

Symboly operandov assembleru MCU-51			
Symbol	Počet bitov	Význam	Príklad
#data	8	konštanta	#20H, #0EFH, #32D, #00100000B, #'a',
#data16	16	konštanta	#1234H
adr	8	adresa v int. RAM	20H, 123D
rel	8	relatívna adresa	< -128, 127 >
adr11	11	adresa (2kB)	
adr16	16	adresa (64kB)	1234H
bit	8	bitová adresa	20H
A	8	akumulátor	
B	8	pomocný register	
Rn	8	register n=0,1,...,7	R0, R1, ... R7
@Ri	8	nepriama adr.-smerník, i=0,1	@R0, @R1
DPTR	16	Data PointeR (64kB)	
C	1	Carry - príznak prenosu	

### Inštrukcie pre presun údajov MCU-51

Meno	Cieľ	Zdroj	Význam	B/C
MOV	A	, #data	(A) <- #data	2/1
	A	, adr	(A) <- (adr)	2/1
	A	, Rn	(A) <- (Rn)	1/1
	A	, @Ri	(A) <- ((Ri))	1/1
MOV	Rn	, A	(Rn) <- (A)	1/1
	Rn	, #data	(Rn) <- #data	2/1
	Rn	, adr	(Rn) <- (adr)	2/2
MOV	@Ri	, A	((Ri)) <- A	1/1
	@Ri	, #data	((Ri)) <- #data	2/1
	@Ri	, adr	((Ri)) <- (adr)	2/2
MOV	adr	, A	(adr) <- A	2/1
	adr	, Rn	(adr) <- (Rn)	2/2
	adr	, @Ri	(adr) <- ((Ri))	2/2
	adr	, #data	(adr) <- #data	3/2
	adr	, adr	(adr) <- (adr)	3/2
MOV	C	, bit	(C) <- (bit)	2/1
	bit	, C	(bit) <- (C)	2/2
MOV	DPTR	, #data16	(DPTR) <- #data16	3/2
MOVC	A	, @A+DPTR	(A) <- ((A) + (DPTR))	1/2
	A	, @A+PC	(A) <- ((A) + (PC))	1/2
MOVX	A	, @DPTR	(A) <- ((DPTR))	1/2
	A	, @Ri	(A) <- ((Ri))	1/2
	@DPTR	, A	((DPTR)) <- (A)	1/2
	@Ri	, A	((Ri)) <- (A)	1/2
XCH	A	, Rn	(A) <-> (Rn)	1/1
	A	, adr	(A) <-> (adr)	2/1
	A	, @Ri	(A) <-> ((Ri))	1/1
XCHD	A	, @Ri	(A3-0) <-> ((Ri3-0))	1/1
SWAP	A		(A3-0) <-> (A7-4)	1/1
PUSH	adr		(SP) <- (SP) + 1 ((SP)) <- (adr)	2/2
POP	adr		(adr) <- ((SP)) (SP) <- (SP) - 1	2/2

### Vysvetlivky:

B - počet bajtov v kóde inštrukcie

C - počet cyklov potrebných pre vykonanie inštrukcie

F - (flag) príznaky ovplyvnené inštrukciou

### Mená inštrukcií MCU-51

Meno	Opis	Význam
MOV	MOVe	presu vo vnútornej RAM
MOVX	MOVe eXternal	presun z/do externej RAM
MOVC	MOVe Code	presun z pamäti programu
XCH	eXCHange	vymena bajtov
XCHD	eXCHange Decimal	vymena dolných 4 bitov
SWAP	SWAP	prehodenie dolnej a hornej štvorice
PUSH	SPUSH	odlož do zásobníka
POP	POP	výber zo zásobníka
ADD	ADD	súčet
ADDC	ADD with Carry	súčet s prenosom
SUBB	SUBtraction with Borrow	odčítanie s výpžičkou
MUL	MULTiply	násobenie
DIV	DIvision	delenie
INC	INCRement	inkrementácia
DEC	DECRement	dekrementácia
DA	DecimAl Adjust	dekadická korekcia
ANL	And Logical	logický súčin
ORL	OR Logical	logický súčet
XRL	eXclusive oR Logical	exkluzívny súčet
CPL	ComPLement	komplement
CLR	CLeaR	nulovanie
SETB	SET Bit	nastavenie
JMP	JuMP	skok
AJMP	Absolute JUMP	absolútny skok
LJMP	Long JUMP	dlhý skok
ACALL	Absolute CALL	absolútne volanie podprogramu
LCALL	Long CALL	dlhé volanie podprogramu
RET	RETurn	návrat z podprogramu
RETI	RETurn from Interrupt	návrat z prerušenia
JB	Jump if Bit is set	skok ak je bit nastavený
JNB	Jump if Not Bit is set	skok ak nie je bit nastavený
JBC	Jump if Bit is set and Clear	skok ak je bit nastavený a nulovanie bitu
JC	Jump if Carry is set	skok ak C je nastavený
JNC	Jump if Not Carry is set	skok ak C nie je nastavený
JZ	Jump if Zero	skok ak je nula
JNZ	Jump if Not Zero	skok ak nie je nula
DJNZ	Decrement and Jump if Not Zero	dekrement a skok ak nie je nula
CJNE	CCompare and Jump if Not Equal	porovnanie a skok ak nerovné
NOP	No OPeration	žiadna operácia

### Aritmetické inštrukcie MCU-51

Meno	Cieľ	Zdroj	Význam	F	B/C
ADD	A	, #data	(A) <- (A) + data	C, AC, OV	2/1
	A	, Rn	(A) <- (A) + (Rn)	C, AC, OV	1/1
	A	, @Ri	(A) <- (A) + ((Ri))	C, AC, OV	1/1
	A	, adr	(A) <- (A) + (adr)	C, AC, OV	2/1
ADDC	A	, #data	(A) <- (A) + data + C	C, AC, OV	2/1
	A	, Rn	(A) <- (A) + (Rn) + C	C, AC, OV	1/1
	A	, @Ri	(A) <- (A) + ((Ri)) + C	C, AC, OV	1/1
	A	, adr	(A) <- (A) + (adr) + C	C, AC, OV	2/1
SUBB	A	, #data	(A) <- (A) - (C) - data	C, AC, OV	2/2
	A	, Rn	(A) <- (A) - (C) - (Rn)	C, AC, OV	1/1
	A	, @Ri	(A) <- (A) - (C) - ((Ri))	C, AC, OV	1/1
	A	, adr	(A) <- (A) - (C) - (adr)	C, AC, OV	2/1
MUL	AB		(B) (A) <- (A) * (B)	OV, CY	1/4
			(A) <- fix(A/B)		
			(B) <- rem(A/B)		
DIV	AB			C, OV	1/4
DA	A		dekadická korekcia	C, AC	1/1
INC	A		(A) <- (A) + 1		1/1
	Rn		(Rn) <- (Rn) + 1		1/1
	@Ri		((Ri)) <- ((Ri)) + 1		1/1
	adr		(adr) <- (adr) + 1		2/1
	DPTR		(DPTR) <- (DPTR) + 1		1/2
DEC	A		(A) <- (A) - 1		1/1
	Rn		(Rn) <- (Rn) - 1		1/1
	@Ri		((Ri)) <- ((Ri)) - 1		1/1
	adr		(adr) <- (adr) - 1		2/1

**Logické inštrukcie MCU-51**

Meno	Cieľ	Zdroj	Význam	F	B/C
ANL	A	,Rn	(A) <- (A) AND (Rn)	-	1/1
	A	,@Ri	(A) <- (A) AND ((Ri))	-	1/1
	A	,adr	(A) <- (A) AND (adr)	-	2/1
	A	,#data	(A) <- (A) AND #data	-	2/1
	adr	,A	(adr) <- (adr) AND (A)	-	2/1
	adr	,#data	(adr) <- (adr) AND #data	-	3/2
	C	,bit	(C) <- (C) AND (bit)	C	2/2
	C	,/bit	(C) <- (C) AND /(bit)	C	2/2
ORL	A	,Rn	(A) <- (A) OR (Rn)	-	1/1
	A	,@Ri	(A) <- (A) OR ((Ri))	-	1/1
	A	,adr	(A) <- (A) OR (adr)	-	2/1
	A	,#data	(A) <- (A) OR #data	-	2/1
	adr	,A	(adr) <- (adr) OR (A)	-	2/1
	adr	,#data	(adr) <- (adr) OR #data	-	3/2
	C	,bit	(C) <- (C) OR (bit)	C	2/2
	C	,/bit	(C) <- (C) OR /(bit)	C	2/2
XRL	A	,Rn	(A) <- (A) XOR (Rn)	-	1/1
	A	,@Ri	(A) <- (A) XOR ((Ri))	-	1/1
	A	,adr	(A) <- (A) XOR (adr)	-	2/1
	A	,#data	(A) <- (A) XOR #data	-	2/1
	adr	,A	(adr) <- (adr) XOR (A)	-	2/1
	adr	,#data	(adr) <- (adr) XOR #data	-	3/2
CPL	A		(A) <- /(A)	-	1/1
	C		(C) <- /(C)	C	1/1
	bit		(bit) <- /(bit)	-	2/1
CLR	A		(A) <- 0	-	1/1
	C		(C) <- 0	C	1/1
	bit		(bit) <- 0	-	2/1
SETB	C		(C) <- 1	C	1/1
	bit		(bit) <- 1	-	2/1
RL	A		(A <sub>0</sub> ) <- (A <sub>7</sub> ), (A <sub>n+1</sub> ) <- (A <sub>n</sub> ), n=0-6	-	1/1
RR	A		(A <sub>7</sub> ) <- (A <sub>0</sub> ), (A <sub>n</sub> ) <- (A <sub>n+1</sub> ), n=0-6	-	1/1
RLC	A		(A <sub>0</sub> ) <- (C), (C) <- (A <sub>7</sub> ) (A <sub>n+1</sub> ) <- (A <sub>n</sub> ), n=0-6	C	1/1
RRC	A		(A <sub>7</sub> ) <- (C), (C) <- (A <sub>0</sub> ) (A <sub>n</sub> ) <- (A <sub>n+1</sub> ), n=0-6	C	1/1

**Riadiace inštrukcie MCU-51**

Meno	Operand	Význam	F	B/C
JMP	@A+DPTR	(PC) <- (A) + (DPTR)	-	1/2
AJMP	adr11	(PC) <- (PC) + 2 (PC10-0) <- adr11	-	2/2
LJMP	adr16	(PC) <- adr16	-	3/2
LCALL	adr16	(PC) <- (PC) + 3 (SP) <- (SP) + 1, ((SP)) <- (PC7-0) (SP) <- (SP) + 1, ((SP)) <- (PC15-8) (PC) <- adr16	-	3/2
ACALL	adr11	(PC) <- (PC) + 2 (SP) <- (SP) + 1, ((SP)) <- (PC7-0) (SP) <- (SP) + 1, ((SP)) <- (PC15-8) (PC10-0) <- adr11	-	2/2
RET		(PC15-8) <- ((SP)) (PC7-0) <- ((SP) - 1) (SP) <- (SP) - 2	-	1/2
RETI		(PC15-8) <- ((SP)) (PC7-0) <- ((SP) - 1) (SP) <- (SP) - 2	-	1/2
JB	bit,rel	ak (bit)=1, (PC) <- (PC) + 3 + rel ak (bit)=0, (PC) <- (PC) + 3	-	3/2
JNB	bit,rel	ak (bit)=0, (PC) <- (PC) + 3 + rel ak (bit)=1, (PC) <- (PC) + 3	-	3/2
JBC	bit,rel	ak (bit)=1, (bit) <- 0, (PC) <- (PC) + 3 + rel ak (bit)=0, (PC) <- (PC) + 3	-	3/2
JC	rel	ak (C)=1, (PC) <- (PC) + 2 + rel ináč (PC) <- (PC) + 2	-	2/2
JNC	rel	ak (C)=0, (PC) <- (PC) + 2 + rel inac (PC) <- (PC) + 2	-	2/2
JZ	rel	ak (A)=0, (PC) <- (PC) + 2 + rel inac (PC) <- (PC) + 2	-	2/2
JNZ	rel	ak (A) > 0, (PC) <- (PC) + 2 + rel inac (PC) <- (PC) + 2	-	2/2
DJNZ	Rn,rel	(Rn) <- (Rn) - 1 ak (Rn) > 0, (PC) <- (PC) + 2 + rel ak (Rn) = 0, (PC) <- (PC) + 2	-	2/2
	adr,rel	(adr) <- (adr) - 1 ak (adr) > 0, (PC) <- (PC) + 2 + rel ak (adr) = 0, (PC) <- (PC) + 2	-	2/2
CJNE	A,adr,rel	ak (A) > (adr), (PC) <- (PC) + 3 + rel ak (A) < (adr), (C) <- 1, inac (C) <- 0 ak (A) = (adr), (PC) <- (PC) + 3	C	3/2
	A,#data,rel	ak (A) > #data, (PC) <- (PC) + 3 + rel ak (A) < #data, (C) <- 1, inac (C) <- 0 ak (A) = #data, (PC) <- (PC) + 3	C	3/2
	Rn,#data,rel	ak (Rn) > #data, (PC) <- (PC) + 3 + rel ak (Rn) < #data, (C) <- 1, inac (C) <- 0 ak (Rn) = #data, (PC) <- (PC) + 3	C	3/2
	@Ri,#data,rel	ak (@Ri) > #data, (PC) <- (PC) + 3 + rel ak (@Ri) < #data, (C) <- 1, inac (C) <- 0 ak (@Ri) = #data, (PC) <- (PC) + 3	C	3/2
NOP		(PC) <- (PC) + 1	-	1/1

**VYBRANÉ DIREKTÍVY ASEMLERU 51**

Meno	Význam	Syntax	Príklad
EQU	Priradzuje menu výraz	meno EQU výraz	VSTUP EQU 20H POCET EQU 31
BIT	Priradí menu bitovú adresu	meno BIT bit	X: DS 1 ; rezervuje byte Y0 BIT X.0 ; priradí bit. adr. Y1 BIT Y0+1 ; priradí bit. adr. Z BIT 20H ; priradí bit adr.
DATA	Priradí menu adresu int. RAM	meno DATA adr	X1 DATA 20H X2 DATA 21H
XDATA	Priradí menu adresu ext. RAM	meno XDATA adr16	CAS DATA 1234H
DB	Ukladá do pamäte programu 8 bitové hodnoty	návestie: DB výraz	TEXT: DB 'HELLO WORLD!' VAL: DB 10, 20H, 10001000B
DW	Ukladá do pamäte programu 16 bitové hodnoty	návestie: DW výraz	VAL: DW 1243, 10000, 'PM', 'AM'
DS	Rezervuje určitý počet slabík v pamäti	návestie: DS výraz	TAB: DS 10 ; rezervuje 10 slabík
DBIT	Rezervuje určitý počet bitov pamäti	návestie: DBIT výraz	BITY: DBIT 10 ; rezervuje 10 bitov
ORG	Nastavuje počítadlo adres na novú hodnotu	ORG výraz	ORG 100H
END	Identifikácia konca zdrojového textu	END	Musí byť posledný príkaz v programe
PUBLIC	Zverejní mená z modulu	PUBLIC zoznam mien	PUBLIC INT2ASC, TAB,
EXTRN	Definuje vonkajšie mená	EXTRN typ segmentu(zoznam mien)	EXTRN CODE(INT2ASC,TAB)