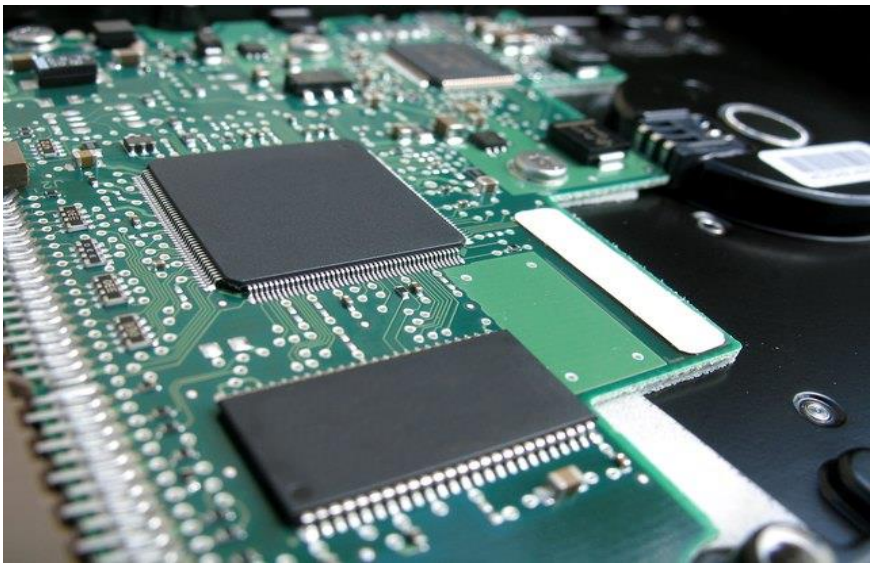
جامعة الحمدانية
كلية التربية

Lecture 8 (part1)

Microprocessors



Prepared By:

Firas Abdulrahman Yosif

Addressing Mode: part(1)

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 , وانماط العنونة هي:

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 الحالة الافتراضية (by default) عند كتابة البرنامج يظهر فقط في مقطع الشفرة CS وكذلك السجلات العامة تظهر في مقطع CS ولكن بعد ذلك يمكن تحويله وخزنه في مقاطع اخرى من الذاكرة .

1) Register addressing mode:

هذا النمط يتعامل مع السجلات التي تقع داخل الـ CPU مثال على ذلك MOV CX , AX حيث ان الايعاز MOV يمثل نقل البيانات من المصدر(source) الى الهدف(destination) اي ان وظيفة الايعاز هو وضع محتوى السجل AX الذي يقع في المصدر الى داخل محتوى السجل CX الذي يقع في الهدف , ويجب ان تكون محتويات سجل المصدر والهدف متساوية بالحجم اي اذا سجل

المصدر 16bit فيجب ان يكون سجل الهدف 16 bit واذا كان سجل المصدر يساوي 8bit فيجب ان يكون سجل الهدف 8bit . في هذا النوع من العنونة يتم تنفيذ الايعازات يتم دون الرجوع إلى الذاكرة أي يكون التنفيذ داخل المعالج ,على سبيل المثال السجلين AX و CX موجودين في وحدة الـ execution unit داخل المعالج فيكون التنفيذ داخليا في الـ CPU.

ملاحظة(1): لايجوز وضع CS في الهدف (Destination).

ملاحظة(2): في نمط عنونة السجلات ممكن التعامل مع جميع سجلات الاغراض العامة وكذلك سجلات التأشير وايضا مقاطع الذاكرة .

ملاحظة(3):لايمكن استخدام السجل IP في عنونة السجلاتregister addressing mode.

وهذه بعض الامثلة على هذا النوع من العنونة:

```
MOV AL, CL
MOV DS, SI
MOV DI, CX
MOV DX, ES
MOV DH, BL
MOV SS, DI
```

وهذه بعض الامثلة على الايعازات الخاطئة وكيفية تصحيحها:

1- MOV AX,CL (False)

تصحيح الايعاز: MOV AX, CX
او MOV AL,CL

2- MOV DL,DS (False)
(يجب ان يكون الهدف 16 bit) تصحيح الايعاز: MOV DX,DS

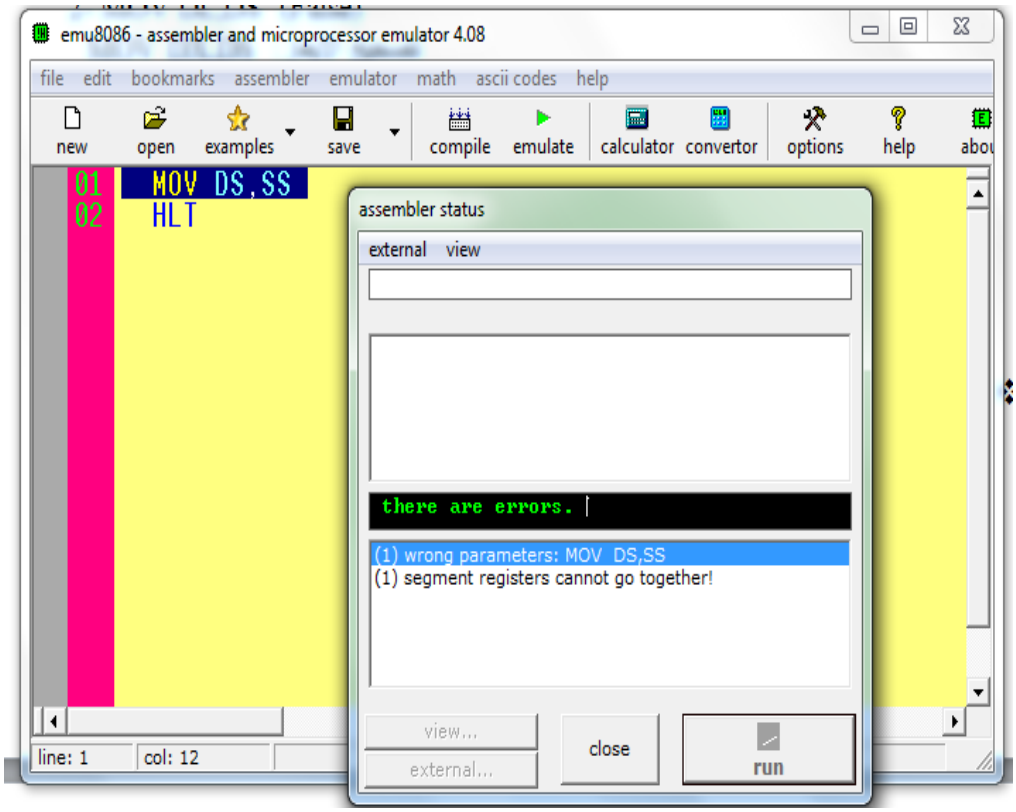
3- MOV CS, SI (False)
التصحيح: MOV SI,CS

4- MOV BP,BH (False)
MOV BP,BX : التصحيح

5- MOV DS,SS (False)
MOV BX, SS :التصحيح
MOV DS,BX

❖ في هذا الايعاز اعلاه احتاج الى سجل وسطي بين المقطعين لكي تتم عملية نقل البيانات.

و من خلال تنفيذ الايعاز MOV DS, SS في برنامج الـ emulator نلاحظ وجود خطأ في صيغة الايعاز بسبب التعامل مع مقطعي ذاكرة في نفس الايعاز وكما موضح في الشكل ادناه:



2) Immediate addressing mode:

هذا النمط يسمى بالعنونة الفورية حيث يتعامل مع البيانات بصورة مباشرة ويعتبر من اسرع الانماظ الموجودة لانه يتعامل مباشرة مع االبيانات مثال على ذلك:

MOV AX,34E6h حيث ان المصدر Source دائما يوجد فيه بيانات وغير مسموح ان تكون البيانات في الهدف Destination. نلاحظ من خلال المثال ان الرقم الذي في الهدف يتكون من 16 bit حيث ان اول قيمة على اليمين (E6) تمثل low byte وثاني قيمة على اليسار(4E) تمثل high byte.

وهذه بعض الامثلة على نمط العنونة الفورية :

MOV SI, 34H

MOV CL, 03DH

MOV BH, 0AH

MOV BP, 0C3H

MOV AX, 5647H

ملاحظة: لايجوز نقل البيانات مباشرة الى مقاطع الذاكرة ولكن ممكن استخدام سجلات وسطية مثل سجلات الاغراض العامة(AX,DX,CX,BX) او سجلات الازاحة (SI,DI,SP,BP).

وهذه بعض الامثلة على الايعازات الخاطئة وكيفية تصحيحها:

1- MOV SS, 3344H (False)

التصحيح: MOV CX, 3344H

MOV SS, CX

2- MOV SI, 02F4D4H (False)

لايجوز استخدام بيانات اكثر من 16 بت التصحيح: MOV SI, 0F4D4H

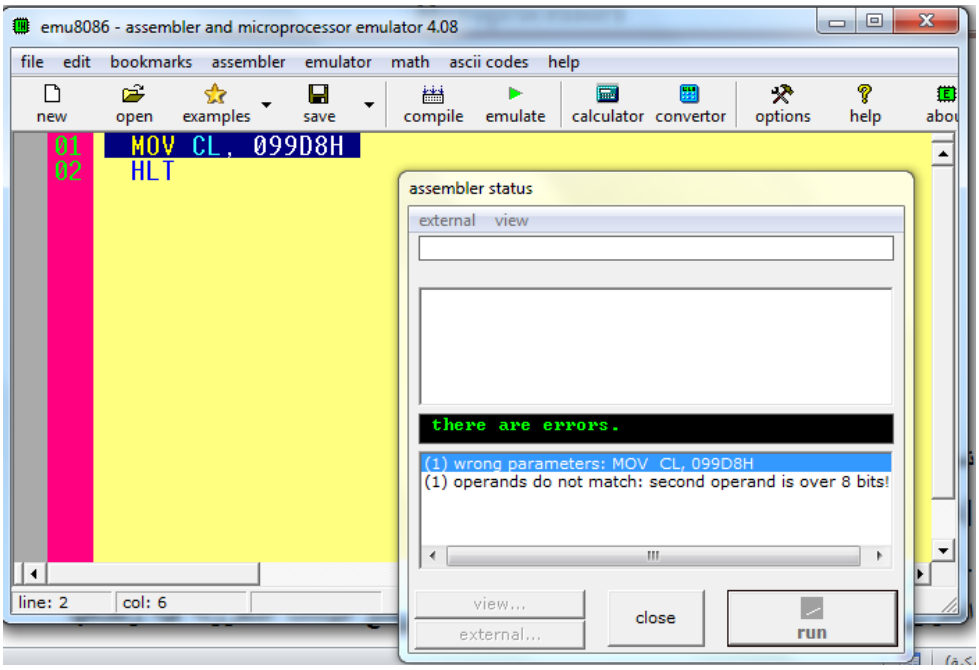
❖ في الايعاز اعلاه تم استخدام بيانات اكثر من 2byte وهذا لايجوز.

3- MOV CL, 099D8H (False)
MOV CX, 099D8H : التصحيح

MOV CL, 0D8H او

MOV CH, 99H

نلاحظ عند استخدام برنامج الـ emulator لتنفيذ الايعاز MOV CL,099D8H وجود خطا في الايعاز بسبب ان سعة البيانات اكبر من حجم السجل , وكما نلاحظ في الشكل ادناه:



3) Direct addressing mode:

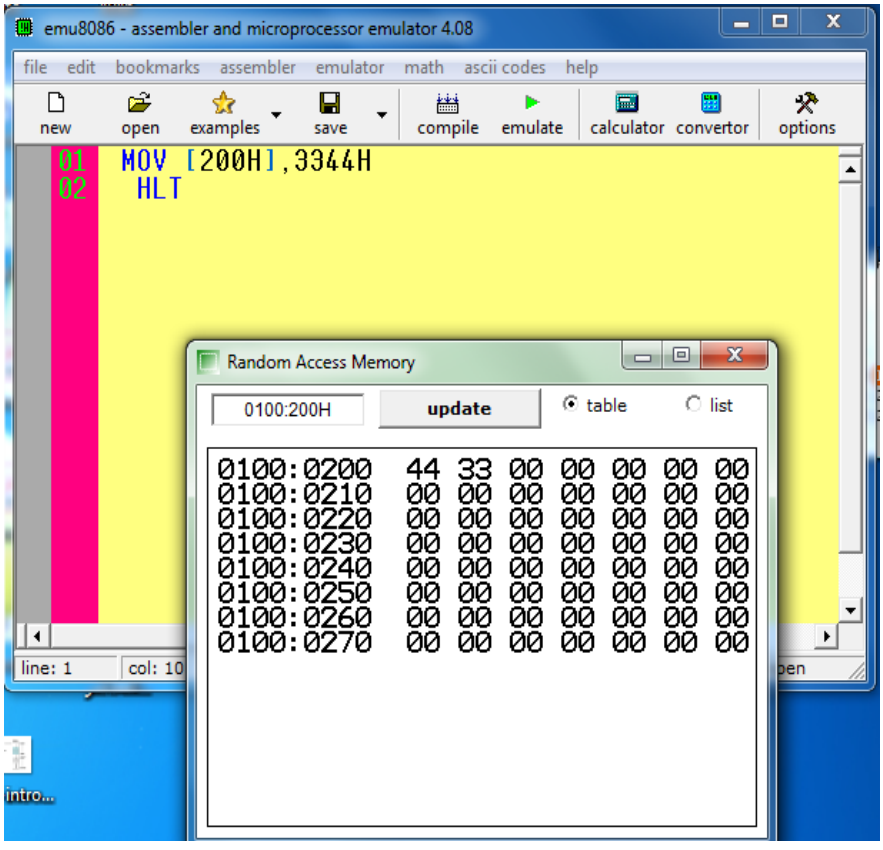
نمط العنونة المباشرة يتعامل مع الذاكرة فقط اما يستقبل منها البيانات اي قراءة من الذاكرة مثل:

MOV AL , [1234h] او يرسل اليها البيانات اي يكتب في الذاكرة مثل:

MOV [4523h] , CL نلاحظ من خلال هذا النمط من العنونة ان العنوان مذكور بشكل صريح في الايعاز حيث يوضع العنوان على الـ address bus ثم نفتح العنوان المذكور ونضع البيانات المخزونة فيه ونضعها على الـ data bus ثم جلبها الى المعالج ومن ثم يفتح السجل المعني للكتابة عليه وخزن القيمة فيه.

وفي حالة التعامل مع مواقع الذاكرة باستخدام بيانات حجمها 16 bit ففي هذه الحالة يتم التعامل مع موقعين من مواقع الذاكرة كما موضح في هذا المثال:

MOV [200H], 3344H اي يتم خزن القيمة 44H في موقع الذي عنوانه 200H وخزن القيمة 33H في الموقع 201H , وكما هو موضح في الذاكرة بعد تنفيذ الايعاز باستخدام برنامج الـ emulator .



وهذه بعض الامثلة على نمط العنونة المباشرة:

MOV AL, [1300H]

MOV [5000H], DX

MOV SI, [7000H]

MOV [2333H], ES

وهذه بعض الامثلة على الايعازات الخاطئة وكيفية تصحيحها:

1- MOV CS, [5677H] (False)

التصحيح: MOV DS, [5677H]

2- MOV [1222H], IP (False)

التصحيح: MOV [1222H], SI

3- MOV [1000H], [8000H] (False)

التصحيح : MOV CL, [8000H]

MOV [1000H], CL

4) Indirect addressing mode:

نمط العنونة الغير مباشرة في هذا النمط يكون العنوان مذكور بشكل غير صريح داخل اقواس الذاكرة . اي سيكون الوصول الى بيانات موجودة في الذاكرة ولكن العنوان الموجود في الذاكرة سيكون في مسجلات هي BP , DI , SI, BX لان هذه السجلات هي التي فقط تتعامل مع مواقع الذاكرة. مثال على ذلك : MOV AX,[BX] هذا الايعاز يوضح نسخ البيانات الموجودة في موقع الذاكرة الذي عنوانه موجود في السجل BX ونقل محتويات الموقع الى السجل AX وهذا مايسمى بالعنونة الغير مباشرة Indirect addressing mode .

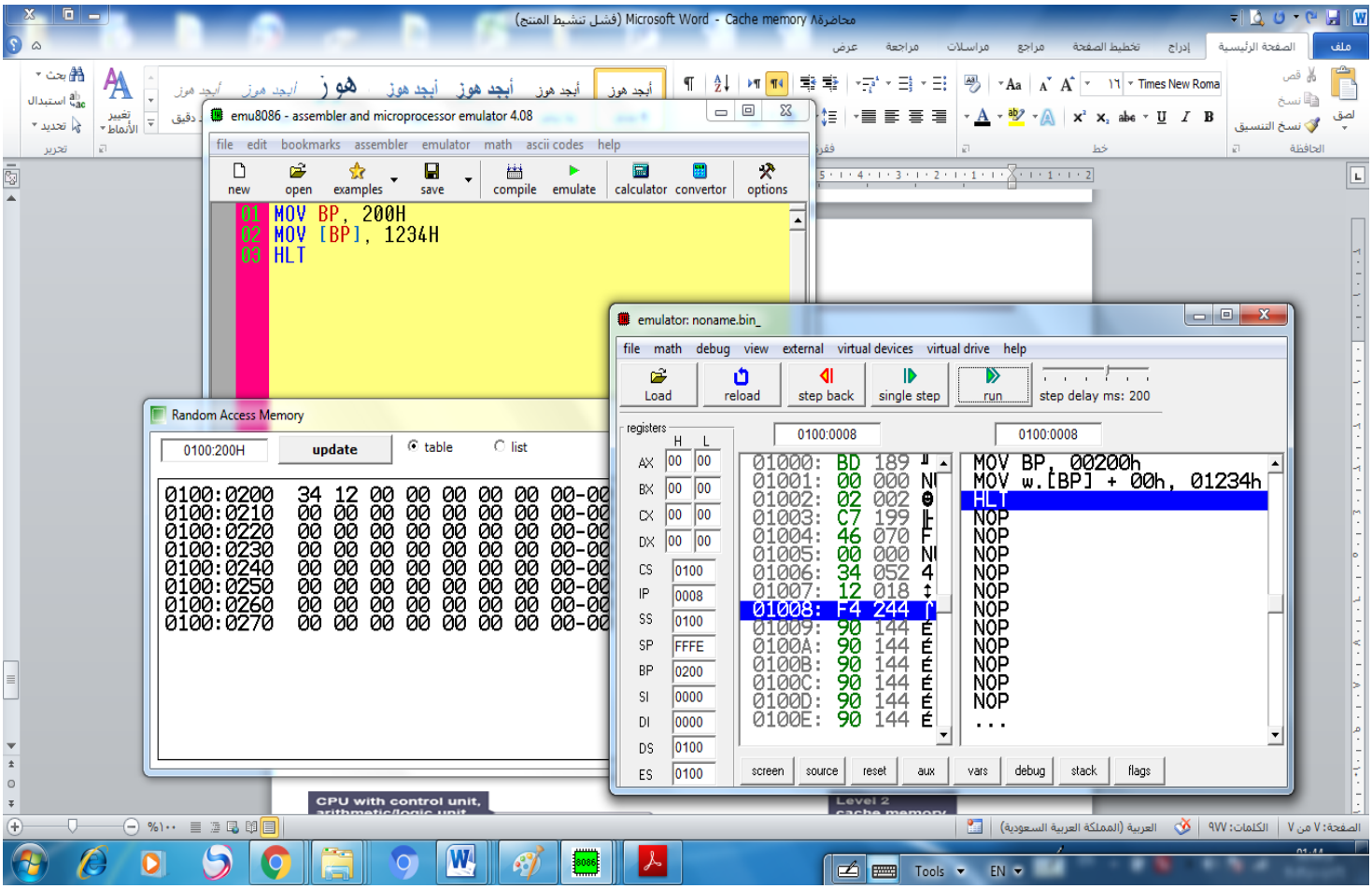
EX)

MOV BP, 200H

MOV [BP], 1234h

HLT

لاحظ الية عمل البرنامج اعلاه: اي ضع البيانات 34h الذي يمثل low byte في موقع الذاكرة الذي عنوانه 200H الموجود في [BP] ثم ضع البيانات 12h الذي يمثل high byte في موقع الذاكرة الذي يليه وعنوانه 201H الذي يمثل [BP+1] ضمن المقطع SS , الشكل ادناه يوضح تنفيذ البرنامج باستخدام الـ emulator.



وهذه امثلة اخرى على العنونة الغير مباشرة:

MOV [DI], 12H

MOV BP, [SI]

MOV [BX], DH

MOV BX, [BP]

وهذه بعض الامثلة على الايعازات الخاطئة وكيفية تصحيحها:

1- MOV [DX], 3444H (False)

MOV [SI], 3444H التصحيح :

2- MOV [CS], CL (False)

MOV [BP], CL التصحيح:

❖ نلاحظ في المثال اعلاه يجب عدم وضع مقطع الذاكرة بين اقواس ذاكرة داخل الايعاز.

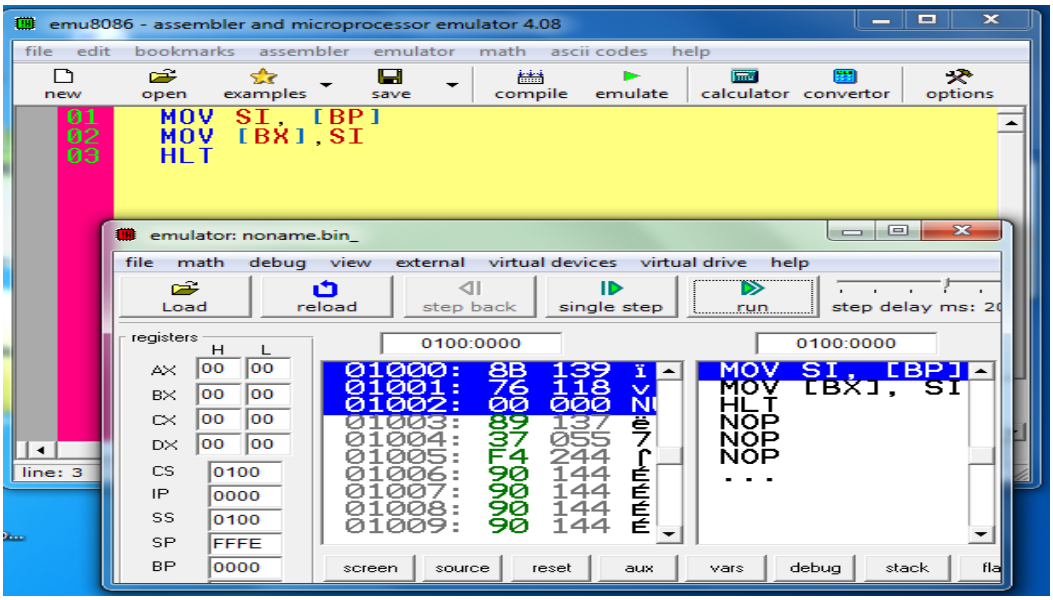
3- MOV [BX], [BP] (False)

MOV CX, [BP] التصحيح:

MOV [BX], CX

❖ ملاحظة: في الايعاز السابق لايمكن نقل بيانات بين موقعي ذاكرة في نفس الايعاز ولكن ممكن استخدام سجل وسطي مثل سجلات الاغراض العامة او سجلات الازاحة.

لاحظ تنفيذ الايعاز MOV [BX], [BP] بصيغة اخرى كما موضح في برنامج الـ emulator بعد التنفيذ:



تستخدم أوضاع [BX] و [SI] و [DI] مقطع DS افتراضيًا, ويستخدم وضع العنونة [BP] مقطع المكدس (SS) افتراضيًا. ولكن يمكنك استخدام رموز بادئة تجاوز المقطع إذا كنت ترغب في الوصول إلى البيانات في قطاعات مختلفة. توضح الايعازات التالية استخدام مقاطع مختلفة:

```
MOV SI, CS:[BX]
MOV AL, DS:[BP]
MOV CH, SS:[SI]
MOV BL, ES:[DI]
MOV ES:[BP], DI
```

يمكن وضع البادئة DS قبل الـ MOV في هذه الحالة يتعامل BP مع المقطع DS كما في المثال التالي:

```
DS: MOV [BP], 334H
HLT
```

نلاحظ تنفيذ الايعاز DS: MOV [BP], 334H باستخدام برنامج الـemulator كما موضح في الشكل ادناه1 :