PC ← PC + k + 1	None	3/4*
ICALL	None		Indirect Call to (Z)	STACK ← PC+1, PC ← Z	None	3/4*
CALL	k	k ∈ [0,4M]	Call Subroutine	STACK ← PC+2, PC ← k	None	4/5*
RET	None		Subroutine Return	PC ← STACK	None	4/5*
RETI	None		Interrupt Return	PC ← STACK	I	4/5*
CPSE	Rd,Rr	d,r ∈ [0,31]	Compare, Skip if equal	if (Rd ==Rr) PC ← PC 2 or 3	None	1/2/3
CP	Rd,Rr	d,r ∈ [0,31]	Compare	Rd -Rr	Z,C,N,V,H,S	1
CPC	Rd,Rr	d,r ∈ [0,31]	Compare with Carry	Rd - Rr - C	Z,C,N,V,H,S	1
CPI	Rd,K	d ∈ [16,31], K∈ [0,255]	Compare with Immediate	Rd - K	Z,C,N,V,H,S	1
SBRC	Rr,b	r ∈ [0,31], b ∈ [0,7]	Skip if bit in register cleared	if(Rr(b)==0) PC ← PC + 2 or 3	None	1/2/3
SBRS	Rr,b	r ∈ [0,31], b ∈ [0,7]	Skip if bit in register set	if(Rr(b)==1) PC ← PC + 2 or 3	None	1/2/3
SBIC	P,b	P ∈ [0,31], b ∈ [0,7]	Skip if bit in I/O register cleared	if(I/O(P,b)==0) PC ← PC + 2 or 3	None	1/2/3
SBIS	P,b	P ∈ [0,31], b ∈ [0,7]	Skip if bit in I/O register set	if(I/O(P,b)==1) PC ← PC + 2 or 3	None	1/2/3
BRBC	s,k	s ∈ [0,7], k ∈ [-64,63]	Branch if Status flag cleared	if(SREG(s)==0) PC ← PC + k + 1	None	1/2
BRBS	s,k	s ∈ [0,7], k ∈ [-64,63]	Branch if Status flag set	if(SREG(s)==1) PC ← PC + k + 1	None	1/2
BREQ	k	k ∈ [-64,63]	Branch if equal	if(Z==1) PC ← PC + k + 1	None	1/2
BRNE	k	k ∈ [-64,63]	Branch if not equal	if(Z==0) PC ← PC + k + 1	None	1/2
BRCS	k	k ∈ [-64,63]	Branch if carry set	if(C==1) PC ← PC + k + 1	None	1/2
BRCC	k	k ∈ [-64,63]	Branch if carry cleared	if(C==0) PC ← PC + k + 1	None	1/2
BRSH	k	k ∈ [-64,63]	Branch if same or higher	if(C==0) PC ← PC + k + 1	None	1/2
BRLO	k	k ∈ [-64,63]	Branch if lower	if(C==1) PC ← PC + k + 1	None	1/2
BRMI	k	k ∈ [-64,63]	Branch if minus	if(N==1) PC ← PC + k + 1	None	1/2
BRPL	k	k ∈ [-64,63]	Branch if plus	if(N==0) PC ← PC + k + 1	None	1/2
BRGE	k	k ∈ [-64,63]	Branch if greater than or equal (signed)	if(S==0) PC ← PC + k + 1	None	1/2
BRLT	k	k ∈ [-64,63]	Branch if less than (signed)	if(S==1) PC ← PC + k + 1	None	1/2
BRHS	k	k ∈ [-64,63]	Branch if half carry flag set	if(H==1) PC ← PC + k + 1	None	1/2
BRHC	k	k ∈ [-64,63]	Branch if half carry flag cleared	if(H==0) PC ← PC + k + 1	None	1/2
BRTS	k	k ∈ [-64,63]	Branch if T flag set	if(T==1) PC ← PC + k + 1	None	1/2
BRTC	k	k ∈ [-64,63]	Branch if T flag cleared	if(T==0) PC ← PC + k + 1	None	1/2
BRVS	k	k ∈ [-64,63]	Branch if overflow flag set	if(V==1) PC ← PC + k + 1	None	1/2
BRVC	k	k ∈ [-64,63]	Branch if overflow flag cleared	if(V==0) PC ← PC + k + 1	None	1/2
BRIE	k	k ∈ [-64,63]	Branch if interrupt enabled	if(I==1) PC ← PC + k + 1	None	1/2
BRID	k	k ∈ [-64,63]	Branch if interrupt disabled	if(I==0) PC ← PC + k + 1	None	1/2

Data Transfer Instructions

Mnemonic	*Operands*	*Range*	*Description*	*Operation*	*Flags*	*Cycles*
MOV	Rd,Rr	d,r ∈ [0,31]	Copy register	Rd ← Rr	None	1
MOVW	Rd,Rr	d,r ∈ {0,2,...,30}	Copy register pair	Rd+1:Rd ← Rr+1:Rr, r,d even	None	1
LDI	Rd,K	d ∈ [16,31], k ∈ [0,255]	Load Immediate	Rd ← K	None	1
LDS	Rd,k	d ∈ [0,31], k ∈ [0,64K]	Load Direct	Rd ← (k)	None	2*
LD	Rd,X	d ∈ [0,31]	Load Indirect	Rd ← (X)	None	2*
LD	Rd,X+	d ∈ [0,31]	Load Indirect and Post-Increment	Rd ← (X), X←X+1	None	2*
LD	Rd,-X	d ∈ [0,31]	Load Indirect and Pre-Decrement	X←X-1, Rd ← (X)	None	2*
LD	Rd,Y	d ∈ [0,31]	Load Indirect	Rd ← (Y)	None	2*
LD	Rd,Y+	d ∈ [0,31]	Load Indirect and Post-Increment	Rd ← (Y), Y←Y+1	None	2*
LD	Rd,-Y	d ∈ [0,31]	Load Indirect and Pre-Decrement	Y←Y-1, Rd ← (Y)	None	2*
LDD	Rd,Y+q	d ∈ [0,31], q ∈ [0,63]	Load Indirect with displacement	Rd ← (Y+q)	None	2*
LD	Rd,Z	d ∈ [0,31]	Load Indirect	Rd ← (Z)	None	2*
LD	Rd,Z+	d ∈ [0,31]	Load Indirect and Post-Increment	Rd ← (Z), Z←Z+1	None	2*
LD	Rd,-Z	d ∈ [0,31]	Load Indirect and Pre-Decrement	Z←Z-1, Rd ← (Z)	None	2*
LDD	Rd,Z+q	d ∈ [0,31], q ∈ [0,63]	Load Indirect with displacement	Rd ← (Z+q)	None	2*
STS	k,Rr	r ∈ [0,31], k ∈ [0,64K]	Store Direct	(k) ← Rr	None	2*
ST	X,Rr	r ∈ [0,31]	Store Indirect	(X) ← Rr	None	2*
ST	X+,Rr	r ∈ [0,31]	Store Indirect and Post-Increment	(X) ← Rr, X←X+1	None	2*
ST	-X,Rr	r ∈ [0,31]	Store Indirect and Pre-Decrement	X←X-1, (X)←Rr	None	2*
ST	Y,Rr	r ∈ [0,31]	Store Indirect	(Y) ← Rr	None	2*
STD	Y+,Rr	r ∈ [0,31]	Store Indirect and Post-Increment	(Y) ← Rr, Y←Y+1	None	2
ST	-Y,Rr	r ∈ [0,31]	Store Indirect and Pre-Decrement	Y←Y-1, (Y) ← Rr	None	2
ST	Y+q,Rr	r ∈ [0,31], q ∈ [0,63]	Store Indirect with displacement	(Y+q) ← Rr	None	2
ST	Z,Rr	r ∈ [0,31]	Store Indirect	(Z) ← Rr	None	2
ST	Z+,Rr	r ∈ [0,31]	Store Indirect and Post-Increment	(Z) ← Rr, Z←Z+1	None	2
ST	-Z,Rr	r ∈ [0,31]	Store Indirect and Pre-Decrement	Z←Z-1, (Z) ← Rr	None	2
STD	Z+q,Rr	r ∈ [0,31], q ∈ [0,63]	Store Indirect with displacement	(Z+q) ← Rr	None	2
LPM	None		Load Program Memory	R0 ← (Z)	None	3
LPM	Rd,Z	d ∈ [0,31]	Load Program Memory	Rd ← (Z)	None	3
LPM	Rd,Z+	d ∈ [0,31]	Load Program Memory and Post-Increment	Rd ← (Z), Z←Z+1	None	3
SPM	None		Store Program Memory	(Z) ← R1:R0	None	-
IN	Rd,P	d ∈ [0,31], P ∈ [0,63]	In Port	Rd ← P	None	1
OUT	P,Rr	r ∈ [0,31], P ∈ [0,63]	Out Port	P ← Rr	None	1
PUSH	Rr	r ∈ [0,31]	Push register on Stack	STACK ← Rr	None	2
POP	Rd	d ∈ [0,31]	Pop register from Stack	Rd ← STACK	None	2

Bit and Bit-test Instructions

Mnem.	*Ops.*	*Range*	*Description*	*Operation*	*Flags*	*Cycles*
LSL	Rd	d ∈ [0,31]	Logical shift left	Rd(n+1)←Rd(n), Rd(0)←0, C←Rd(7)	Z,C,N,V,H,S	1
LSR	Rd	d ∈ [0,31]	Logical shift right	Rd(n)←Rd(n+1), Rd(7)←0, C←Rd(0)	Z,C,N,V,S	1
ROL	Rd	d ∈ [0,31]	Rotate left through carry	Rd(0)←C, Rd(n+1)←Rd(n), C←Rd(7)	Z,C,N,V,H,S	1
ROR	Rd	d ∈ [0,31]	Rotate right through carry	Rd(7)←C, Rd(n)←Rd(n+1), C←Rd(0)	Z,C,N,V,S	1
ASR	Rd	d ∈ [0,31]	Arithmetic shift right	Rd(n)←Rd(n+1), n=0,...,6	Z,C,N,V,S	1
SWAP	Rd	d ∈ [0,31]	Swap nibbles	Rd(3..0) ← Rd(7..4), Rd(7..4) ← Rd(3..0)	None	1
BSET	s	s ∈ [0,7]	Set flag	SREG(s) ← 1	SREG(s)	1
BCLR	s	s ∈ [0,7]	Clear flag	SREG(s) ← 0	SREG(s)	1
SBI	P,b	P ∈ [0,31], b ∈ [0,7]	Set bit in I/O register	I/O(P,b) ← 1	None	2
CBI	P,b	P ∈ [0,31], b ∈ [0,7]	Clear bit in I/O register	I/O(P,b) ← 0	None	2
BST	Rr,b	r ∈ [0,31], b ∈ [0,7]	Bit store from register to T	T ← Rr(b)	T	1
BLD	Rd,b	d ∈ [0,31], b ∈ [0,7]	Bit load from register to T	Rd(b) ← T	None	1
SEC	None		Set carry flag	C ←1	C	1
CLC	None		Clear carry flag	C ← 0	C	1
SEN	None		Set negative flag	N ← 1	N	1
CLN	None		Clear negative flag	N ← 0	N	1
SEZ	None		Set zero flag	Z ← 1	Z	1
CLZ	None		Clear zero flag	Z ← 0	Z	1
SEI	None		Set interrupt flag	I ← 1	I	1
CLI	None		Clear interrupt flag	I ← 0	I	1
SES	None		Set signed flag	S ← 1	S	1
CLN	None		Clear signed flag	S ← 0	S	1
SEV	None		Set overflow flag	V ← 1	V	1
CLV	None		Clear overflow flag	V ← 0	V	1
SET	None		Set T-flag	T ← 1	T	1
CLT	None		Clear T-flag	T ← 0	T	1
SEH	None		Set half carry flag	H ← 1	H	1
CLH	None		Clear half carry flag	H ← 0	H	1
NOP	None		No operation	None	None	1
SLEEP	None		Sleep	See instruction manual	None	1
WDR	None		Watchdog Reset	See instruction manual	None	1

Test (CP Rd,Rr)	Booleana	Mnemonico	Comentario
Rd ≥Rr	(N ⊕ V) = 0	BRGE	Signo
Rd < Rr	(N ⊕ V) = 1	BRLT	Signo
Rd = Rr	Z = 1	BREQ	Signo/Sin signo
Rd ≠ Rr	Z = 0	BRNE	Signo/Sin signo
Rd ≥ Rr	C = 0	BRCC/BRSH	Sin signo
Rd < Rr	C = 1	BRCS/BRLO	Sin signo
Carry	C=1	BRCS	Simple
Sin carry	C=0	BRCC	Simple
Negativo	N=1	BRMI	Simple
Positivo	N=0	BRPL	Simple
Overflow	V=1	BRVS	Simple
Sin overflow	V=0	BRVC	Simple
Cero	Z=1	BREQ	Simple
No cero	Z=0	BRNE	Simple

CS12	CS11	CS10	Descripción
0	0	0	Temporizador parado
0	0	1	Frecuencia clk/1
0	1	0	Frecuencia clk/8
0	1	1	Frecuencia clk/64
1	0	0	Frecuencia clk/256
1	0	1	Frecuencia clk/1024
1	1	0	Pin T1 en flanco de bajada
1	1	1	Pin T1 en flanco de subida

Bit	7	6	5	4	3	2	1	0	
(0x81)	ICNC1	ICES1	–	WGM13	WGM12	CS12	CS11	CS10	TCCR1B
Read/Write	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Bits de selección del Reloj

Bit	7	6	5	4	3	2	1	0	
(0x6F)	–	–	ICIE1	–	–	OCIE1B	OCIE1A	TOIE1	TIMSK1
Read/Write	R	R	R/W	R	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Bit	7	6	5	4	3	2	1	0	
0x16 (0x36)	–	–	ICF1	–	–	OCF1B	OCF1A	TOV1	TIFR1
Read/Write	R	R	R/W	R	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

WGM12	0 : Cuenta desde 0 hasta 0xFFFF 1 : Cuenta desde 0 hasta que sea igual que OCR1A
TOV1	Se activa cuando el contdor llega a 0xFFFF
OCF1	Se activa cuando el contador llega a OCR1A
TOV1	1: Interupción TIMER1 OVF habilitada 0: Interupción TIMER1 OVF deshabilitada
OCIEA1	1: Interupción TIMER1 COMPA habilitada 0: Interupción TIMER1 COMPA deshabilitada

VectorNo.	Program Address[2]	Source	Interrupt Definition
1	0x0000[1]	RESET	External Pin, Power-on Reset, Brown-out Reset and Watchdog System Reset
2	0x0002	INT0	External Interrupt Request 0
3	0x0004	INT1	External Interrupt Request 1
4	0x0006	PCINT0	Pin Change Interrupt Request 0
5	0x0008	PCINT1	Pin Change Interrupt Request 1
6	0x000A	PCINT2	Pin Change Interrupt Request 2
7	0x000C	WDT	Watchdog Time-out Interrupt
8	0x000E	TIMER2 COMPA	Timer/Counter2 Compare Match A
9	0x0010	TIMER2 COMPB	Timer/Counter2 Compare Match B
10	0x0012	TIMER2 OVF	Timer/Counter2 Overflow
11	0x0014	TIMER1 CAPT	Timer/Counter1 Capture Event
12	0x0016	TIMER1 COMPA	Timer/Counter1 Compare Match A
13	0x0018	TIMER1 COMPB	Timer/Coutner1 Compare Match B
14	0x001A	TIMER1 OVF	Timer/Counter1 Overflow
15	0x001C	TIMER0 COMPA	Timer/Counter0 Compare Match A
16	0x001E	TIMER0 COMPB	Timer/Counter0 Compare Match B
17	0x0020	TIMER0 OVF	Timer/Counter0 Overflow
18	0x0022	SPI, STC	SPI Serial Transfer Complete
19	0x0024	USART, RX	USART Rx Complete
20	0x0026	USART, UDRE	USART, Data Register Empty
21	0x0028	USART, TX	USART, Tx Complete
22	0x002A	ADC	ADC Conversion Complete
23	0x002C	EE READY	EEPROM Ready
24	0x002E	ANALOG COMP	Analog Comparator
25	0x0030	TWI	2-wire Serial Interface
26	0x0032	SPM READY	Store Program Memory Ready