נספח

פורמט פקודות מכונה 8086/8

ב-8086/8 גודל יצוג של פקודה היה מבית אחד עד ששה בתים.

פקודות מכונה בגודל בית

בדרך כלל פקודות על אופרנד אוגר 16 ביט אחד או ללא אופרנדים.

יצוג פקודות ללא אופרנדים הם פשוט מספר פקודה. הפקודות הללו הן:

XLAT, LAHF, SAHF, PUSHF, POPF, AAA, DAA, CLC, CMC, STC, CLD, STD, CLI, STI, HLT, IRET, WAIT, RET (ללא אופרנד), INTO, INT3

חלק מהפקודות הפועלות על אופרנד אוגר 16 ביט כללי כמו PUSH, POP , INC ממומשות בפורמט

7	6	5	4	3	2	1	0
O P C O D E					R E G		

כאשר REG הוא קידוד של האוגרים הכלליים מוגדר ע"י

REG	
000	AX
001	CX
010	DX
011	BX
100	SP
101	BP
110	SI
111	DI

הפקודות PUSH sr ו-POP sr כאשר sr הוא אחד מאוגרי הסגמנטים הם מהצורה

7	6	5	4	3	2	1	0
0	0	0	S	R	1	1	1

כאשר SR הוא קידוד של אוגרי הסגמנט, שישמש אותנו גם בהמשך,

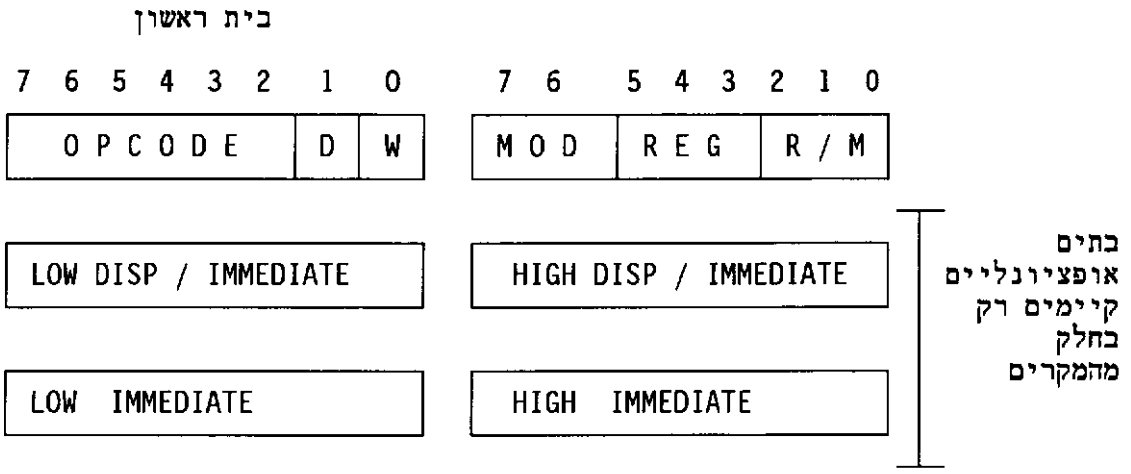
ES	0 0
CS	0 1
SS	1 0
DS	1 1

הפקודות IN AL,DX, IN AX,DX, OUT DX,AL, OUT DX,AX ממשמשות בפורמט:

7 6 5 4 3 2 1	0
Opcode7	W

כאשר W == 0 כאשר מדובר בפעולת ביט (AL)
W == 1 כאשר מדובר בפעולת מילה (AX)

הפורמט הבא מתאר את מרבית פקודות המכונה של ה-8086/8:



המפתח לפענוח פקודות הארוכות יותר הם שני הבתים הראשונים:

6 הביטים הראשונים מאפינים את סוג הפקודה (... ADD, MOV)

ביט D (פקודות מכונה 2 אופרנדים בלבד):
אם D==1 שדה ה-REG בבית השני מצין את אופרנד היעד
אם D==0 שדה ה-REG בבית השני מצין את אופרנד מקור

במקרה של פעולות אוגר כללי - אוגר כללי D==1

ביט W: W==0 מדובר הפעולת בית
W==1 מדובר הפעולת מילה

הערות:

- בפקודות אריתמטיות עם קבועים (למשל CMP AX,-6, ADD CX,121) ביט ה-D נקרא S והוא מצין אם להרחיב קבוע 8 ביט ל-16 ביט תוך התחשבות בסימן. S==0 אין התחשבות בסימן. S==1 הרחבה תוך התחשבות בסימן אם W==1.

- בפקודות הזזה הביטיות ביט ה-D נקרא V והוא משמש להבדיל בין גירסאות ההזזה הבודדת של הפקודות הללו לבין הגירסאות המצינות את ההזזה ב-CL, למשל להבחין בין הפקודה SHL AX,1 לבין SHL AX,CL. V==0 הזזה בודדת. V==1 הזזה לפי CL.

- בפקודות LOOPE, LOOPNE ובקידומות REPE, REPNE ביט ה-W נקרא Z והוא משמש לאבחנה של התנאי הנוסף מעבר ל- CX > 0 להסתעפות:

Z == 0 <==> LOOPNE או REPNE

Z == 1 <==> LOOPE או REPE

<u>בית 2 בפקודה</u>

MOD - 2 ביטים המבדילים בין מיעון זכרון ואוגרים ובמקרה של זכרון מצינים כמה בתים של הזזה (Displacement) באים לאחר בית זה.

REG (פקודות 2 אופרנדים בלבד) - 3 ביטים המצינים שדה אוגר.

R/M - 3 ביטים שיחד עם ה-MOD קובעים את שיטת המיעון

MOD ==	00 אין Displacement	
	01 Displacement של בית אחד	זכרון
	10 Displacement של שני בתים	
	11 פעולת אוגר- אוגר	

REG - יחד עם ביט ה-W בוחר את האוגר:

REG	W==0	W==1
000	AL	AX
001	CL	CX
010	DL	DX
011	BL	BX
100	AH	SP
101	CH	BP
110	DH	SI
111	BH	DI

בפקודות MOV שמעורבות בהם אוגרי הסגמנטים, השימוש ב-REG יהיה לפי הפורמט:

7 6	5	4 3	2 1 0
M O D	0	S R	R / M

כאשר SR הוא הקידוד של אוגרי הסגמנטים שתואר קודם.

הוראות שבהם יש אופרנד אחד (למשל INC AL, INC BL, NOT AX, INC WORD PTR [BX]) או אופרנד מתפרש אחד (MUL, DIV) או פקודות 2 אופרנדים שאחד מהם קבוע (למעט כאשר AX או AL הם האופרנד השני) משתמשים בביטים בשדה ה-REG כהשלמה ל-opcode. כלומר יש לו משמעויות אחרות.

R/M - מצין את המיעון יחד עם ה-MOD כמתואר להלן:

				MOD 11	
R/M	MOD==00	MOD==01	MOD==10	W==0	W==1
000	BX+SI	BX+SI+DISP8	BX+SI+DISP16	AL	AX
001	BX+DI	BX+DI+DISP8	BX+DI+DISP16	CL	CX
010	BP+SI	BP+SI+DISP8	BP+SI+DISP16	DL	DX
011	BP+DI	BP+DI+DISP8	BP+DI+DISP16	BL	BX
100	SI	SI+DISP8	SI+DISP16	AH	SP
101	DI	DI+DISP8	DI+DISP16	CH	BP
110	Direct Address	BP+DISP8	BP+DISP16	DH	SI
111	BX	BP+DISP8	BP+DISP16	BH	DI

Direct Address - הכתובת נלקחת ישר משני הבתים אחרי בית שיטת המיון.

התיחסות טהורה ל-BP (למשל MOV [BP],AX) ממומשת כהתיחסות דרך [BP+0] כלומר displacement 0 בגודל בית. ממילא השימוש שהאוגר BP נועד לו מכתיב שכמעט ואין צורך בגרסה הזו.

בתים מקדימים Prefix bytes

לחלק מהפקודות (מכל הפורמטים) ישנם אופציות של בתים מקדימים prefix bytes משני סוגים: Instruction Prefix ו- Segment Override. הדבר יכול להגדיל יצוג של פקודת מכונה בעד 2 בתים. לפיכך הגודל המירבי של פקודת מכונה ב-8086 הוא 8 בתים.

האפשריות של Instruction Prefix הן

```
F3h = 1111 0011b   REP prefix (פקודות מחרוזת בלבד)
F3h = 1111 0011b   REPE / REPZ prefix
                              (פקודות מחרוזת בלבד )
F2h = 1111 0010b   REPNE / REPNZ prefix
                              (פקודות מחרוזת בלבד )
F0h = 1111 0010b   LOCK prefix
```

האפשריות של Segment Override הן

```
26h - 0010 0110b - ES segment override prefix
2Eh - 0010 1110b - CS segment override prefix
36h - 0011 0110b - SS segment override prefix
3Eh - 0011 1110b - DS segment override prefix
```

מיעון בשיטת Segment Override

כאשר בפקודת היתיחסות לזכרון יש Segment Override (למעט DS) מופיע לפני הפקודה בית מיוחד המצין ל-CPU לבחור אוגר סגמנט אחר מברירת המחדל (DS למעט כאשר BP מופיע אז נבחר SS). לאחר מכן תופיע הפקודה כמו קודם.

הערכים של ה-Segment Override נבדלים ב-2 הביטים האמצעיים:

```
                    SR
                   ----
ES:  26h = 0 0 1 0  0 1 1 0        (0 0)
CS:  2Eh = 0 0 1 0  1 1 1 0        (0 1)
SS:  36h = 0 0 1 1  0 1 1 0        (1 0)
DS:  3Eh = 0 0 1 1  1 1 1 0        (1 1)
                   ----
```

פקודות נוספת שאינן לפי הפורמט הנ"ל

פקודות 2 אופרנדים בין קבוע כאפרנד מקור והאוגר AX או AL כאופרנד יעד (למשל MOV AL,9 ADD AX,7 אבל גם פקודות קלט/פלט עם קבוע למשל OUT 60h,AL) הם מהפורמט

7 6 5 4 3 2 1	0	
Opcode7	W	Const

כאשר W משמש להבדיל בין פעולת בית למילה, כמו קודם.
Const יכול להיות בית או מילה, למעט פקודות קלט/פלט (בית בלבד).

פקודות ההסתעפות (עם אופרנד JMP, CALL, JE, INT, RET), למעט הסתעפויות עקיפות, הם לפי הפורמט:

Opcode8	Const

כאשר Const הוא קבוע בגודל בין בית אחד לארבעה.

בפקודות הסתעפות מותנות Const תמיד בית אחד.

פקודה בעלת מבנה מיוחד

ESC -	11011 XXX	MOD YYY R/M	Disp-Low	Disp-High

דוגמאות

הפקודה ADD AX,BX מקודדת בהקסה: 03 C3

בבינארי

Opcode	D	W	MOD	R E G	R/M
0 0 0 0 0 0	1	1	1 1	0 0 0	0 1 1

הפקודה MOV BX,AX מקודדת בהקסה: 8B D8

בבינארי

Opcode	D	W	MOD	R E G	R/M
1 0 0 0 1 0	1	1	1 1	0 1	1 0 0 0

הפקודה MOV AL,CH מקודדת בהקסה: 8A C5

בבינארי

Opcode	D	W	MOD	R E G	R/M
1 0 0 0 1 0	1	0	1 1	0 0	0 1 0 1

הפקודה MOV AX,[BX] מקודדת בהקסה: 8B 07

בבינארי

Opcode	D	W	MOD	R E G	R/M
1 0 0 0 1 0	1	1	0 0	0 0	0 1 1 1

הפקודה MOV CX,SS:[BX+DI] מקודדת בהקסה: 36 8B 09

בבינארי

Prefix	Opcod	D	W	MOD	R E G	R/M
0 0 1 1 0 1 1 0	1 0 0 0 1 0	1	1	0 0	0 0	1 0 0 1

<u>יצירת דוגמאות נוספות</u>

כתוב תוכנית חוקית מבחינת השפה באסמבלי (היא לא חייבת לרוץ) והרץ עליה את tasm עם האופציה /la. תקבל קובץ lst. עם פירוט הקידוד בהקסה.

לדוגמא הפקודה:

```
tasm /la myprog1.asm
```

תיצור קובץ בשם myprog1.lst שאותו תוכל לקרוא בתוכנית editor. ליד הפקודות תראה את פירוט הקידוד הבינארי: היסטי הפקודות ותוכנם.

מספר שורה	היסט	קידוד בהקסה	הפקודה באסמבלי
32	0028	03 C3	ADD AX,BX
33	002A	2B C3	SUB AX,BX
34	002C	8B 90 0080	MOV DX,[BX+SI+80h]
35	0030	8B 90 4000	MOV DX,[BX+SI+4000h]
36	0034	8B 07	MOV AX,[BX]
37	0036	B8 0040	MOV AX,64
38	0039	B9 0080	MOV CX,128
39	003C	8B 46 06	MOV AX,[BP+6]
40	003F	2E: 8B 5D 06	MOV BX,CS:[DI+6]
41	0043	26: 8B 07	MOV AX,ES:[BX]
42	0046	36: 8B 07	MOV AX,SS:[BX]
43	0049	2E: 8B 07	MOV AX,CS:[BX]
44	004C	26: 8B 07	MOV AX,ES:[BX]
45	004F	42	INC DX
46	0050	FE 00	INC BYTE PTR [BX+SI]
47	0052	FF 00	INC WORD PTR [BX+SI]
48	0054	03 14	ADD DX,[SI]
49	0056	03 14	ADD DX,DS:[SI]