

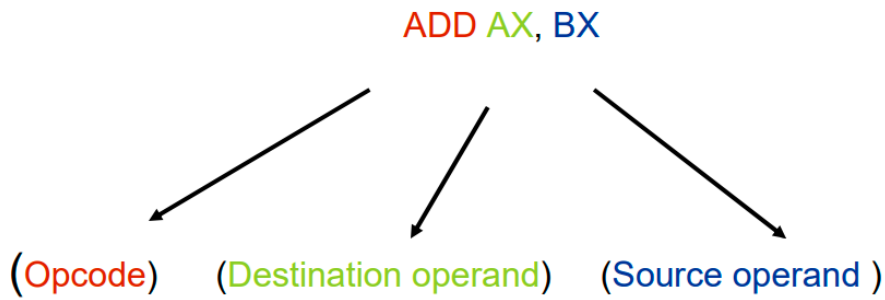
توليد ترميز التعليمات

الغاية من الجلسة:

تعليم الطالب استخلاص ترميز التعليمات البرمجية في المعالج 8086

مقدمة:

عند تنفيذ التعليمات في المعالج فإن هذه التعليمات تقسم إلى مجموعة من الأقسام كما يلي:



يتم تمثيل التعليمات في المعالج وفقاً لمجموعة من البتات:

Opcode - 6	D - 1	W - 1	1 st byte
MOD - 2	Reg - 3	R/M - 3	2 nd byte
Displacement or data (optional) up to 4 bytes			

حيث سنوضح فيما يلي كل قسم من هذه الأقسام:

opcode : بمقدار ستة بتات وهو يمثل ترميز العملية المنجزة ويختلف من عملية إلى أخرى كما قد يختلف باختلاف نمط العنوان.

D : بمقدار 1 بت يحدد موقع هدف التعليمات حسب ما يلي

D=0 فإن الهدف موجود في خانة R/M

D=1 فإن الهدف موجود في خانة REG

W : بمقدار 1 بت وهو يحدد حجم البيانات التي نتعامل معها.

W=0 فإن حجم البيانات التي نتعامل معها هو بايت واحد

W=0 فإن حجم البيانات التي نتعامل معها هو بايتين

MOD: بمقدار خانتين وهو يحدد نمط العنوان في التعليمات ويتم الحصول على قيم MOD كما يلي:

نمط العنوان	القيمة
موقع ذاكرة بدون إزاحة	00
موقع ذاكرة بإزاحة 8 بت	01
موقع ذاكرة بإزاحة 16 بت	10
عنوان مسجلات	11

Reg : بمقدار 3 بت تحدد المسجل الذي يتم التعامل معه ويبين الجدول التالي كيفية الحصول على قيم بنات هذا الحقل:

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

R/M بمقدار 3 بت تحدد هدف التعليمات.

حيث يتم الحصول على قيمة R/M من الجدول:

MOD = 11			EFFECTIVE ADDRESS CALCULATION			
R/M	W=0	W=1	R/M	MOD=00	MOD=01	MOD=10
000	AL	AX	000	(BX) + (SI)	(BX) + (SI) + D8	(BX) + (SI) + D16
001	CL	CX	001	(BX) + (DI)	(BX) + (DI) + D8	(BX) + (DI) + D16
010	DL	DX	010	(BP) + (SI)	(BP) + (SI) + D8	(BP) + (SI) + D16
011	BL	BX	011	(BP) + (DI)	(BP) + (DI) + D8	(BP) + (DI) + D16
100	AH	SP	100	(SI)	(SI) + D8	(SI) + D16
101	CH	BP	101	(DI)	(DI) + D8	(DI) + D16
110	DH	SI	110	DIRECT ADDRESS	(BP) + D8	(BP) + D16
111	BH	DI	111	(BX)	(BX) + D8	(BX) + D16

يحدد حقل Displacement or Data مقدار الإزاحة ضمن موقع الذاكرة الذي نتعامل معه ويحدد القيم الفورية ضمن التعليمات.

أمثلة محلولة:

1- اوجد ترميز التعليمة MOV BL,AL

ترميز عملية MOV هنا هو 100010

بما أن الهدف هو مسجل سيتم وضعه في حقل REG بالتالي D=1

بما أن المسجل AL وليس AX وبالتالي فإن التعامل مع تعليمات بطول 8bit وبالتالي W=0.

بما أننا ننقل من مسجل إلى مسجل أي MOD=11

بما أن المصدر هو AL فإن R/M=000 وبما أن الهدف هو BL بالتالي REG=011.

فيكون ترميز التعليمة هو $(8AD8)_{16}$

هناك بعض المترجمات تقوم بالبحث عن المسجل المصدر وتضعه في حقل REG بالتالي سيصبح لدينا

REG=000 R/M=011 كما أن قيمة الخانة D ستصبح 0 لأن الهدف سيصبح هو المسجل الموجود

ضمن حقل R/M

فيكون ترميز التعليمة هو $(88C3)_{16}$

2- اوجد ترميز التعليمة MOV AX,BX

MOV reg/mem to/from reg/mem

- This instruction has the structure:
100010dw modregr/m Disp-lo Disp-hi
- Where 0, 1 or 2 displacement bytes are present depending on the MOD bits
- **MOV AX, BX**
w = 1 because we are dealing with words
MOD = 11 because it is register-register
- if d = 0 then REG = source (BX) and R/M = dest (AX)
= 1000 1001 1101 1000 (89 D8)
- if d = 1 then REG = source (AX) and R/M = dest (BX)
= 1000 1011 1010 0011 (8B C3)

MOV reg/mem to/from reg/mem

- **MOV [BX+10h],CL**
 w = 0 because we are dealing with a byte
 d = 0 because we need R/M Table 2 to encode [BX+10h]
- therefore first byte is 10001000 = 88H
- since 10H can be encoded as an 8-bit displacement, we can use
 MOD=01 REG=001 and R/M=111 = 0100 1111 = 4FH
- and the last byte is 10H
 result: 88 4F 10
 Note: MOV [BX+10H],CX = 89 4F 10
- since 10H can also be encoded as a 16-bit displacement, we can use
 MOD=10 REG=001 and R/M=111 = 1000 1111 = 8FH
- and the last bytes are 00 10
 result: 88 8F 00 10

MOV reg/mem, imm

- This instruction has the structure:
 1100 011w MOD 000 R/M disp1 disp2
- Where 0, 1 or 2 displacement bytes are present depending on value of MOD
- **MOV BYTE PTR [100h],10h**
 w = 0 because we have byte operand
 MOD = 00 (R/M Table 1) R/M = 110 (Direct Addr)
 bytes 3 and 4 are address; byte 5 immediate data
- Result
 c6 06 00 01 10
- **MOV WORD PTR [BX+SI],10h**
 w = 1 because we have word operand
 MOD = 00 (R/M Table 1) R/M = 000 ([BX+SI])
 bytes 3 and 4 are immediate data
- Result
 c7 00 10 00

MOV imm to reg

- This instruction is optimized as a 4-bit opcode, with register encoded into the instruction

1011wreg

- Examples

MOV bx, 3	1011 w=1 reg=011=BX
10111011 imm	BB 03 00
MOV bh, 3	1011 w=0 reg=111=BH
10110111 imm	B7 03
MOV bl, 3	1011 w=0 reg=011=BL
10110011 imm	B3 03