Setul de instructiuni al microprocesorului Intel 8086

1. Notatii utilizate

Operatii:

<-- = atribuire (valoarea din dreapta copiata in stinga)
<--> = permutare (interschimbare)

Operatori:

R = registru general
R8 = registru general de 8 biti: AL, AH, BL, BH, CL, CH, DL, DH
R16 = registru general de 16 biti: AX, BX, CX, DX, SP, BP, SI, DI
R_n = bitul *n* al registrului *R*
A = registru acumulator: AL, AH, AX
SR = registru segment: CS, DS, SS, ES

M = data din memorie
M8 = octet din memorie
M16 = cuvint din memorie (2 octeti)
M32 = pointer din memorie (4 octeti = 2 cuvinte)
M_n = bitul *n* al operandului din memorie *M*

EA(M) = adresa efectiva a locatiei de memorie *M*

(adr) = octetul aflat la adresa *adr* in memorie
(adr+1,adr) = cuvintul aflat la adresa *adr* in memorie

D = data de tip imediat (constanta numerica sau simbolica)
D8 = octet de tip imediat
D16 = cuvint de tip imediat

info = R, M sau D
MSb(info) = cel mai semnificativ bit al valorii *info*
LSb(info) = cel mai putin semnificativ bit al valorii *info*

etich = nume utilizat pentru referirea unei instructiuni (eticheta = adresa simbolica)

DEPL(etich) = deplasarea (diferenta adreselor) intre instructiunea curenta si *etich*

PORT(adr) = portul aflat la adresa *adr*

(flag) = flag este nedefinit

Mnemonica	Operatie (efect)	Indicatori afectati	Exemplu de instructiune

2. Instructiuni de transfer

2.1. Instructiuni de transfer generale

1. Atribuire (Move)

MOV R, R'	R <-- R'	-	MOV AX, BX
MOV R, M	R <-- M	-	MOV DX, VAR1
MOV M, R	M <-- R	-	MOV TAB [SI][DI], BX
MOV R, D	R <-- D	-	MOV AL, 20H
MOV M, D	M <-- D	-	MOV TAB [SI+2][DI], 1200H
MOV SR,R16	SR <-- R16 (SR cu exceptia lui CS)	-	MOV ES, AX
MOV SR,M16	SR <-- M16 (SR cu exceptia lui CS)	-	MOV DS, DATASEGBASE
MOV R16,SR	R16 <-- SR (SR cu exceptia lui CS)	-	MOV AX, SS
MOV M16,SR	M16 <-- SR (SR cu exceptia lui CS)	-	MOV DATASEGBASE, DS

2. Permutare (Exchange)

XCHG R, R'	R <--> R'	-	XCHG AX, DX
XCHG M	M <--> R	-	XCHG TAB [SI][DI+20], AX

3. Salvare in stiva (Push)

PUSH R16	SP <-- SP-2 (SP+1,SP) <-- R16	_ *notatie prescurtata:* / *STIVA <-- R16*	-	PUSH SI
PUSH SR	STIVA <-- SR		-	PUSH DS
PUSH M16	STIVA <-- M16		-	PUSH VAR2

4. Extragere din stiva (Pop)

POP R16	R16 <-- (SP+1,SP) SP <-- SP+2	_ *notatie prescurtata:* / *R16 <-- STIVA*	-	POP DX
POP SR	SR <-- STIVA		-	POP ES
POP M16	M16 <-- STIVA		-	POP VAR1

Mnemonica	Operatie (efect)	Indicatori afectati	Exemplu de instructiune

2.2. Instructiuni de transfer cu porturile

1. Citire din port

IN A, D8	A <-- PORT(D8)	-	IN AL, 0FAH
IN A, DX	A <-- PORT(DX)	-	IN AX, DX

2. Sriere in port

OUT D8, A	PORT(D8) <-- A	-	OUT 44, AX
OUT DX, A	PORT(DX) <-- A	-	OUT DX, AL

2.3. Instructiuni de transfer (calcul) de adrese

1. Calcul adresa efectiva (Load Effective Adress)

LEA R16, M	R16 <-- EA(M)	-	LEA BX, TAB [SI]

2. Calcul adresa fizice, folosind DS (Load Data Segment Register)

LDS R16, M	R16 <-- (M+1,M) DS <-- (M+3,M+2)	-	LDS DI, TAB [BX]

3. Calcul adresa fizice, folosind ES (Load Extra Segment Register)

LES R16, M	R16 <-- (M+1,M) ES <-- (M+3,M+2)	-	LES BX, TAB [SI]

2.4. Instructiuni de transfer de flag-uri

1. Transfer flag-uri in AH (Load AH from Flags)

LAHF	AH <-- SF,ZF,x,AF,x,PF,x,CF	-	LAHF

2. Transfer flag-uri din AH (Store AH in Flags)

SAHF	SF,ZF,x,AF,x,PF,x,CF <-- AH	*SF,ZF,AF,PF,CF*	SAHF

Mnemonica	Operatie (efect)	Indicatori afectati	Exemplu de instructiune
3. Salvare flag-uri in stiva (Push Flags)			
PUSHF	STIVA <-- F	-	PUSHF
4. Extragere flag-uri din stiva (Pop Flags)			
POP	F <-- STIVA	*Tot registrul F*	POPF

3. Instructiuni aritmetice

3.1. Instructiuni de adunare

1. Adunare simpla (Add)			
ADD R, R'	R <-- R + R'	*OF,SF,ZF,AF,PF,CF*	ADD CX, DX
ADD R, M	R <-- R + M	*OF,SF,ZF,AF,PF,CF*	ADD AL, TAB [SI+1]
ADD M, R	M <-- M + R	*OF,SF,ZF,AF,PF,CF*	ADD [BX], DX
ADD R, D	R <-- R + D	*OF,SF,ZF,AF,PF,CF*	ADD CL, 10H
ADD M, D	M <-- M + D	*OF,SF,ZF,AF,PF,CF*	ADD [SI], 1000H
2. Adunare cu transport (Add with Carry)			
ADC R, R'	R <-- R + R'+ CF	*OF,SF,ZF,AF,PF,CF*	ADC CL, DL
ADC R, M	R <-- R + M + CF	*OF,SF,ZF,AF,PF,CF*	ADC AX, TAB
ADC M, R	M <-- M + R + CF	*OF,SF,ZF,AF,PF,CF*	ADC [DI], DL
ADC R, D	R <-- R + D + CF	*OF,SF,ZF,AF,PF,CF*	ADC CX, 1000H
ADC M, D	M <-- M + D + CF	*OF,SF,ZF,AF,PF,CF*	ADC VAR2, 10H
3. Adunare cu 1 (Increment)			
INC R	R <-- R + 1	*OF,SF,ZF,AF,PF,CF*	INC CL
INC M	M <-- M + 1	*OF,SF,ZF,AF,PF,CF*	INC CNTR

Mnemonica	Operatie (efect)	Indicatori afectati	Exemplu de instructiune

3.2. Instructiuni de scadere

1. Scadere simpla (Substract)

SUB R, R'	R <-- R - R	*OF,SF,ZF,AF,PF,CF*	SUB AX, CX
SUB R, M	R <-- R - M	*OF,SF,ZF,AF,PF,CF*	SUB AX, TAB [BX]
SUB M, R	M <-- M - R	*OF,SF,ZF,AF,PF,CF*	SUB DS : TAB [BP], CL
SUB R, D	R <-- R - D	*OF,SF,ZF,AF,PF,CF*	SUB AH, 10H
SUB M, D	M <-- M - D	*OF,SF,ZF,AF,PF,CF*	SUB VAR1, 2

2. Scadere simpla cu imprumut (Substract with Carry)

SBB R, R'	R <-- R - R'- CF	*OF,SF,ZF,AF,PF,CF*	SBB BX, DX
SBB R, M	R <-- R - M - CF	*OF,SF,ZF,AF,PF,CF*	SBB BH, VAR1
SBB M, R	M <-- M - R - CF	*OF,SF,ZF,AF,PF,CF*	SBB VAR2, CX
SBB R, D	R <-- R - D - CF	*OF,SF,ZF,AF,PF,CF*	SBB CX, 1000H
SBB M, D	M <-- M - D - CF	*OF,SF,ZF,AF,PF,CF*	SBB TAB [SI+2], 4

3. Scadere cu 1 (Decrement)

DEC R	R <-- R + 1	*OF,SF,ZF,AF,PF,CF*	DEC CL
DEC M	M <-- M + 1	*OF,SF,ZF,AF,PF,CF*	DEC TAB [SI]

4. Scadere din 0 (Negate)

NEG R	R <-- 0 - R (R negat in C2)	*OF,SF,ZF,AF,PF,CF*	NEG AX
NEG M	M <-- 0 - M (M negat in C2)	*OF,SF,ZF,AF,PF,CF*	NEG TAB [BX+2]

5. Comparatie aritmetica prin scadere (Compare)

CMP R, R'	R - R' (afecteaza doar flagurile)	*OF,SF,ZF,AF,PF,CF*	CMP AH, CL
CMP R, M	R - M (afecteaza doar flagurile)	*OF,SF,ZF,AF,PF,CF*	CMP AX, VAR1
CMP M, R	M - R (afecteaza doar flagurile)	*OF,SF,ZF,AF,PF,CF*	CMP VAR2, DX
CMP R, D	R - D (afecteaza doar flagurile)	*OF,SF,ZF,AF,PF,CF*	CMP AX, 0
CMP M, D	M - D (afecteaza doar flagurile)	*OF,SF,ZF,AF,PF,CF*	CMP VAR1, 0

Mnemonica	Operatie (efect)	Indicatori afectati	Exemplu de instructiune

3.3. Instructiuni de inmultire

1. Inmultire intre numere fara semn (Multiply Acc. Register by Register or Memory; Unsigned)

MUL R8	AX <-- AL*R8 (produs)	*OF,CF,(SF,ZF,AF,PF)*	MUL BL
MUL R16	DX,AX <-- AX*R16 (produs)	*OF,CF,(SF,ZF,AF,PF)*	MUL CX
MUL M8	AX <-- AL*M8 (produs)	*OF,CF,(SF,ZF,AF,PF)*	MUL VAR1
MUL M16	DX,AX <-- AX*M16 (produs)	*OF,CF,(SF,ZF,AF,PF)*	MUL TAB [SI]

2. Inmultire intre numere cu semn (IntegerMultiply Acc. Register by Register or Memory)

IMUL R8	AX <-- AL*R8 (produs)	*OF,CF,(SF,ZF,AF,PF)*	IMUL DH
IMUL R16	DX,AX <-- AX*R16 (produs)	*OF,CF,(SF,ZF,AF,PF)*	IMUL BX
IMUL M8	AX <-- AL*M8 (produs)	*OF,CF,(SF,ZF,AF,PF)*	IMUL VAR2
IMUL M16	DX,AX <-- AX*M16 (produs)	*OF,CF,(SF,ZF,AF,PF)*	IMUL TAB [DI]

3.4. Instructiuni de impartire

1. Impartire intre numere fara semn (Division Unsigned)

DIV R8	AL <--AX/R8	(cit)		
	AH <-- AX mod R8	(rest)	*nedefiniti*	DIV DL
DIV R16	AX <--DX,AX/R16	(cit)		
	DX <-- DX,AX mod R8	(rest)	*nedefiniti*	DIV CX
DIV M8	AL <--AX/M8	(cit)		
	AH <-- AX mod R8	(rest)	*nedefiniti*	DIV VAR1
DIV M16	AX <--DX,AX/M16	(cit)		
	DX <-- DX,AX mod R8	(rest)	*nedefiniti*	DIV VAR2

2. Impartire intre numere cu semn (Integer Division)

IDIV R8	AL <--AX/R8	(cit)		
	AH <-- AX mod R8	(rest)	*nedefiniti*	IDIV DH
IDIV R16	AX <--DX,AX/R16	(cit)		
	DX <-- DX,AX mod R8	(rest)	*nedefiniti*	IDIV BX

Mnemonica	Operatie (efect)		Indicatori afectati	Exemplu de instructiune
IDIV M8	AL <--AX/M8	(cit)		
	AH <-- AX mod R8	(rest)	*nedefiniti*	IDIV TAB [SI]
IDIV M16	AX <--DX,AX/M16	(cit)		
	DX <-- DX,AX mod R8	(rest)	*nedefiniti*	IDIV [DI]

3.5. Instructiuni de conversie (extindere a semnului)

1. Conversie de la octet la cuvant (Convert Byte to Word)

CBW	daca $AL_7 = 0$ => AH <-- 0	-	CBW
	daca $AL_7 = 1$ => AH <-- 0FFH		

2. Conversie de la cuvant la dublu cuvant (Convert Word to Double Word)

CWD	daca $AX_{15} = 0$ => DX <-- 0	-	CWD
	daca $AX_{15} = 1$ => DX <-- 0FFH		

4. Instructiuni logice, deplasari si rotatii

4.1. Instructiuni logice

1. Negare logica (Logical Not)

NOT R	R <-- R\ (R negat in C1, bit cu bit)	-	NOT AX
NOT M	M <-- M\ (R negat in C1, bit cu bit)	-	NOT VAR1

2. "Si" logic (And)

AND R, R'	R <-- R AND R' (bit cu bit)	*CF=0,OF=0,(AF),SF,ZF,PF*	AND CX, DX
AND R, M	R <-- R AND M (bit cu bit)	*CF=0,OF=0,(AF),SF,ZF,PF*	AND BX, TAB [BX-2]
AND M, R	M <-- M AND R (bit cu bit)	*CF=0,OF=0,(AF),SF,ZF,PF*	AND VAR1, AL
AND R, D	R <-- R AND D (bit cu bit)	*CF=0,OF=0,(AF),SF,ZF,PF*	AND BX, 10H
AND M, D	M <-- M AND D (bit cu bit)	*CF=0,OF=0,(AF),SF,ZF,PF*	AND [BX], 10H

Mnemonica	Operatie (efect)	Indicatori afectati	Exemplu de instructiune
3. "Sau" logic (Or)			
OR R, R'	R <-- R OR R' (bit cu bit)	*CF=0,OF=0,(AF),SF,ZF,PF*	OR AL, CH
OR R, M	R <-- R OR M (bit cu bit)	*CF=0,OF=0,(AF),SF,ZF,PF*	OR CL, TAB
OR M, R	M <-- M OR R (bit cu bit)	*CF=0,OF=0,(AF),SF,ZF,PF*	OR VAR1, DH
OR R, D	R <-- R OR D (bit cu bit)	*CF=0,OF=0,(AF),SF,ZF,PF*	OR AX, 100H
OR M, D	M <-- M OR D (bit cu bit)	*CF=0,OF=0,(AF),SF,ZF,PF*	OR DX, 1000H
4. "Sau exclusiv" logic (Exclusive Or)			
XOR R, R'	R <-- R XOR R' (bit cu bit)	*CF=0,OF=0,(AF),SF,ZF,PF*	XOR BX, AX
XOR R, M	R <-- R XOR M (bit cu bit)	*CF=0,OF=0,(AF),SF,ZF,PF*	XOR BX, [SI]
XOR M, R	M <-- M XOR R (bit cu bit)	*CF=0,OF=0,(AF),SF,ZF,PF*	XOR [DI-4], CX
XOR R, D	R <-- R XOR D (bit cu bit)	*CF=0,OF=0,(AF),SF,ZF,PF*	XOR BL, 3
XOR M, D	M <-- M XOR D (bit cu bit)	*CF=0,OF=0,(AF),SF,ZF,PF*	XOR VAR2, 5
5. "Si" logic fara memorarea rezultatului (Test)			
TEST R, R'	R AND R' (afecteaza doar flagurile)	*CF=0,OF=0,(AF),SF,ZF,PF*	TEST CH, DL
TEST R, M	R AND M (afecteaza doar flagurile)	*CF=0,OF=0,(AF),SF,ZF,PF*	TEST AX, TAB [DI-2]
TEST M, R	M AND R (afecteaza doar flagurile)	*CF=0,OF=0,(AF),SF,ZF,PF*	TEST [DI], BX
TEST R, D	R AND D (afecteaza doar flagurile)	*CF=0,OF=0,(AF),SF,ZF,PF*	TEST DL, 10
TEST M, D	M AND D (afecteaza doar flagurile)	*CF=0,OF=0,(AF),SF,ZF,PF*	TEST DS : [BP], 1

4.2. Instructiuni de deplasare

1. Deplasare logica la stinga (Shift Logical Left)			
SHL R,1	R <-- R*2	*CF=MSb(R),OF,SF,ZF,(AF),PF*	SHL AL, 1
SHL M,1	M <-- M*2	*CF=MSb(M), OF,SF,ZF,(AF),PF*	SHL VAR1, 1
SHL R,CL	R <-- $R*2^{CL}$	$CF=R16_{16-CL}$ *sau* $R8_{16-CL}$*,OF,SF,ZF,(AF),PF*	SHL BX, CL
SHL M,CL	M <-- $M*2^{CL}$	$CF=M16_{16-CL}$ *sau* $M8_{16-CL}$*,OF,SF,ZF,(AF),PF*	SHL TAB [SI], CL

Mnemonica	Operatie (efect)	Indicatori afectati	Exemplu de instructiune
2. Deplasare aritmetica la stinga (Shift Arithmetic Left)			
SAL R,1	R <-- R*2	*CF=MSb(R), OF,SF,ZF,(AF),PF*	SAL AX, 1
SAL M,1	M <-- M*2	*CF=MSb(M),OF,SF,ZF,(AF),PF*	SAL VAR2, 1
SAL R,CL	R <-- R*2^{CL}	*CF=R16$_{16-CL}$ sau R8$_{16-CL}$,OF,SF,ZF,(AF),PF*	SAL DX, CL
SAL M,CL	M <-- M*2^{CL}	*CF=M16$_{16-CL}$ sau M8$_{16-CL}$,OF,SF,ZF,(AF),PF*	SAL [SI], CL
3. Deplasare logica la dreapta (Shift Logical Right)			
SHR R,1	R <-- R/2	*CF=LSb(R),OF,SF,ZF,(AF),PF*	SHR BX, 1
SHR M,1	M <-- M/2	*CF=LSb(M),OF,SF,ZF,(AF),PF*	SHR [BX], 1
SHR R,CL	R <-- R/2^{CL}	*CF=R$_{CL-1}$,OF,SF,ZF,(AF),PF*	SHR AH, CL
SHR M,CL	M <-- M/2^{CL}	*CF=M$_{CL-1}$,OF,SF,ZF,(AF),PF*	SHR DS : [BP], CL
4. Deplasare aritmetica la dreapta, cu pastrarea semnului (Shift Arithmetic Right)			
SAR R,1	R <-- R/2	*CF=LSb(R),OF,SF,ZF,(AF),PF*	SAR CX, 1
SAR M,1	M <-- M/2	*CF=LSb(M),OF,SF,ZF,(AF),PF*	SAR [SI], 1
SAR R,CL	R <-- R/2^{CL}	*CF=R$_{CL-1}$,OF,SF,ZF,(AF),PF*	SAR AL, CL
SAR M,CL	M <-- M/2^{CL}	*CF=M$_{CL-1}$,OF,SF,ZF,(AF),PF*	SAR VAR2, CL

4.3. Instructiuni de rotatie

1. Rotatie la stinga (Rotate Left)			
ROL R,1	R <-- R rotit cu 1 poz. la stinga	*CF= MSb(R),OF*	ROL DL, 1
ROL M,1	M <-- M rotit cu 1 poz. la stinga	*CF= MSb(M),OF*	ROL [DI+2], 1
ROL R,CL	R <-- R rotit cu CL poz. la stinga	*CF= R16$_{16-CL}$ sau R8$_{16-CL}$,OF*	ROL DH, CL
ROL M,CL	M <-- M rotit cu CL poz. la stinga	*CF= M16$_{16-CL}$ sau M8$_{16-CL}$,OF*	ROL VAR1, CL
2. Rotatie la dreapta (Rotate Right)			
ROR R,1	R <-- R rotit cu 1 poz. la dreapta	*CF= LSb(R),OF*	ROR BX, 1
ROR M,1	M <-- M rotit cu 1 poz. la dreapta	*CF= LSb(M),OF*	ROR TAB [BX], 1
ROR R,CL	R <-- R rotit cu CL poz. la dreapta	*CF= R$_{CL-1}$,OF*	ROR AL, CL
ROR M,CL	M <-- M rotit cu CL poz. la dreapta	*CF= M$_{CL-1}$,OF*	ROR [SI] [DI], CL

Mnemonica	Operatie (efect)	Indicatori afectati	Exemplu de instructiune
3. Rotatie la stinga prin bitul CF (Rotate Left with Carry)			
RCL R,1	R <-- perechea R,CF rotita cu 1 poz. la stinga	*CF= MSb(R),OF*	RCL CX, 1
RCL M,1	M <-- perechea M,CF rotita cu 1 poz. la stinga	*CF=* M_{15}, *OF*	RCL TAB [SI], 1
RCL R,CL	R <-- perechea R,CF rotita cu CL poz. la stinga	*CF=* R_{16-CL}, *OF*	RCL BL, CL
RCL M,CL	M <-- perechea M,CF rotita cu CL poz. la stinga	*CF=* M_{16-CL}, *OF*	RCL [SI] [BX], CL
4. Rotatie la dreapta prin bitul CF (Rotate Right with Carry)			
RCR R,1	R <-- R rotit cu 1 poz. la dreapta	*CF=* R_{15}, *OF*	RCR CL, 1
RCR M,1	M <-- M rotit cu 1 poz. la dreapta	*CF=* M_{15}, *OF*	RCR VAR1, 1
RCR R,CL	R <-- R rotit cu CL poz. la dreapta	*CF=* R_{16-CL},*OF*	RCR AX, CL
RCR M,CL	M <-- M rotit cu CL poz. la dreapta	*CF=* M_{16-CL},*OF*	RCR VAR2, CL

5. Instructiuni de lucru cu siruri de date

5.1. Instructiuni de transfer

1. Transfer intre doua siruri cu operanzi expliciti (Move String)

MOVS M8,M8' *initial DI = EA(M8), SI = EA(M8')* - MOVS SIRB1, SIRB2

- daca DF=0: (DI) <-- (SI)
DI <-- DI+1 *= actualizare DI cu 1*
SI <-- SI+1 *= actualizare SI cu 1*

- daca DF=1: (DI) <-- (SI)
DI <-- DI-1 *= actualizare DI cu -1*
SI <-- SI-1 *= actualizare SI cu -1*

MOVS M16,M16' *initial DI = EA(M16), SI = EA(M16')* - MOVS SIRW1, SIRW2

- daca DF=0: (DI+1,DI) <-- (SI+1,SI)
DI <-- DI+2 *= actualizare DI cu 2*
SI <-- SI+2 *= actualizare SI cu 2*

Mnemonica	Operatie (efect)	Indicatori afectati	Exemplu de instructiune
	- daca DF=1: (DI+1,DI) <-- (SI+1,SI) DI <-- DI-2 = *actualizare DI cu -2* SI <-- SI-2 = *actualizare SI cu -2*		

2. Incarcare acumulator dintr-un sir cu operanzi expliciti (Load String)

LODS AL,M8	*initial SI = EA(M8)* *- daca DF=0:* AL <-- (SI) *actualizare SI cu 1* *- daca DF=1:* AL <-- (SI) *actualizare SI cu -1*	-	LODS AL, SIRB
LODS AX,M16	*initial SI = EA(M16)* *- daca DF=0:* AX <-- (SI+1,SI) *actualizare SI cu 2* *- daca DF=1:* AX <-- (SI+1,SI) *actualizare SI cu -2*	-	LODS AX, SIRW

3. Stocare intr-un sir din acumulator cu operanzi expliciti (Store String)

STOS M8,AL	*initial DI = EA(M8)* *- daca DF=0:* (DI) <-- AL *actualizare DI cu 1* *- daca DF=1:* (DI) <-- AL *actualizare DI cu -1*	-	STOS SIRB, AL
STOS M16,AX'	*initial DI = EA(M16)* *- daca DF=0:* (DI+1,DI) <-- AX *actualizare DI cu 2* *- daca DF=1:* (DI+1,DI) <-- AX *actualizare DI cu -2*	-	STOS SIRW, AX

Mnemonica	Operatie (efect)	Indicatori afectati	Exemplu de instructiune

4. Transfer intre doua siruri cu operanzi impliciti (Move String)

dupa initializare cu DI = EA(M8), SI = EA(M8'):

MOVSB	*- daca DF=0:* (ES:DI) <-- (DS:SI) DI <-- DI+1 *= actualizare DI cu 1* SI <-- SI+1 *= actualizare SI cu 1*	-	MOVSB
	- daca DF=1: (ES:DI) <-- (DS:SI) DI <-- DI-1 *= actualizare DI cu -1* SI <-- SI-1 *= actualizare SI cu -1*		

dupa initializare cu DI = EA(M16), SI = EA(M16'):

MOVSW	*- daca DF=0:* (ES:DI+1,DI) <-- (DS:SI+1,SI) DI <-- DI+2 *= actualizare DI cu 2* SI <-- SI+2 *= actualizare SI cu 2*	-	MOVSW
	- daca DF=1: (ES:DI+1,DI) <-- (DS:SI+1,SI) DI <-- DI-2 *= actualizare DI cu -2* SI <-- SI-2 *= actualizare SI cu -2*		

5. Incarcare acumulator dintr-un sir cu operanzi impliciti (Load String)
dupa initializare cu SI = EA(M8'):

LODSB	*- daca DF=0:* AL <-- (DS:SI) SI <-- SI+1 *= actualizare SI cu 1*	-	LODSB
	- daca DF=1: AL <-- (DS:SI) SI <-- SI-1 *= actualizare SI cu -1*		

dupa initializare cu SI = EA(M16'):

LODSW	*- daca DF=0:* AX <-- (DS:SI+1,SI) SI <-- SI+2 *= actualizare SI cu 2*	-	LODSW
	- daca DF=1: AX <-- (DS:SI+1,SI) SI <-- SI-2 *= actualizare SI cu -2*		

Mnemonica	Operatie (efect)	Indicatori afectati	Exemplu de instructiune

6. Stocare intr-un sir din acumulator cu operanzi impliciti (Store String)

dupa initializare cu DI = EA(M8):

STOSB	*- daca DF=0:*	(ES:DI) <-- AL	-	STOSB
		DI <-- DI+1 *= actualizare DI cu 1*		
	- daca DF=1:	(ES:DI) <-- AL		
		DI <-- DI-1 *= actualizare DI cu -1*		

dupa initializare cu DI = EA(M16):

STOSW	*- daca DF=0:*	(ES:DI+1,DI) <-- AX	-	STOSW
		DI <-- DI+2 *= actualizare DI cu 2*		
	- daca DF=1:	(ES:DI+1,DI) <-- AX		
		DI <-- DI-2 *= actualizare DI cu -2*		

7. Prefixul de repetare

REP *transfsir*	*transfsir*	REP MOVSB
	CX <-- CX-1	REP MOVSW
	- daca CX#0 se reia *transfsir*	REP STOSB
	- daca CX=0 se trece la instructiunea urmatoare	

5.2. Instructiuni de comparatie

1. Comparatie intre doua siruri cu operanzi expliciti (Compare String)

CMPS M8,M8'	*initial DI = EA(M8), SI = EA(M8')*		*CF,OF,SF,ZF,AF,PF*	CMPS SIRB1, SIRB2
	- daca DF=0:	(DI) - (SI)		
		DI <-- DI+1 *= actualizare DI cu 1*		
		SI <-- SI+1 *= actualizare SI cu 1*		
	- daca DF=1:	(DI) - (SI)		
		DI <-- DI-1 *= actualizare DI cu -1*		
		SI <-- SI-1 *= actualizare SI cu -1*		
Mnemonica	Operatie (efect)		Indicatori afectati	Exemplu de instructiune
CMPS M16,M16'	*initial DI = EA(M16), SI = EA(M16')*		*CF,OF,SF,ZF,AF,PF*	CMPS SIRW1, SIRW2

- daca DF=0: (DI+1,DI) - (SI+1,SI)
DI <-- DI+2 = *actualizare DI cu 2*
SI <-- SI+2 = *actualizare SI cu 2*
- daca DF=1: (DI+1,DI) - (SI+1,SI)
DI <-- DI-2 = *actualizare DI cu -2*
SI <-- SI-2 = *actualizare SI cu -2*

2. Comparatie intre un sir si acumulator cu operanzi expliciti (Scan String)

SCAS AL,M8	*initial SI = EA(M8)*	*CF,OF,SF,ZF,AF,PF*	SCAS AL, SIRB
	- daca DF=0: AL - (SI) *actualizare SI cu 1*		
	- daca DF=1: AL - (SI) *actualizare SI cu -1*		
SCAS AX,M16	*initial SI = EA(M16)*	*CF,OF,SF,ZF,AF,PF*	SCAS AX, SIRW
	- daca DF=0: AX - (SI+1,SI) *actualizare SI cu 2*		
	- daca DF=1: AX - (SI+1,SI) *actualizare SI cu -2*		

3. Comparatie intre doua siruri cu operanzi impliciti (Compare String)

dupa initializare cu DI = EA(M8), SI = EA(M8'):

CMPSB	*- daca DF=0:* (ES:DI) - (DS:SI) DI <-- DI+1 = *actualizare DI cu 1* SI <-- SI+1 = *actualizare SI cu 1*	*CF,OF,SF,ZF,AF,PF*	CMPSB
	- daca DF=1: (ES:DI) - (DS:SI) DI <-- DI-1 = *actualizare DI cu -1* SI <-- SI-1 = *actualizare SI cu -1*		

Mnemonica	Operatie (efect)	Indicatori afectati	Exemplu de instructiune

dupa initializare cu DI = EA(M16), SI = EA(M16'):

CMPSW	*- daca DF=0:*	(ES:DI+1,DI) - (DS:SI+1,SI)	*CF,OF,SF,ZF,AF,PF*	CMPSW
		DI <-- DI+2 *= actualizare DI cu 2*		
		SI <-- SI+2 *= actualizare SI cu 2*		
	- daca DF=1:	(ES:DI+1,DI) - (DS:SI+1,SI)		
		DI <-- DI-2 *= actualizare DI cu -2*		
		SI <-- SI-2 *= actualizare SI cu -2*		

4. Comparatie intre un sir si acumulator cu operanzi expliciti (Scan String)
dupa initializare cu SI = EA(M8'):

SCASB	*- daca DF=0:*	AL - (DS:SI)	*CF,OF,SF,ZF,AF,PF*	SCASB
		SI <-- SI+1 *= actualizare SI cu 1*		
	- daca DF=1:	AL - (DS:SI)		
		SI <-- SI-1 *= actualizare SI cu -1*		

dupa initializare cu SI = EA(M16'):

SCASW	*- daca DF=0:*	AX - (DS:SI+1,SI)	*CF,OF,SF,ZF,AF,PF*	SCASW
		SI <-- SI+2 *= actualizare SI cu 2*		
	- daca DF=1:	AX - (DS:SI+1,SI)		
		SI <-- SI-2 *= actualizare SI cu -2*		

5. Prefixul de repetare conditionata

REPE *compsir*	*compsir*		REPE CMPSW
	CX <-- CX-1		
	- daca (CX#0 si ZF=1) se reia *compsir*		
	- daca (CX=0 sau ZF=0) se trece la instructiunea urmatoare		
REPZ *compsir*	*compsir*		REPZ SCASB
	CX <-- CX-1		
	- daca (CX#0 si ZF=1) se reia *compsir*		
	- daca (CX=0 sau ZF=0) se trece la instructiunea urmatoare		
Mnemonica	Operatie (efect)	Indicatori afectati	Exemplu de instructiune
REPNE *compsir*	*compsir*		REPNE CMPSB
	CX <-- CX-1		
	- daca (CX#0 si ZF=0) se reia *compsir*		

	- daca (CX=0 sau ZF=1) se trece la instructiunea urmatoare	
REPNZ *compsir*	*compsir*	REPNZ SCASW
	CX <-- CX-1	
	- daca (CX#0 si ZF=1) se reia *compsir*	
	- daca (CX=0 sau ZF=0) se trece la instructiunea urmatoare	

6. Instructiuni de transfer al comenzii

6.1. Instructiuni de salt neconditionat

1. Salt neconditionat intrasegment direct (Jump)

JMP etich	IP <-- IP + DEPL(etich)	*= salt la etich*	-	JMP ET1

2. Salt neconditionat intrasegment indirect (Jump)

JMP R16	IP <-- R16	*= salt la (R16)*	-	JMP AX
JMP M16	IP <-- M16	*= salt la (M16)*	-	JMP OFFS

3. Salt neconditionat intersegment direct (Jump)

JMP etich	IP <-- OFFSET(etich)		-	JMP ET2
	CS <-- SEG(etich)	*= salt la etich in alt segment*		

4. Salt neconditionat intersegment indirect (Jump)

JMP M32	IP <-- (M+1,M)		-	JMP ADRLOG
	CS <-- (M+3,M+2)	*= salt la ((M+3,M+2) : (M+1,M))*		

Mnemonica	Operatie (efect)	Indicatori afectati	Exemplu de instructiune

6.2. Instructiuni de apel de subprogram

1. Apel de subprogram intrasegment direct (Call)

CALL etich	STIVA <-- IP IP <-- IP + DEPL(etich)	-	CALL ET3

2. Apel de subprogram intrasegment indirect (Call)

CALL R16	STIVA <-- IP IP <-- R16	-	CALL CX
CALL M16	STIVA <-- IP IP <-- M16	-	CALL OFF

3. Apel de subprogram intersegment direct (Call)

CALL etich FAR	STIVA <-- CS STIVA <-- IP IP <-- OFFSET(etich) CS <-- SEG(etich)	-	CALL ET FAR

4. Apel de subprogram intersegment indirect (Call)

CALL M32 FAR	STIVA <-- CS STIVA <-- IP IP <-- (M+1,M) CS <-- (M+3,M+2)	-	CALL ADRLOGICA

6.3. Instructiuni de revenire din subprogram

1. Revenire din subprogram intrasegment fara POP (Return)

RET	IP <-- STIVA	-	RET

2. Revenire din subprogram intrasegment cu POP (Return)

RET D16	IP <-- STIVA SP <-- SP + D16	-	RET 4

Mnemonica	Operatie (efect)	Indicatori afectati	Exemplu de instructiune

3. Revenire din subprogram intersegment fara POP (Return)

RET	IP <-- STIVA CS <-- STIVA	-	RET

4. Revenire din subprogram intersegment cu POP (Return)

RET D16	IP <-- STIVA CS <-- STIVA SP <-- SP + 2 + D16	-	RET 8

6.4. Instructiuni de salt conditionat

1. Salt conditionat direct apropiat (Jump)

J*conditie* etich - daca *conditie* este indeplinita atunci: - JZ REZNUL

IP <-- IP + DEPL(etich) = *salt la etich*

- altfel trece la instructiunea urmatoare

Mnemonica	Conditie de salt	Interpretare conditie
1.a. Pentru orice tip de valori:		
JC etich	(CF=1)	rezultat ALU cu transport
JNC etich	(CF=0)	rezultat ALU fara transport
JE etich	(ZF=1)	rezultat ALU nul
JZ etich	(ZF=1)	rezultat ALU nul
JNE etich	(ZF=0)	rezultat ALU nenul
JNZ etich	(ZF=0)	rezultat ALU nenul
JP etich	(PF=1)	rezultat ALU cu numar par de 1
JPE etich	(PF=1)	rezultat ALU cu numar par de 1
JNP etich	(PF=0)	rezultat ALU cu numar impar de 1
JPO etich	(PF=0)	rezultat ALU cu numar impar de 1
JCXZ etich	(CX=0)	contor nul
Mnemonica	Conditie de salt	Interpretare conditie

1.b. Pentru valori fara semn:

JA etich	(CF=0) si (ZF=0)	rezultat ALU > 0
JNBE etich	(CF=0) si (ZF=0)	rezultat ALU > 0
JAE etich	(CF=0)	rezultat ALU >= 0

JNB	etich	(CF=0)	rezultat ALU >= 0
JB	etich	(CF=1)	rezultat ALU < 0
JNAE	etich	(CF=1)	rezultat ALU < 0
JBE	etich	(CF=1) sau (ZF=1)	rezultat ALU <= 0
JNA	etich	(CF=1) sau (ZF=1)	rezultat ALU <= 0

1.c. Pentru valori cu semn:

JS	etich	(SF=0)	rezultat ALU negativ
JNS	etich	(SF=1)	rezultat ALU pozitiv
JO	etich	(OF=1)	rezultat ALU cu depasire de gama
JNO	etich	(OF=0)	rezultat ALU fara depasire de gama
JG	etich	(SF=OF) si (ZF=0)	rezultat ALU > 0
JNLE	etich	(SF=OF) si (ZF=0)	rezultat ALU > 0
JGE	etich	(SF=OF)	rezultat ALU >= 0
JNL	etich	(SF=OF)	rezultat ALU >= 0
JL	etich	(SF<>OF)	rezultat ALU < 0
JNGE	etich	(SF<>OF)	rezultat ALU < 0
JLE	etich	(SF<>OF) sau (ZF=1)	rezultat ALU > 0
JNG	etich	(SF<>OF) sau (ZF=1)	rezultat ALU > 0

Mnemonica	Operatie (efect)	Indicatori afectati	Exemplu de instructiune

6.5. Instructiuni iterative

1. Ciclu cu test final (Loop)

LOOP	etich	- daca *CX#0* atunci *salt la etich* - altfel *(CX=0)* trece la instructiunea urmatoare	-	LOOP RELUARE

2. Ciclu cu test final cu dubla conditie (Loop)

LOOPE etich	- daca (*CX#0 si ZF=1)* atunci *salt la etich* - altfel *(CX=0 sau ZF=0)* trece la instructiunea urmatoare	-	LOOPE OPNOU
LOOPZ etich	- daca (*CX#0 si ZF=1)* atunci *salt la etich* - altfel *(CX=0 sau ZF=0)* trece la instructiunea urmatoare	-	LOOPZ NOUTEST
LOOPNE etich	- daca (*CX#0 si ZF=1)* atunci *salt la etich* - altfel *(CX=0 sau ZF=1)* trece la instructiunea urmatoare	-	LOOPNE REV
LOOPNZ etich	- daca (*CX#0 si ZF=1)* atunci *salt la etich* - altfel *(CX=0 sau ZF=1)* trece la instructiunea urmatoare	-	LOOPNZ BUCLA1

6.6. Instructiuni de control al intreruperilor

1. Apel intrerupere software (Interupt)

INT *tip* (= D8)	STIVA <-- F	= salvare flag-uri	*IF, TF*	INT 21H
	IF <-- 0 , TF <-- 0	= invalidare intreruperi		
	STIVA <-- CS	= salvare segment adresa		
	STIVA <-- IP	= salvare offset adresa		
	CS <-- (4**tip*+3, 4**tip*+2)	= incarcare segment adresa subprogram		
	IP <-- (4**tip*+1, 4**tip*)	= incarcare offset adresa subprogram		

2. Revenire in programul intrerupt (Return from Interupt)

IRET	IP <-- STIVA	= restabilire offset adresa	*Tot registrul F*	IRET
	CS <-- STIVA	= restabilire segment adresa		
	F <-- STIVA	= restabilire flag-uri		
Mnemonica	Operatie (efect)		Indicatori afectati	Exemplu de instructiune

7. Instructiuni de control al procesorului

7.1. Operatii asupra flag-urilor

1. Fortare CF = 1 (Set Carry)

STC	CF <-- 1	= transport = 1	*CF*	STC

2. Fortare CF = 0 (Clear Carry)

CLC	CF <-- 0	= transport = 0	*CF*	CLC

3. Complementare CF (Complement Carry)

CMC	CF <-- CF\	= complementare transport	*CF*	CMC

4. Fortare DF = 1 (Set Direction)

STD	DF <-- 1	= directie inapoi la parcurgerea sirurilor	*DF*	STD

5. Fortare DF = 0 (Clear Direction)

CLD	DF <-- 0	= directie inainte la parcurgerea sirurilor	*DF*	CLD

6. Fortare IF = 1 (Set Interupt)

STI	IF <-- 1	= validare intreruperi	*IF*	STI

7. Fortare IF = 0 (Clear Interupt)

CLI	IF <-- 0	= invalidare intreruperi	*IF*	CLI

7.2. Sincronizare externa

1. Oprire (Halt)

HLT	UCP intra in starea HALT (din care iese prin: NMI sau RESET sau INTR si IF)		HLT

2. Stare de asteptare (Wait)

WAIT	asteapta pana intrarea TEST=0	-	WAIT

7.3. Nici o operatie (no operation)

NOP	nici o operatie	-	NOP