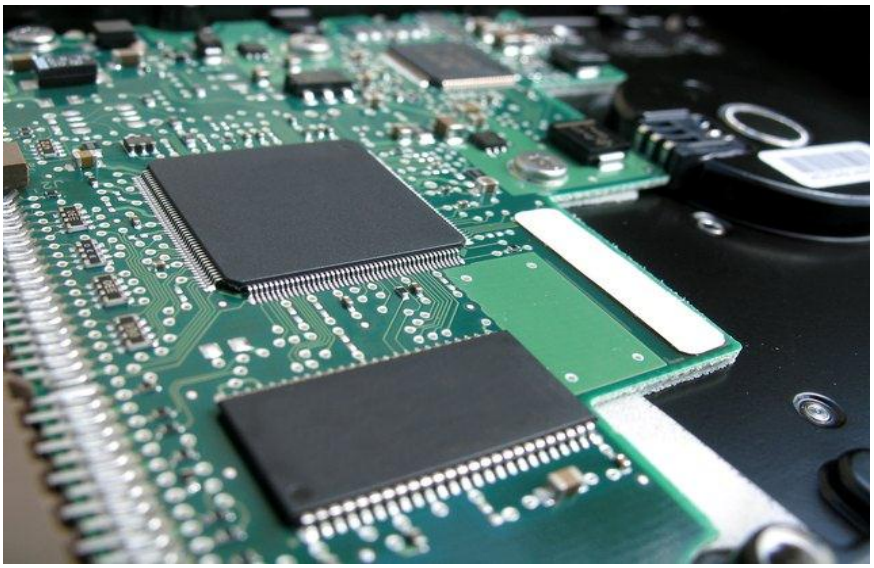
جامعة الحمدانية
كلية التربية

Microprocessors



Prepared By:

Firas Abdulrahman Yosif

Addressing Mode : part(2)

إن المعالج 8086 مزود بسبعة أنظمة عنونة مختلفة، وهذه انماط العنونة البعض منها يتعامل مع الذاكرة RAM والبعض الاخر يتعامل مع البيانات او مع السجلات داخل الـ processor ، وتشمل هذه الانماط الانواع التالية:

1. ***Register addressing mode***
2. ***Immediate addressing mode***
3. ***Direct addressing mode***
4. ***Indirect addressing mode***
5. ***Based addressing mode***
6. ***Indexed addressing mode***
7. ***Based indexed addressing mode***

في الجزء الثاني من هذه المحاضرة سوف نستكمل شرح بقية انماط العنونة والتي تشمل نمط العنونة القاعدية (Based addressing mode) ونمط عنونة الفهرسة (Indexed addressing mode) ونمط عنونة الفهرسة القاعدية (Based indexed addressing mode) .

5. Based addressing mode:

هذا النمط يسمى العنونة القاعدية وهو يتعامل فقط مع الذاكرة عن طريق السجلين BX و BP فقط .
وتقسم العنونة القاعدية الى قسمين:

A)Based addressing mode without displacement:

هذا النمط يسمى نمط العنونة القاعدية بدون ازاحة ويتعامل مع مواقع الذاكرة عن طريق السجلين BX , BP فقط. وهذا مثال على نمط عنونة قاعدية بدون ازاحة :

Example) MOV AL, [BX]

وظيفة هذا الايعاز هو نقل محتوى بايت واحد من موقع الذاكرة الذي عنوانه مخزون داخل السجل BX وتحويله الى السجل AL وخزنه بداخله وهذه العملية تتم بصورة افتراضية في المقطع DS وهذه العملية تمثل قراءة من الذاكرة.

Example) MOV AL, [BP]

وظيفة هذا الايعاز هو نقل محتوى بايت واحد من موقع الذاكرة الذي عنوانه مخزون داخل السجل BP وتحويله الى السجل AL وخزنه بداخله وهذه العملية تتم بصورة افتراضية في المقطع Stack segment (SS) وهذه العملية تمثل قراءة من الذاكرة. والشكل (1) عبارة عن يوضح الية عمل المثالين السابقين.

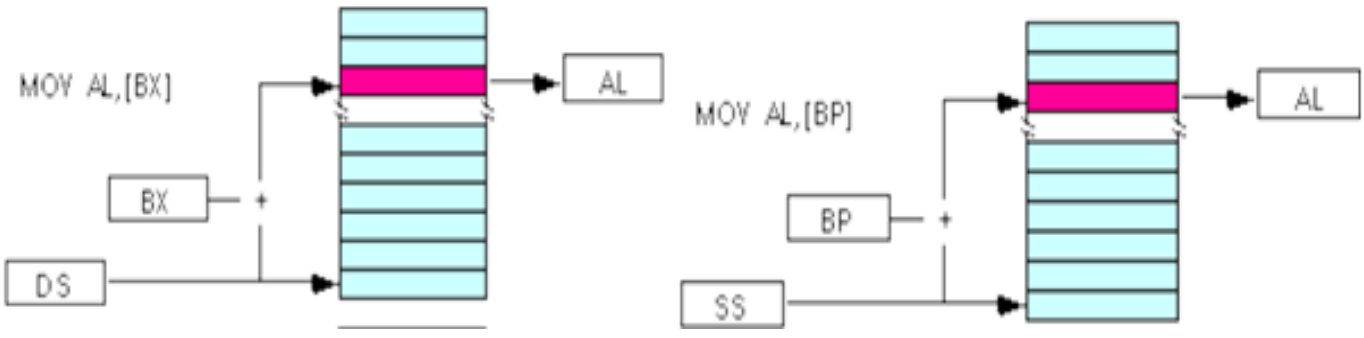
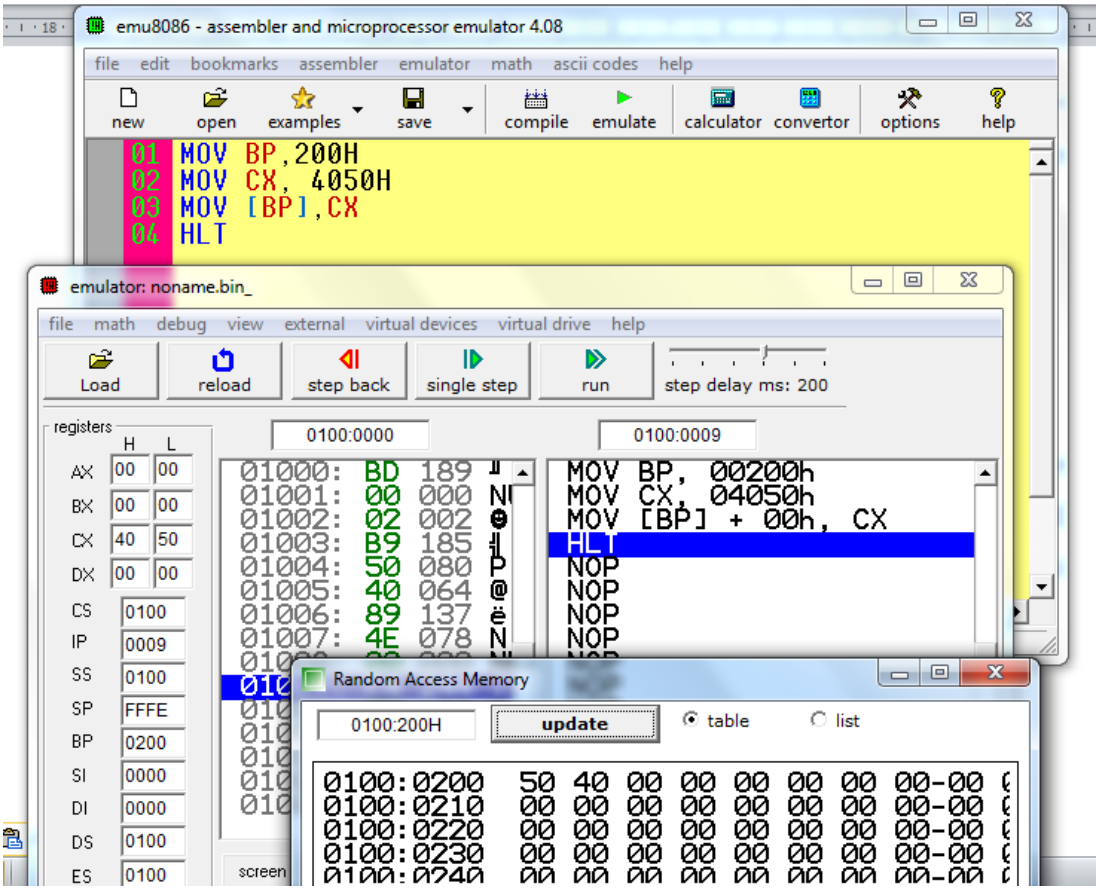


Fig. (1) Examples of Based addressing mode without displacement

وهذا برنامج يوضح العنونة القاعدية : Example)

```
MOV BP,2000H
MOV CX, 4050H
MOV [BP], CX
HLT
```

وظيفة هذا البرنامج هو تحميل القيمة 50H التي تمثل الـ low byte في موقع الذاكرة الذي عنوانه 2000H المخزون في السجل BP والقيمة الثانية 40H التي تمثل الـ high byte تخزن في موقع الذاكرة الذي عنوانه 2001H المخزون في السجل BP+1 . لاحظ تنفيذ البرنامج بأستخدام الـ emulator 8086 .



B) Based addressing mode with displacement:

هذا النمط يسمى نمط العنونة القاعدية مع ازاحة ويتعامل مع السجلات BP , BX. ولكن هذا النوع من العنونة يستخدم ازاحة اي انتقال في مواقع الذاكرة بعد اضافة رقم الى السجلات القاعدية BP او BX ، والازاحة اما ان تكون ذات سعة 8 bit او 16 bit . الازاحة 8bit المقصود بها اضافة رقم متكون من بايت الى احد السجلين BP او BX مثال على ذلك :

MOV DL, [BP+8 Disp.]

والمقصود بهذا الايعاز هو نقل محتويات العنوان الموجود في السجل BP مضاف اليه ازاحة مقدارها 8 bit داخل المقطع SS ثم نقلها الى السجل DL . مثال اخر:

MOV [BX+8 Disp.], CH

وظيفة الايعاز هو نقل محتوى السجل CH الى موقع الذاكرة الذي عنوانه موجود في السجل BX مضاف اليه ازاحة مقدارها 8bit .

و الازاحة 16 bit اي (16 bit displacement) والمقصود بها اضافة رقم مكون من 2 byte الى احد السجلين BP او BX ، مثال على ذلك :

MOV CL, [BP+16 Disp.]

والمقصود بهذا الايعاز هو نسخ محتويات العنوان الموجود في السجل BP مضاف اليه ازاحة مقدارها 16 bit الى مواقع ذاكرة داخل المقطع SS ثم نقلها الى السجل CL .

وهذه بعض الامثلة على نمط العنونة القاعدية مع ازاحة 8bit اوازاحة 16 bit :

MOV AH, [BP + 4]

MOV [BX+55H], DL

MOV CH, [BP+ 3344H]

MOV [BX -3667H], DX

ويمكن كتابة ايعازات عنونة القاعدية مع الازاحة بصيغة اخرى كما في الامثلة التالية:

MOV 34H[BX], CL

MOV DX, 4989H[BP]

Example) What's contain of CX after execution this program?

MOV [77H], 6688H

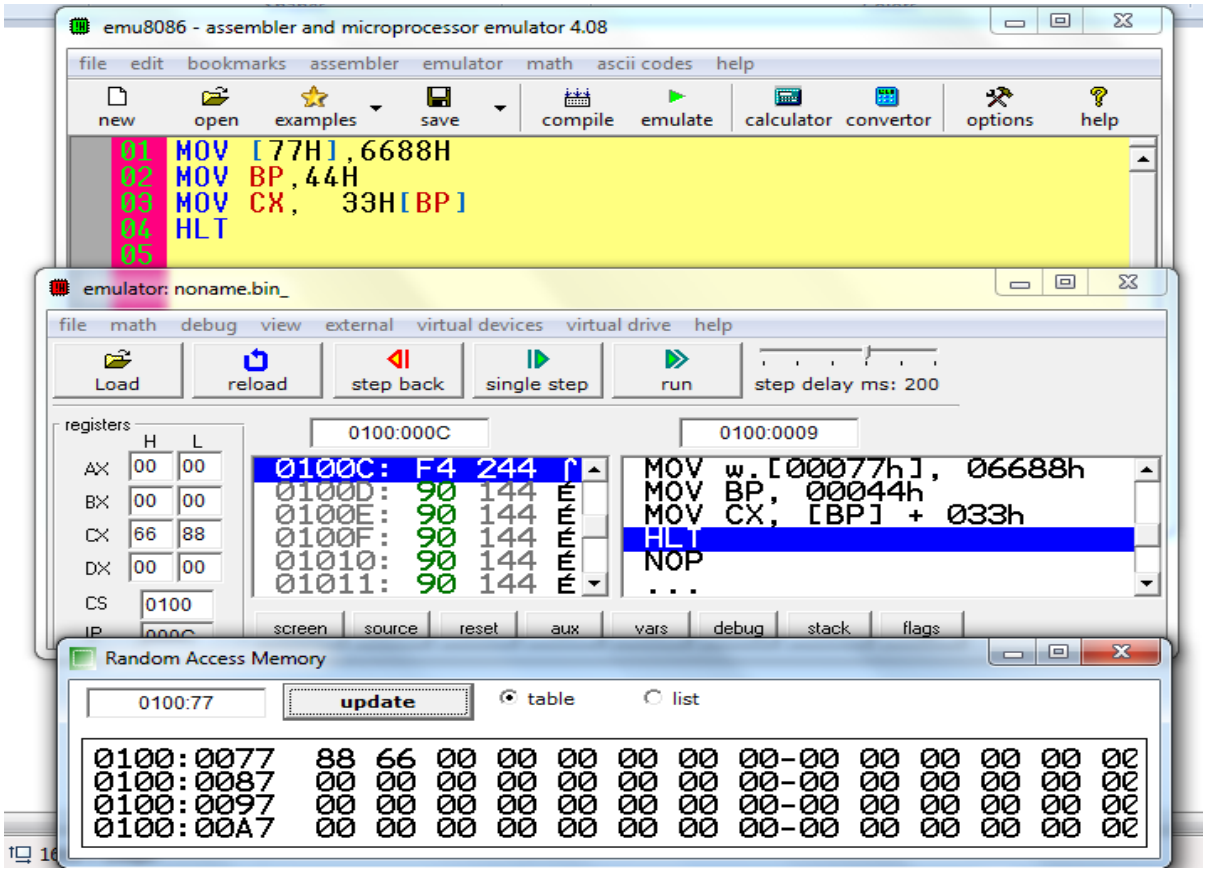
MOV BP,44H

MOV CX, 33H [BP]

HLT

Solution) CX= 6688H

لاحظ نتيجة السجل CX بعد تنفيذ البرنامج بأستخدام الـ EMULATOR 8086 :



6. Indexed addressing mode

هذا النمط يسمى عنونة الفهرسة وهو يتعامل فقط مع الذاكرة عن طريق السجلين SI و DI فقط .
وتقسم العنونة المفهرسة الى قسمين:

a)Index addressing mode without displacement:

هذا النمط يسمى نمط عنونة الفهرسة بدون ازاحة اي ليس هناك رقم يضاف الى سجلات الفهرسة.
ويتعامل هذا النمط فقط مع السجلات SI , DI ولكن بصورة منفصلة اي يتعامل مع احد السجلات داخل الايعاز الواحد وليس مع كليهما. وهذا مثال على نمط عنونة الفهرسة بدون ازاحة :

```
MOV DX, [SI]
```

يقصد في هذا الامر نقل محتويات عنوان الذاكرة الموجود داخل السجل SI والموقع SI+1 الى السجل DX .

The indexed addressing modes use the following syntax:

```
MOV DI, [SI]
MOV BX, [DI]
MOV [SI], 33H
MOV AL, [DI]
MOV [DI], 4455H
```

b) Index addressing mode with displacement:

هذا النمط يسمى نمط عنونة الفهرسة مع ازاحة اي ان هناك رقم يضاف الى سجلات الفهرسة او رقم يطرح من سجلات الفهرسة وقد تكون ازاحة 8bit او ازاحة 16bit .ويتعامل هذا النمط فقط مع السجلات SI , DI ولكن بصورة منفصلة اي يتعامل مع احد السجلات داخل الايعاز الواحد وليس مع كليهما.

وهذه امثلة على عنونة الفهرسة مع ازاحة 8 bit :

MOV [DI+32H], BX

MOV DH, [SI-4+12H]

MOV CL, [DI-20H]

ويقصد بالايعاز الاخير انقل محتويات عنوان الذاكرة الموجود داخل السجل DI مع طرح 20 H موقع من مواقع الذاكرة الى السجل CL (اي يحدث تناقص في مواقع الذاكرة الى الاعلى).

وهذا مثال على نمط عنونة الفهرسة مع ازاحة 16 bit :

MOV CL, [DI -234H]

MOV [SI]+1452H, BH

MOV AL, [SI-100H]

Example) What's the result for this program after execution it?

MOV AL,0FFH

MOV SI, 18H

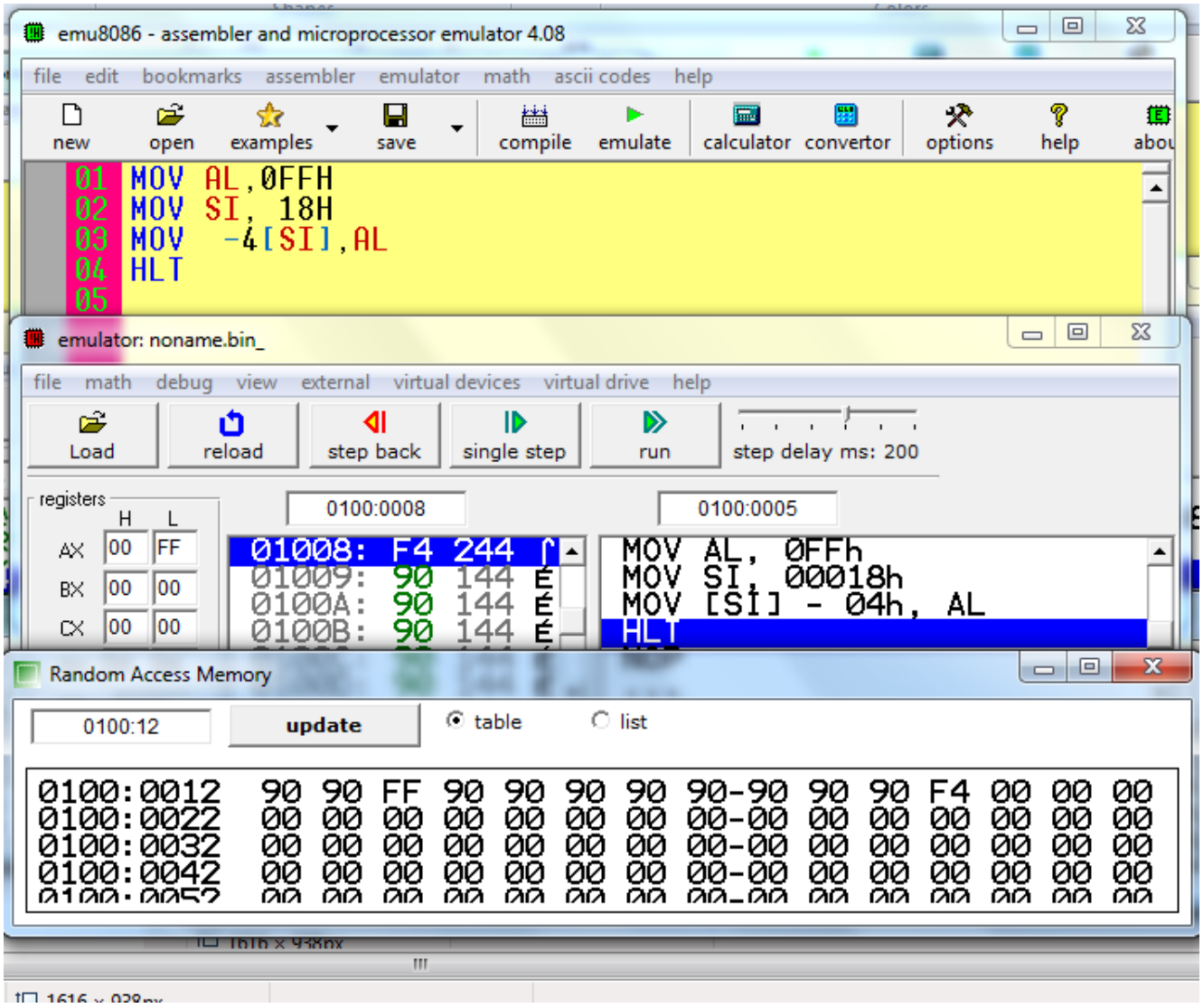
MOV -4[SI],AL

HLT

Solution)

[0014H]=0FFH

لاحظ محتوى الذاكرة الذي عنوانه 0014H بعد تنفيذ البرنامج اعلاه في المقطع DS بأستخدام الـ emulator 8086 :



7. Based indexed addressing mode:

هذا النمط يسمى نمط عنونة الفهرسة القاعدية يستخدم هذا النمط السجلات BP , BX , DI , SI داخل موقع الذاكرة . ينتج عن الجمع بين وضع العنونة القاعدية ووضع العنونة المفهرسة وضع جديد أكثر قوة يُعرف باسم وضع العنونة المبني على الفهرسة. يمكن استخدام وضع العنونة هذا للوصول إلى هياكل البيانات المعقدة مثل المصفوفات ثنائية الأبعاد. يمكن استخدام هذا الوضع للوصول إلى عناصر في مجموعة بيانات m X n . يحدد السجل الأساسي إحداثي m من المصفوفة ، ويحدد سجل

الفهرس إحداثي n يتيح تغيير القيم في مسجلات القاعدة والفهرسة الوصول إلى أي عنصر في المصفوفة.

وتقسم العنونة الفهرسة القاعدية الى قسمين:

a) Based indexed without displacement addressing mode:

في هذا النوع الذي يسمى نمط عنونة الفهرسة القاعدية بدون ازاحة حيث يتم دمج نمط عنونة القاعدية مع نمط عنونة المفهرسة في ايعاز واحد، حيث يتم استخدام سجلات القاعدية (**BX, BP**) مع سجلات الفهرسة (**SI, DI**) . وهذه بعض الامثلة على نمط عنونة الفهرسة القاعدية بدون ازاحة:

```
MOV AL, [BX+SI]
MOV [BX+DI], CX
MOV SP, [BP+SI]
```

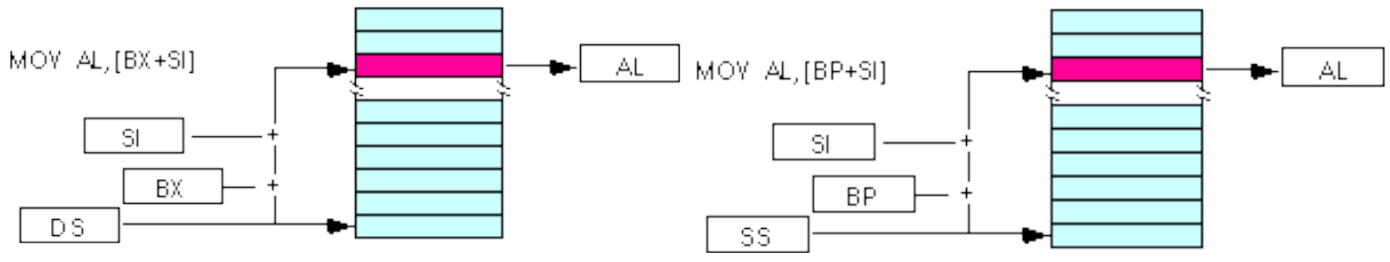
لاحظ صيغة اخرى لهذا النمط كما في الايعازين التاليين:

```
MOV AL, SI[BX]
```

والمقصود بهذا الايعاز نقل محتويات موقع الذاكرة الموجود داخل السجل SI مضاف اليه السجل BX داخل المقطع DS الى محتوى السجل AL.

```
MOV AL, SI[BP]
```

1وهذا الايعاز يمثل نقل محتويات موقع الذاكرة الموجود داخل السجل SI مضاف اليه السجل BP داخل المقطع SS الى محتوى السجل AL. وكما موضح في المخطط ادناه:



ملاحظة: لايجوز الجمع بين السجلين BX وBP داخل اقواس ذاكرة في ايعاز واحد كما في هذا المثال:

MOV [BX+BP], SP (False)

التصحيح: MOV [BX+SI], SP

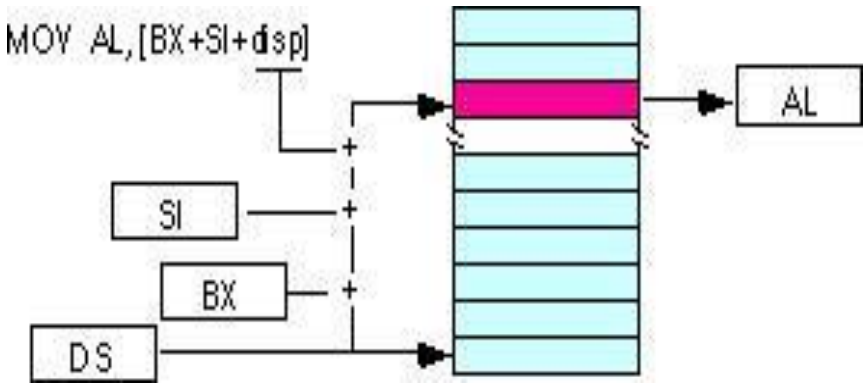
او MOV [DI+BP], SP

b) Based indexed with displacement addressing mode:

هذا النمط يسمى الفهرسة القاعدية مع ازاحة ففي هذا النوع من العنونة يتم استخدام ازاحة مع إضافة او طرح 8bit او 16 bit لعنوان الذاكرة. وهذا مثال يوضح هذا النمط:

MOV AL, [BX+SI+44H]

الية عمل الايعاز تتم عن طريق نقل محتوى موقع الذاكرة الذي عنوانه موجود في حاصل جمع محتوى السجل BX مع محتوى السجل SI ومضاف اليه ازاحة مقدارها 44H وهذا محتوى موقع الذاكرة ينقل الى محتوى السجل AL في الهدف داخل المقطع DS . والمخطط ادناه يوضح ذلك:



وهذه بعض الامثلة على نمط عنونة الفهرسة القاعدية مع ازاحة 8bit او ازاحة 16bit :

MOV CX, [BX+DI+23H]

MOV [SI+BP-84H], DL

MOV [SI+DI]+453H, 66H

MOV AX, [DI+BP-3975H]

وهذه بعض الايعازات الخاطئة وكيفية تصحيحها:

MOV SI, [BP+BX+0D3H] (False)

(لان في ايعاز واحد لايجوز استخدام سجلين قاعدة داخل اقواس ذاكرة)

التصحيح: MOV SI, [BX+ 0D3H]

او MOV SI, [BP+ 0D3H]

Example) What's result for this program after execute it?

MOV BP, 8

MOV DI, 4

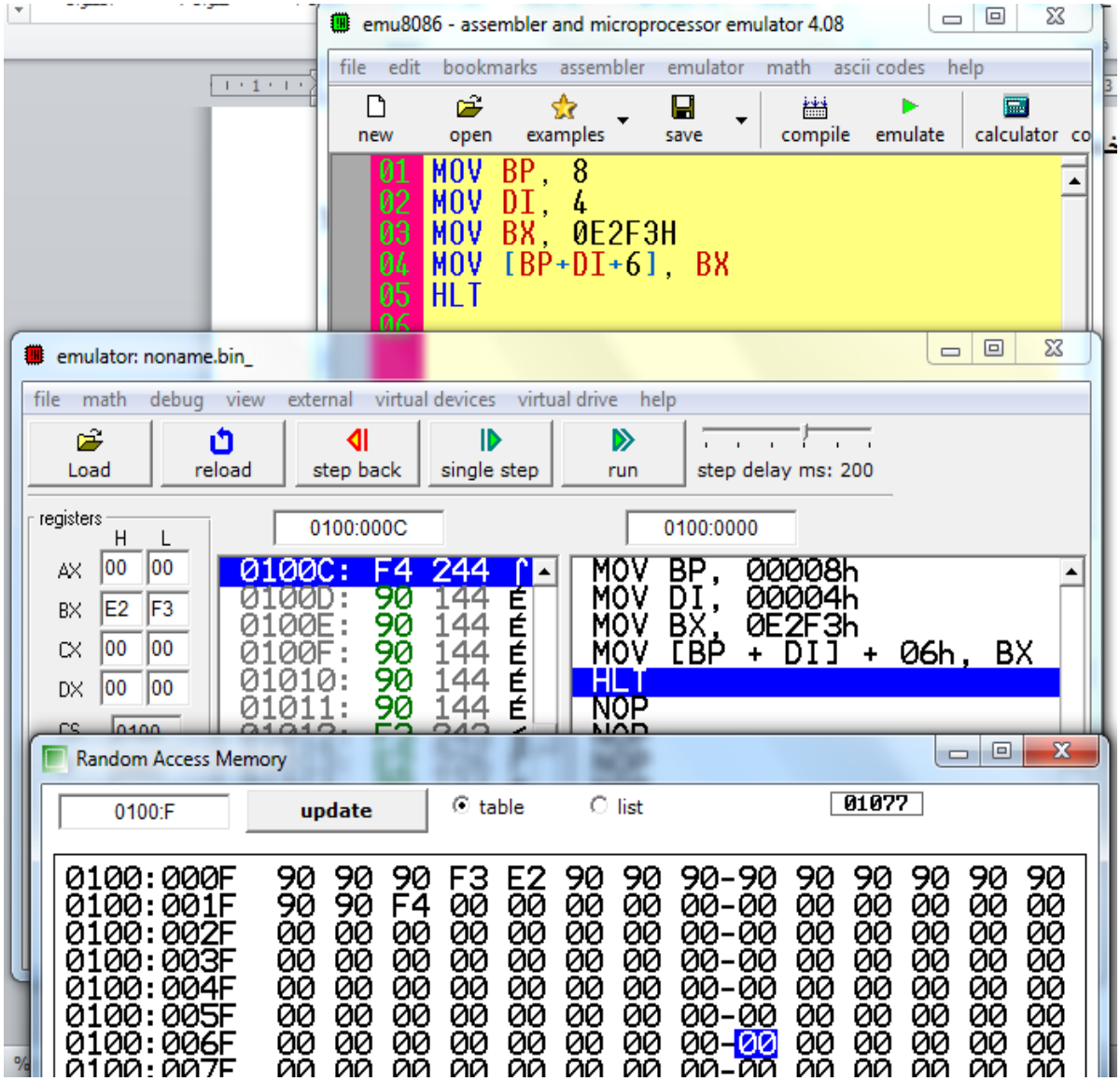
MOV BX, 0E2F3H

MOV [BP+DI+6], BX

HLT

Solution)

اي ان الموقع الذي عنوانه 12H الموجود في [BP+DI+6] يحتوي على القيمة F3H والموقع الذي عنوانه 13H المتمثل بالـ 1+ [BP+DI+6] يحتوي على القيمة E2H في المقطع SS. لاحظ تنفيذ البرنامج بأستخدام الـ emulator 8086 :


emu8086 - assembler and microprocessor emulator 4.08
file edit bookmarks assembler emulator math ascii codes help
new open examples save compile emulate calculator
01 MOV BP, 8
02 MOV DI, 4
03 MOV BX, 0E2F3H
04 MOV [BP+DI+6], BX
05 HLT
emulator: noname.bin_
file math debug view external virtual devices virtual drive help
Load reload step back single step run step delay ms: 200
registers
H L
AX 00 00
BX E2 F3
CX 00 00
DX 00 00
0100:000C
0100:0000
0100C: F4 244
0100D: 90 144
0100E: 90 144
0100F: 90 144
01010: 90 144
01011: 90 144
MOV BP, 00008h
MOV DI, 00004h
MOV BX, 0E2F3h
MOV [BP + DI] + 06h, BX
HLT
NOP
Random Access Memory
0100:F
update
table
list
01077
0100:000F 90 90 90 F3 E2 90 90 90-90 90 90 90 90 90
0100:001F 90 90 F4 00 00 00 00 00-00 00 00 00 00 00
0100:002F 00 00 00 00 00 00 00 00-00 00 00 00 00 00
0100:003F 00 00 00 00 00 00 00 00-00 00 00 00 00 00
0100:004F 00 00 00 00 00 00 00 00-00 00 00 00 00 00
0100:005F 00 00 00 00 00 00 00 00-00 00 00 00 00 00
0100:006F 00 00 00 00 00 00 00 00-00 00 00 00 00 00