

ATMEL assembly utasítások

Rd	Cél regiszter (és forrás regiszter is egyben) ltt képződik az eredmény	0≤Rd≤31 ; 16≤Rd≤31
Rr	Forrás regiszter	0≤Rr≤31 ; 16≤Rr≤31
K	Konstans	0≤K≤255
k	konstans cím	-2K≤k≤2K ; -64≤k≤63
b	bit sorszáma	0≤b≤7
q	offset	0≤q≤63
P	portcím	0≤q≤31 ; 0≤q≤63

Mnemoik	Operandus	Leírás	Művelet	Flagek	Ciklus
ADC	Rd , Rr	Két regiszter összeadása átvitel bittel	$Rd \leftarrow Rd + Rr + C$	Z,C,N,V,H	1
ADD	Rd , Rr	Két regiszter összeadása	$Rd \leftarrow Rd + Rr$	Z,C,N,V,H	1
ADIW	Rdl , K	Regiszterpárhoz konstans hozzáadása $0 \leq K \leq 63$	$Rdh:Rdl \leftarrow Rdh:Rdl + K$	Z,C,N,V,S	2
AND	Rd , Rr	Két regiszter ÉS kapcsolata	$Rd \leftarrow Rd \cdot Rr$	Z,N,V	1
ANDI	Rd , K	Regiszter és konstans ÉS kapcsolata	$Rd \leftarrow Rd \cdot K$	Z,N,V	1
ASR	Rd	aritmetikai jobbra léptetés, 7.bit helyén marad, 0.bitet a Carryban tárolja	$Rd(n) \leftarrow Rd(n+1), n=0..6$	Z,C,N,V	1
BCLR	b	SREG-ben 0-ba állítja a kívánt bitet	$SREG(s) \leftarrow 0$	SREG(s)	1
BLD	Rd , b	regiszterbe adott helyére írja az SREG T flagjét	$Rd(b) \leftarrow T$	None	1
BRBC	b , k	Ugrik, ha SREG-ben tesztelt bit 0	$\text{if } (SREG(s) = 0) \text{ then } PC \leftarrow PC + k + 1$	None	1/2
BRBS	b , k	Ugrik, ha SREG-ben tesztelt bit 1	$\text{if } SREG(s) = 1 \text{ then } PC \leftarrow PC + k + 1$	None	1/2
BRCC	k	Ugrik, ha SREG-ben Carry=1 [megegyezik: BRBS 0,k]	$\text{if } C = 0 \text{ then } PC \leftarrow PC + k + 1$	None	1/2
BRCS	k	Ugrik, ha SREG-ben Carry=1 [megegyezik: BRBS 0,k]	$\text{if } C = 1 \text{ then } PC \leftarrow PC + k + 1$	None	1/2
BREAK			Csak CHIP debugolóhoz	None	N/A
BREQ	k	Ugrik, ha egyenlő (CP,CPI,SUB,SUBI utasítások után használható) [megegyezik:BRBS 1,k]	$\text{if } Rd = Rr \text{ (Z = 1) then } PC \leftarrow PC + k + 1$	None	1/2
BRGE	k	Ugrik, ha Rd nagyobb vagy egyenlő,mint Rr (előjeles) (CP,CPI,SUB,SUBI utasítások után használható) [megegyezik: BRBC 4,k]	$\text{if } (N \oplus V = 0) \text{ then } PC \leftarrow PC + k + 1$	None	1/2
BRHC	k	Ugrik, ha Hafl Carry bit (SREG-ben) =0 [megegyezik:BRBC 5,k]	$\text{if } (H = 0) \text{ then } PC \leftarrow PC + k + 1$	None	1/2
BRHS	k	Ugrik, ha Hafl Carry bit (SREG-ben) =1 [megegyezik:BRBS 5,k]	$\text{if } (H = 1) \text{ then } PC \leftarrow PC + k + 1$	None	1/2
BRID	k	Ugrik, ha az Interrupt (megszakítás [I]) bit (SREG-ben) = 0 [megegyezik: BRBC 7,k]	$\text{if } (I = 0) \text{ then } PC \leftarrow PC + k + 1$	None	1/2
BRIE	k	Ugrik, ha az Interrupt (megszakítás [I]) bit (SREG-ben) = 1 [megegyezik: BRBS 7,k]	$\text{if } (I = 1) \text{ then } PC \leftarrow PC + k + 1$	None	1/2
BRLO	k	Ugrik, ha Rd kisebb, mint Rr (előjel nélküli) (CP,CPI,SUB,SUBI utasítások után használható) [megegyezik: BRBS 0,k]	$\text{if } Rd < Rr \text{ (C = 1) then } PC \leftarrow PC + k + 1$	None	1/2
BRLT	k	Ugrik, ha Rd kisebb mint Rr (előjeles) (CP,CPI,SUB,SUBI utasítások után használható) [megegyezik: BRBS 4,k]	$\text{if } (N \oplus V = 1) \text{ then } PC \leftarrow PC + k + 1$	None	1/2
BRMI	k	Ugrik, ha N=1, negatív érték képződött. [megegyezik: BRBS 2,k]	$\text{if } (N = 1) \text{ then } PC \leftarrow PC + k + 1$	None	1/2
BRNE	k	Ugrik, ha NEM egyenlő (CP,CPI,SUB,SUBI utasítások után használható) [megegyezik: BRBC 1,k]	$\text{if } Rd \neq Rr \text{ (Z = 0) then } PC \leftarrow PC + k + 1$	None	1/2
BRPL	k	Ugrik, ha N=0, negatív érték képződött. [megegyezik: BRBC 2,k]	$\text{if } (N = 0) \text{ then } PC \leftarrow PC + k + 1$	None	1/2
BRSH	k	Ugrik, ha Rd nagyobb vagy egyenlő,mint Rr (előjel nélküli) (CP,CPI,SUB,SUBI utasítások után használható) [megegyezik: BRBC 0,k]	$\text{if } Rd \geq Rr \text{ (C = 0) then } PC \leftarrow PC + k + 1$	None	1/2
BRTC	k	Ugrik, ha T flag (SREG-ben) = 0 [megegyezik:BRBC 6,k]	$\text{if } (T = 0) \text{ then } PC \leftarrow PC + k + 1$	None	1/2
BRTS	k	Ugrik, ha T flag (SREG-ben) = 1 [megegyezik:BRBS 6,k]	$\text{if } (T = 1) \text{ then } PC \leftarrow PC + k + 1$	None	1/2
BRVC	k	Ugrik, ha az Overflow (túlszordulás [V]) bit (SREG-ben) = 0 [megegyezik: BRBC 3,k]	$\text{if } (V = 0) \text{ then } PC \leftarrow PC + k + 1$	None	1/2
BRVS	k	Ugrik, ha az Overflow (túlszordulás [V]) bit (SREG-ben) = 1 [megegyezik: BRBS 3,k]	$\text{if } (V = 1) \text{ then } PC \leftarrow PC + k + 1$	None	1/2
BSET	b	SREG-ben 1-be állítja a kívánt bitet	$SREG(s) \leftarrow 1$	SREG(s)	1
BST	Rr , b	regiszter adott bitjét tárolja, az SREG T flagjében	$T \leftarrow Rr(b)$	T	1
CALL	k	Szubrutin hívása $0 \leq k \leq 64k$	$PC \leftarrow k$	None	4
CBI	P , b	Törli az adott bitet az IO regiszterben $0 \leq P \leq 31$	$I/O(P,b) \leftarrow 0$	None	2
CBR	Rd , K	Bit(ek) törlése adott regiszterben	$Rd \leftarrow Rd \cdot (\$FF - K)$	Z,N,V	1
CLC		Carry törlése	$C \leftarrow 0$	C	1
CLH		Half Carry törlése	$H \leftarrow 0$	H	1
CLI		Megszakítások tiltása	$I \leftarrow 0$	I	1
CLN		Negatív flag törlése	$N \leftarrow 0$	N	1
CLR	Rd	Regiszter törlése	$Rd \leftarrow Rd \oplus Rd$	Z,N,V	1
CLS		Előjel jelző bit törlése	$S \leftarrow 0$	S	1
CLT		T flag törlése	$T \leftarrow 0$	T	1
CLV		Kettes komplementum jelző bit törlése	$V \leftarrow 0$	V	1
CLZ		Zero (0) jelző bit törlése	$Z \leftarrow 0$	Z	1
COM	Rd	Egyes komplementum	$Rd \leftarrow \$FF - Rd$	Z,C,N,V	1
CP	Rd , Rr	Összehasonlít Rd,Rr, értékük nem módosul, utána elágazást lehet használni.	$Rd - Rr$	Z, N,V,C,H	1
CPC	Rd , Rr	Összehasonlít Rd,Rr a Carry értékének figyelembevételével.Értékük nem módosul, utána elágazást lehet használni.	$Rd - Rr - C$	Z, N,V,C,H	1
CPI	Rd , K	Összehasonlít Rd-t egy Konstanssal.Értéke nem módosul, utána elágazást lehet használni	$Rd - K$	Z, N,V,C,H	1
CPSE	Rd , Rr	Összehasonlít Rd,Rr, és a következő utasítást kihagyja, ha Rd=Rr	$(Rd = Rr) PC \leftarrow PC + 2 \text{ or } 3$	None	1/2/3
DEC	Rd	Egyel csökkentti Rd-t	$Rd \leftarrow Rd - 1$	Z,N,V	1
EOR	Rd , Rr	Két regiszter KIZÁRÓ VAGY kapcsolata	$Rd \leftarrow Rd \oplus Rr$	Z,N,V	1
FMUL	Rd , Rr	Szorzás előjel nélkül és az eredmény balra léptetése 1-el	$R1:R0 \leftarrow (Rd \times Rr) \ll 1$	Z,C	2
FMULS	Rd , Rr	Két előjeles szám szorzása és az eredmény balra léptetése 1-el $16 \leq Rd \leq 23 ; 16 \leq Rr \leq 23$	$R1:R0 \leftarrow (Rd \times Rr) \ll 1$	Z,C	2
FMULSU	Rd , Rr	Előjeles szám szorzása előjel nélküli számmal és az eredmény balra léptetése 1-el $16 \leq Rd \leq 23 ; 16 \leq Rr \leq 23$	$R1:R0 \leftarrow (Rd \times Rr) \ll 1$	Z,C	2
ICALL		A Z regiszterben levő cím szubrutinjának hívása	$PC \leftarrow Z$	None	3
IJMP		A Z regiszterben levő címre ugrik	$PC \leftarrow Z$	None	2
IN	Rd , P	Betölti Rd-be a IO hely értékét (port,időzítők,konfig regiszter) $0 \leq P \leq 63$	$Rd \leftarrow P$	None	1
INC	Rd	Egyel növeli Rd-t	$Rd \leftarrow Rd + 1$	Z,N,V	1
JMP	k	Ugrik adott címre $0 \leq k \leq 4M$	$PC \leftarrow k$	None	3
LD	Rd , X	Betölti a X pointer által(XH:R27,XL:R26) mutatott címet Rd-be	$Rd \leftarrow (X)$	None	2
LD	Rd , X+	Betölti a X pointer által(XH:R27,XL:R26) mutatott KÖVETKEZŐ cím tartalmát Rd-be	$Rd \leftarrow (X), X \leftarrow X + 1$	None	2
LD	Rd , -X	Betölti a X pointer által(XH:R27,XL:R26) mutatott ELŐZŐ cím tartalmát Rd-be	$X \leftarrow X - 1, Rd \leftarrow (X)$	None	2
LD	Rd , Y	Betölti a Y pointer által(XH:R29,XL:R28) mutatott cím tartalmát Rd-be	$Rd \leftarrow (Y)$	None	2

LD	Rd , Y+	Betölti a Y pointer által(XH:R29,XL:R28) mutatott KÖVETKEZŐ cím tartalmát Rd-be	$Rd \leftarrow (Y), Y \leftarrow Y + 1$	None	2
LD	Rd , -Y	Betölti a Y pointer által(XH:R29,XL:R28) mutatott ELŐZŐ cím tartalmát Rd-be	$Y \leftarrow Y - 1, Rd \leftarrow (Y)$	None	2
LD	Rd , Z	Betölti a Z pointer által(XH:R31,XL:R30) mutatott cím tartalmát Rd-be	$Rd \leftarrow (Z)$	None	2
LD	Rd , Z+	Betölti a Z pointer által(XH:R31,XL:R30) mutatott KÖVETKEZŐ cím tartalmát Rd-be	$Rd \leftarrow (Z), Z \leftarrow Z+1$	None	2
LD	Rd , -Z	Betölti a Z pointer által(XH:R31,XL:R30) mutatott ELŐZŐ cím tartalmát Rd-be	$Z \leftarrow Z - 1, Rd \leftarrow (Z)$	None	2
LDD	Rd , Y+q	Betölti a Y pointer által(XH:R29,XL:R28) mutatott q-val eltölt cím tartalmát Rd-be	$Rd \leftarrow (Y + q)$	None	2
LDD	Rd , Z+q	Betölti a Z pointer által(XH:R31,XL:R30) mutatott q-val eltölt cím tartalmát Rd-be	$Rd \leftarrow (Z + q)$	None	2
LDI	Rd , K	Konstans értéket tölt az Rd-be (16-31regisztereknél)	$Rd \leftarrow K$	None	1
LDS	Rd , k	Memóritartományból tölt adatot Rd-be (I/O,SRAM, külső SRAM) 0sk≤65535	$Rd \leftarrow (k)$	None	2
LPM		R0 regiszterbe tölti a Z pointer által(XH:R31,XL:R30) mutatott címen lévő adatot	$R0 \leftarrow (Z)$	None	3
LPM	Rd , Z	Regiszterbe tölti a Z pointer által(XH:R31,XL:R30) mutatott címen lévő adatot	$Rd \leftarrow (Z)$	None	3
LPM	Rd , Z+	Regiszterbe tölti a Z pointer által(XH:R31,XL:R30) mutatott következő címen lévő adatot	$Rd \leftarrow (Z), Z \leftarrow Z+1$	None	3
LSL	Rd	logikai léptetés balra, 0-t léptet be 0.bitnek, 7 bit a Carryban	$Rd(n+1) \leftarrow Rd(n), Rd(0) \leftarrow 0$	Z,C,N,V	1
LSR	Rd	logikai léptetés jobbra, 0-t léptet be 7.bitnek, 0 bit a Carryban	$Rd(n) \leftarrow Rd(n+1), Rd(7) \leftarrow 0$	Z,C,N,V	1
MOV	Rd , Rr	Rr értékét Rd-be másolja, Rr-t nem módosítja	$Rd \leftarrow Rr$	None	1
MOVW	Rd , Rr	16 bites másolás Rd-be, Rr-t. $Rd \in \{0,2,\dots,30\}, Rr \in \{0,2,\dots,30\}$	$Rd+1:Rd \leftarrow Rr+1:Rr$	None	1
MUL	Rd , Rr	Szorzás előjel nélkül	$R1:R0 \leftarrow Rd \times Rr$	Z,C	2
MULS	Rd , Rr	Két előjeles szám szorzása	$R1:R0 \leftarrow Rd \times Rr$	Z,C	2
MULSU	Rd , Rr	Előjeles szám szorzása előjel nélküli számmal $16 \leq Rd \leq 23; 16 \leq Rr \leq 23$	$R1:R0 \leftarrow Rd \times Rr$	Z,C	2
NEG	Rd	Kettes komplementum	$Rd \leftarrow \$00 - Rd$	Z,C,N,V,H	1
NOP		nincs utasítás		None	1
OR	Rd , Rr	Két regiszter VAGY kapcsolata	$Rd \leftarrow Rd \vee Rr$	Z,N,V	1
ORI	Rd , K	Regiszter és konstans VAGY kapcsolata	$Rd \leftarrow Rd \vee K$	Z,N,V	1
OUT	P , Rr	Kirítja Rd értékét az adott IO helyre (port,időzítők,konfig regiszter) 0≤P≤31	$P \leftarrow Rr$	None	1
POP	Rd	Stack legfelső értékét betölti adott regiszterbe	$Rd \leftarrow STACK$	None	2
PUSH	Rr	Stack legfelső helyére elmenti adott regiszter értékét	$STACK \leftarrow Rr$	None	2
RCALL	k	Szubrutin hívása	$PC \leftarrow PC + k + 1$	None	3
RET		Szubrutin vége,visszatérési pont	$PC \leftarrow STACK$	None	4
RETI		Megszakítás vége	$PC \leftarrow STACK$	I	4
RJMP	k	ugrás adott címre	$PC \leftarrow PC + k + 1$	None	2
ROL	Rd	forgatás balra, Carryt lépteti be 0. bitnek, 7. bitet a Carryban tárolja	$Rd(0) \leftarrow C, Rd(n+1) \leftarrow Rd(n), C \leftarrow Rd(7)$	Z,C,N,V	1
ROR	Rd	forgatás jobbra Carryt lépteti be 7. bitnek, 0. bitet a Carryban tárolja	$Rd(7) \leftarrow C, Rd(n) \leftarrow Rd(n+1), C \leftarrow Rd(0)$	Z,C,N,V	1
SBC	Rd , Rr	Két regiszter és átvitel különbsége	$Rd \leftarrow Rd - Rr - C$	Z,C,N,V,H	1
SBCI	Rd , K	Regiszterből konstans és átvitel kivonása	$Rd \leftarrow Rd - K - C$	Z,C,N,V,H	1
SBI	P , b	Beállítja az adott bitet az IO regiszterben 0≤P≤31	$I/O(P,b) \leftarrow 1$	None	2
SBIC	P , b	Soronkövetkező utasítást kihagyja, ha adott I/O adat bit értéke 0	$\text{if } (P(b)=0) PC \leftarrow PC + 2 \text{ or } 3$	None	1/2/3
SBIS	P , b	Soronkövetkező utasítást kihagyja, ha adott I/O adat bit értéke 1	$\text{if } (P(b)=1) PC \leftarrow PC + 2 \text{ or } 3$	None	1/2/3
SBIW	Rdl , K	Regiszterpárhoz konstans kivonása 0≤K≤63	$Rdh:Rdl \leftarrow Rdh:Rdl - K$	Z,C,N,V,S	2
SBR	Rd , K	Bit(ek) beállítása adott regiszterben	$Rd \leftarrow Rd \vee K$	Z,N,V	1
SBRC	Rr , b	Soronkövetkező utasítást kihagyja, ha adott bit értéke 0	$\text{if } (Rr(b)=0) PC \leftarrow PC + 2 \text{ or } 3$	None	1/2/3
SBRS	Rr , b	Soronkövetkező utasítást kihagyja, ha adott bit értéke 1	$\text{if } (Rr(b)=1) PC \leftarrow PC + 2 \text{ or } 3$	None	1/2/3
SEC		Carry 1-be állítása	$C \leftarrow 1$	C	1
SEH		Half Carry 1-be állítása	$H \leftarrow 1$	H	1
SEI		Megszakítások engedélyezése	$I \leftarrow 1$	I	1
SEN		Negatív flag 1-be állítása	$N \leftarrow 1$	N	1
SER	Rd	Regiszer \$FF-be állítása	$Rd \leftarrow \$FF$	None	1
SES		Előjel jelző bit 1-be állítása	$S \leftarrow 1$	S	1
SET		T flag 1-be állítása	$T \leftarrow 1$	T	1
SEV		Kettes komplementum jelző bit 1-be állítása	$V \leftarrow 1$	V	1
SEZ		Zero (0) jelző bit 1-be állítása	$Z \leftarrow 1$	Z	1
SLEEP		Lásd adatlapon a Sleep mode leírást		None	1
SPM		(Z) ← R1:R0		None	-
ST	- X , Rr	Menti a X pointer által(XH:R27,XL:R26) mutatott ELŐZŐ címre Rr-t	$X \leftarrow X - 1, (X) \leftarrow Rr$	None	2
ST	X , Rr	Menti a X pointer által(XH:R27,XL:R26) mutatott címre Rr-t	$(X) \leftarrow Rr$	None	2
ST	X+ , Rr	Menti a X pointer által(XH:R27,XL:R26) mutatott KÖVETKEZŐ címre Rr-t	$(X) \leftarrow Rr, X \leftarrow X + 1$	None	2
ST	Y , Rr	Menti a Y pointer által(XH:R29,XL:R28) mutatott címre Rr-t	$(Y) \leftarrow Rr$	None	2
ST	Y+ , Rr	Menti a Y pointer által(XH:R29,XL:R28) mutatott KÖVETKEZŐ címre Rr-t	$(Y) \leftarrow Rr, Y \leftarrow Y + 1$	None	2
ST	- Y , Rr	Menti a Y pointer által(XH:R29,XL:R28) mutatott ELŐZŐ címre Rr-t	$Y \leftarrow Y - 1, (Y) \leftarrow Rr$	None	2
ST	Z , Rr	Menti a Z pointer által(XH:R31,XL:R30) mutatott címre Rr-t	$(Z) \leftarrow Rr$	None	2
ST	Z+ , Rr	Menti a Z pointer által(XH:R31,XL:R30) mutatott KÖVETKEZŐ címre Rr-t	$(Z) \leftarrow Rr, Z \leftarrow Z + 1$	None	2
ST	-Z , Rr	Menti a Z pointer által(XH:R31,XL:R30) mutatott ELŐZŐ címre Rr-t	$Z \leftarrow Z - 1, (Z) \leftarrow Rr$	None	2
STD	Y+q , Rr	Menti a Y pointer által(XH:R29,XL:R28) mutatott q-val eltölt címre Rr-t	$(Y + q) \leftarrow Rr$	None	2
STD	Z+q , Rr	Menti a Z pointer által(XH:R31,XL:R30) mutatott q-val eltölt címre Rr-t	$(Z + q) \leftarrow Rr$	None	2
STS	k , Rr	Memóritartományba menti adatot Rd-ből (I/O,SRAM, külső SRAM) Rr nem módosul. 0sk≤65535	$(k) \leftarrow Rr$	None	2
SUB	Rd , Rr	Két regiszter különbsége	$Rd \leftarrow Rd - Rr$	Z,C,N,V,H	1
SUBI	Rd , K	Regiszterből konstans kivonása	$Rd \leftarrow Rd - K$	Z,C,N,V,H	1
SWAP	Rd	Alsó és felső 4 bitet felcseréli	$Rd(3..0) \leftarrow Rd(7..4), Rd(7..4) \leftarrow Rd(3..0)$	None	1
TST	Rd	Ellenőrés 0-ra vagy negatívra	$Rd \leftarrow Rd \cdot Rd$	Z,N,V	1
WDR		Lásd adatlapon a WatchDog időzítő leírását		None	1

Ez a táblázat a szerző engedélye nélkül szabadon másolható,terjeszthető és felhasználható módosítások nélkül.
Készítette: kiborg@tvn.hu