

SETUL DE INSTRUCȚIUNI

Toate instrucțiunile μP 8085 au codul operației pe un singur octet. Din totalul celor $2^8 = 256$ de combinații posibile se folosesc numai 246. Principalele categorii de instrucțiuni sunt:

- **instrucțiuni de transfer** - copierea datelor între registre sau între registre și memorie; aceste instrucțiuni nu afectează fanioanele de condiție.
- **instrucțiuni aritmetice** - operații aritmetice de adunare și scădere sau de incrementare/ decrementare cu cuvinte din registre sau din memorie. Instrucțiunile din această categorie afectează toate fanioanele de condiții.
- **instrucțiuni logice** - operații logice asupra cuvintelor binare. Toți indicatorii de condiție sunt afectați (CY și AC sunt resetați).
- **instrucțiuni de ramificare** - schimbarea succesiunii uniforme de execuție a instrucțiunilor. Ramificările pot fi **necondiționate** sau **condiționate** de valorile fanioanelor. Există 8 condiții distincte referitoare la valorile fanioanelor **Z, CY, P și S**:
 - **NZ (Not Zero)** - rezultatul ultimei operații aritmetice/logice este diferit de 0 ($Z=0$);
 - **Z (Zero)** - rezultatul este 0 ($Z=1$);
 - **NC (Not Carry)** - nu a existat transport/împrumut ($CY=0$);
 - **C (Carry)** - a existat transport/împrumut ($CY=1$);
 - **PO (Parity Odd)** - rezultatul are un număr impar de biți „1” ($P=0$);
 - **PE (Parity Even)** - numărul de biți „1” al rezultatului este par ($P=1$);
 - **P (Plus)** - în acumulator se află un număr nenegativ ($S=A_7=0$);
 - **M (Minus)** - în acumulator se află un număr negativ ($S=A_7=1$).
- **instrucțiuni de lucru cu stiva, de I/E și de comandă** a microprocesorului – permit accesul la stivă, la porturile de I/E și controlează regimurile microprocesorului.

Nr. crt.	Mnemonica instrucțiunii	Descrierea instrucțiunii	Mod(uri) de adresare	Flaguri afectate	CM/stări
Instrucțiuni de transfer					
1	MOV r_1, r_2	MOVE reg. to reg.	$(r_1) \leftarrow (r_2)$	la registru	- 1/4
2	MOV r, M	MOVE Mem. to reg.	$(r) \leftarrow ((H)(L))$	ind.+la reg.	- 2/7
3	MOV M, r	MOVE reg. to Mem.	$((H)(L)) \leftarrow (r)$	la reg.+ind.	- 2/7
4	MVI $r, data8$	MOVE to reg. Immed.	$(r) \leftarrow data8$	imed.+la reg.	- 2/7
5	MVI $M, data8$	MOVE to Mem. Immed.	$((H)(L)) \leftarrow data8$	imed.+ind.	- 3/10
6	LXI $rp, data16$	Load reg. pair Immed.	$(r_L) \leftarrow \text{low}(data16)$ $(r_H) \leftarrow \text{high}(data16)$	imed.+la reg.	- 3/10
7	LDA $addr$	LoaD Acc. direct	$(A) \leftarrow (addr)$	directă+impl.	- 4/13
8	STA $addr$	Store Acc. direct	$(addr) \leftarrow (A)$	impl.+dir.	- 4/13
9	LHLD $addr$	Load H and L Direct	$(L) \leftarrow (addr)$ $(H) \leftarrow (addr+1)$	directă+impl.	- 5/16
10	SHLD $addr$	Store H and L Direct	$(addr) \leftarrow (L)$ $(addr+1) \leftarrow (H)$	impl.+directă	- 5/16
11	LDAX rp	LoaD Acc. indirect	$(A) \leftarrow ((rp)), rp=B,D$	ind.+impl.	- 2/7
12	STAX rp	Store Acc. indirect	$((rp)) \leftarrow (A), rp=B,D$	ind.+impl.	- 2/7
13	XCHG	eXCHanGe H,L with D,E	$(H) \leftrightarrow (D), (L) \leftrightarrow (E)$	implicită	- 1/4

Instrucțiuni aritmetice						
14	ADD <i>r</i>	ADD reg. to acc.	$(A) \leftarrow (A) + (r)$	impl.+la reg.	toate	1/4
15	ADD M	ADD Mem. to acc.	$(A) \leftarrow (A) + ((H)(L))$	impl.+ind.	toate	2/7
16	ADI <i>data8</i>	ADd Immed. to acc.	$(A) \leftarrow (A) + data8$	impl.+imed.	toate	2/7
17	ADC <i>r</i>	ADd reg. to A with Cy	$(A) \leftarrow (A) + (r) + (CY)$	impl.+la reg.	toate	1/4
18	ADC M	ADd mem. to A with Cy	$(A) \leftarrow (A) + ((H)(L)) + (CY)$	impl.+ind.	toate	1/4
19	ACI <i>data8</i>	Add Immed. to A with CY	$(A) \leftarrow (A) + data8 + (CY)$	impl.+imed.	toate	2/7
20	SUB <i>r</i>	SUBtract register from A	$(A) \leftarrow (A) - (r)$	impl.+la reg.	toate	1/4
21	SUB M	SUBtract Memory from A	$(A) \leftarrow (A) - ((H)(L))$	impl.+ind.	toate	2/7
22	SUI <i>data8</i>	Subtract Immed. from A	$(A) \leftarrow (A) - data8$	impl.+imed.	toate	2/7
23	SBB <i>r</i>	SuB. reg. fr. A w. Borrow	$(A) \leftarrow (A) - (r) - (CY)$	impl.+la reg.	toate	1/4
24	SBB M	SuB. Mem. fr. A with Borr.	$(A) \leftarrow (A) - ((H)(L)) - (CY)$	impl.+ind.	toate	2/7
25	SBI <i>data8</i>	Sub. Imm. Fr. A with Borr.	$(A) \leftarrow (A) - data8 - (CY)$	impl.+imed.	toate	2/7
26	INR <i>r</i>	INcRement register	$(r) \leftarrow (r) + 1$	la registru	toate	1/4
27	INR M	INcRement Memory	$((H)(L)) \leftarrow ((H)(L)) + 1$	indirectă	toate	3/10
28	DCR <i>r</i>	DeCRement register	$(r) \leftarrow (r) - 1$	la registru	toate	1/4
29	DCR M	DeCRement Memory	$((H)(L)) \leftarrow ((H)(L)) - 1$	indirectă	toate	3/10
30	INX <i>rp</i>	INcrement register pair	$(rp) \leftarrow (rp) + 1,$ $rp = B, D, H, \text{ sau } SP$	la registru	-	1/6
31	DCX <i>rp</i>	DeCrement register pair	$(rp) \leftarrow (rp) - 1$	la registru	-	1/6
32	DAD <i>rp</i>	ADD <i>rp</i> to HL	$(H)(L) \leftarrow (H)(L) + (r_H)(r_L)$	impl.+la reg.	CY	3/10
33	DAA	Decimal Adjust Accumulator	$(A_3 \div A_0) > 9$ sau $(AC) = 1$ $\Rightarrow (A) \leftarrow (A) + 6;$ $(A_7 \div A_4) > 9$ sau $(CY) = 1$ $\Rightarrow (A) \leftarrow (A) + 6 \times 2^4$	implicită	toate	1/4

Instrucțiuni logice						
34	ANA <i>r</i>	ANd reg. with A	$(A) \leftarrow (A) \wedge (r)$	impl.+la reg.	Z,S,P CY=0 AC=0	1/4
35	ANA M	ANd Mem. with A	$(A) \leftarrow (A) \wedge ((H)(L))$	impl.+ind.		2/7
36	ANI <i>data8</i>	ANd Imm. with A	$(A) \leftarrow (A) \wedge data8$	impl.+imed.		2/7
37	XRA <i>r</i>	eXclusive oR reg. with A	$(A) \leftarrow (A) \oplus (r)$	impl.+la reg.		1/4
38	XRA M	eXcl. oR Mem. with A	$(A) \leftarrow (A) \oplus ((H)(L))$	impl.+ind.		2/7
39	XRI <i>data8</i>	eXcl. oR Immed. with A	$(A) \leftarrow (A) \oplus data8$	impl.+imed.		2/7
40	ORI <i>data8</i>	OR Immed. with A	$(A) \leftarrow (A) \vee data8$	impl.+imed.		2/7
41	ORA <i>r</i>	OR register with A	$(A) \leftarrow (A) \vee (r)$	impl.+la reg.		1/4
42	ORA M	OR Memory with A	$(A) \leftarrow (A) \vee ((H)(L))$	impl.+ind.		2/7
43	CMP <i>r</i>	CoMPare register with A	$(A) - (r)$	impl.+la reg.	toate	1/4
44	CMP M	CoMPare Memory with A	$(A) - ((H)(L))$	impl.+ind.	toate	2/7
45	CPI <i>data8</i>	ComPare Immed. with A	$(A) - data8$	impl.+imed.	toate	2/7
46	RLC	Rotate A Left with Carry	$(CY) \leftarrow (A_7) \rightarrow (A_0), (A_{n+1}) \leftarrow (A_n), n=0 \div 6$	implicită	CY	1/4
47	RRC	Rotate A Right with CY	$(CY) \leftarrow (A_0) \rightarrow (A_7), (A_{n+1}) \rightarrow (A_n), n=0 \div 6$	implicită	CY	1/4
48	RAL	Rotate A Left through CY	$(A_0) \leftarrow (CY) \leftarrow (A_7), (A_{n+1}) \leftarrow (A_n), n=0 \div 6$	implicită	CY	1/4
49	RAR	Rotate A Right through CY	$(A_0) \rightarrow (CY) \rightarrow (A_7), (A_{n+1}) \rightarrow (A_n), n=0 \div 6$	implicită	CY	1/4
50	CMA	CoMplement Acc.	$(A) \leftarrow \overline{(A)}$	implicită	-	1/4
51	STC	SeT Carry	$(CY) \leftarrow 1$	implicită	CY=1	1/4
52	CMC	CoMplement Carry	$(CY) \leftarrow \overline{(CY)}$	implicită	CY	1/4

Instrucțiuni de ramificare						
53	JMP <i>addr</i>	JuMP unconditional	$(PC) \leftarrow addr$	directă	-	3/10
54	Jcc <i>addr</i>	Jump on condition <i>cc</i>	Dacă <i>cc</i> : $(PC) \leftarrow addr$; Altfel: $(PC) \leftarrow (PC) + 3$	directă	-	3/10 1/5
55	CALL <i>addr</i>	CALL unconditional	$(SP) \leftarrow (SP) - 1$ $((SP)) \leftarrow (PC_H)$ $(SP) \leftarrow (SP) - 1$ $((SP)) \leftarrow (PC_L)$ $(PC) \leftarrow addr$	directă + indirectă + implicită	-	5/17
56	Ccc <i>addr</i>	Call on condition <i>cc</i>	Dacă <i>cc</i> : $(SP) \leftarrow (SP) - 1$ $((SP)) \leftarrow (PC_H)$ $(SP) \leftarrow (SP) - 1$ $((SP)) \leftarrow (PC_L)$ $(PC) \leftarrow addr$; Altfel: $(PC) \leftarrow (PC) + 3$	directă + indirectă + implicită	-	5/17 3/11
57	RET	RETurn	$(PC_L) \leftarrow ((SP))$ $(SP) \leftarrow (SP) + 1$ $(PC_H) \leftarrow ((SP) + 1)$ $(SP) \leftarrow (SP) + 1$	indirectă + implicită	-	3/10
58	Rcc	Return on condition <i>cc</i>	Dacă <i>cc</i> : $(PC_L) \leftarrow ((SP))$ $(SP) \leftarrow (SP) + 1$ $(PC_H) \leftarrow ((SP) + 1)$ $(SP) \leftarrow (SP) + 1$; Altfel: $(PC) \leftarrow (PC) + 1$	indirectă + implicită	-	3/11 1/5
59	RST <i>n</i>	ReSTart <i>n</i> = 0, 1, 2, ..., 7	$(SP) \leftarrow (SP) - 1$ $((SP)) \leftarrow (PC_H)$ $(SP) \leftarrow (SP) - 1$ $((SP)) \leftarrow (PC_L)$ $(PC) \leftarrow n \times 8$	directă + indirectă + implicită	-	3/11
60	PCHL	move HL to PC	$(PC_H) \leftarrow (H), (PC_L) \leftarrow (L)$	implicită	-	1/5

Instrucțiuni de lucru cu stiva, I/E și de comandă

61	PUSH <i>rp</i>	PUSH register pair on stack <i>rp</i> = B, D sau H	$(SP) \leftarrow (SP) - 1$ $((SP)) \leftarrow (r_H)$ $(SP) \leftarrow (SP) - 1$ $((SP)) \leftarrow (r_L)$	la registru + indirectă + implicită	-	3/12
62	PUSH PSW	PUSH accumulator and flags on stack; Flags: S Z 0 AC 0 P 0 CY	$(SP) \leftarrow (SP) - 1$ $((SP)) \leftarrow (A)$ $(SP) \leftarrow (SP) - 1$ $((SP)) \leftarrow (Flags)$	indirectă + implicită	-	3/12
63	POP <i>rp</i>	POP register pair off stack <i>rp</i> = B, D sau H	$(r_L) \leftarrow ((SP))$ $(SP) \leftarrow (SP) + 1$ $(r_H) \leftarrow ((SP))$ $(SP) \leftarrow (SP) + 1$	la registru + indirectă + implicită	-	3/10
64	POP PSW	POP accumulator and Flags off stack; Flags: S Z - AC - P - CY	$(Flags) \leftarrow ((SP))$ $(SP) \leftarrow (SP) + 1$ $(A) \leftarrow ((SP))$ $(SP) \leftarrow (SP) + 1$	indirectă + implicită	toate	3/10
65	XTHL	eXchange stack Top with HL	$(L) \leftrightarrow ((SP))$ $(H) \leftrightarrow ((SP)+1)$	implicită + indirectă	-	5/16
66	SPHL	move HL to SP	$(SP) \leftarrow (HL)$	la registru	-	1/6
67	IN <i>port</i>	Input	$(A) \leftarrow (port)$	impl.+dir.	-	3/10
68	OUT <i>port</i>	OUTput	$(port) \leftarrow (A)$	impl.+dir.	-	3/10
69	EI	Enable Interrupts	$(INTE) \leftarrow 1$	implicită	-	1/4
70	DI	Disable Interrupts	$(INTE) \leftarrow 0$	implicită	-	1/4
71	HLT	HaLT	PC - blocat	-	-	1/7
72	NOP	No OPeration	$(PC) \leftarrow (PC) + 1$	-	-	1/4
73	SIM	Set Interrupt Mask and write serial bit	$SOD \leftarrow (A_7)$, maschează RST 7.5, 6.5 și 5.5	implicită	-	1/4
74	RIM	Read serial bit and Interrupt Mask	$(A_7) \leftarrow SID$, citește măștile liniilor INTR, RST 7.5, 6.5 și 5.5	implicită	-	1/4