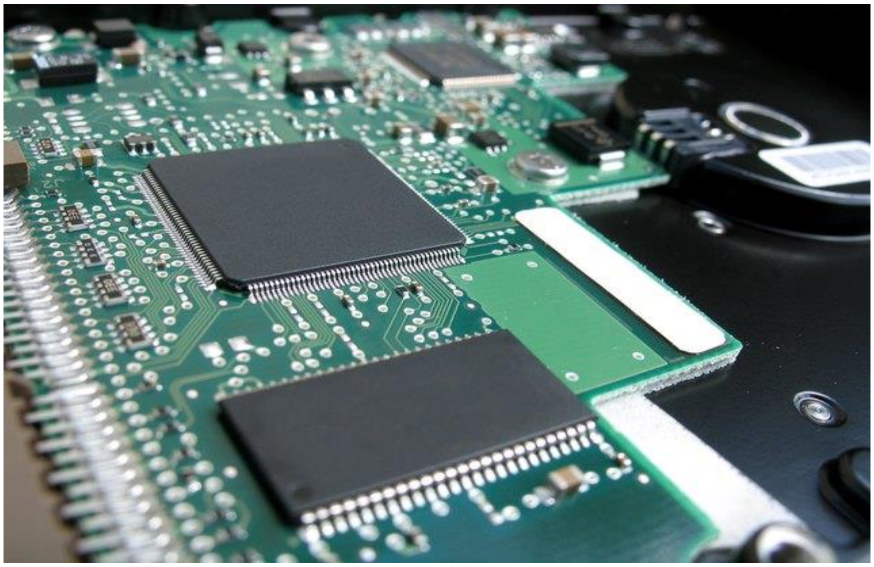
جامعة الحمدانية
كلية التربية

Lecture 15

Microprocessors



Prepared by:

Firas Abdulrahman Yosif

Multiplication & Division Instructions

The 8086 can perform arithmetic operations on binary numbers (signed or unsigned) and on decimal numbers.

1. Multiplication Instructions

Multiply (MUL) multiplies unsigned numbers and Integer Multiply (IMUL) multiplies signed numbers. Both can multiply bytes or words.

ايعاز الضرب MUL يضرب ارقام بدون اشارة والايعاز IMUL يضرب الارقام مع الاشارة وكلاهما يضرب ارقام بطول byte او word .

The 8086 multiply instructions have the general forms:

MUL Source

IMUL Source

ملاحظة: المصدر اما ان يكون register او memory .

Note:

- The product after a multiplication is always a double-width product. If two 8-bit numbers are multiplied they generate a 16-bit product. If two 16-bit numbers are multiplied, they generate a 32-bit product.

<u>There are two types of multiplication:</u>

a) 8-bit Multiplication

Examples:

- MUL CL ; AL is multiplied by CL and the unsigned product store in AX
- MUL [DI]

ملاحظة(1): لايمكن وضع مقاطع الذاكرة Segment memory في الـ source , وكذلك لايجوز وضع data في الـ source .

Example:

- MUL DS ;false

Example:

- MUL 40H ;false

Correction:

```
MOV BH, 40H
MUL BH
```

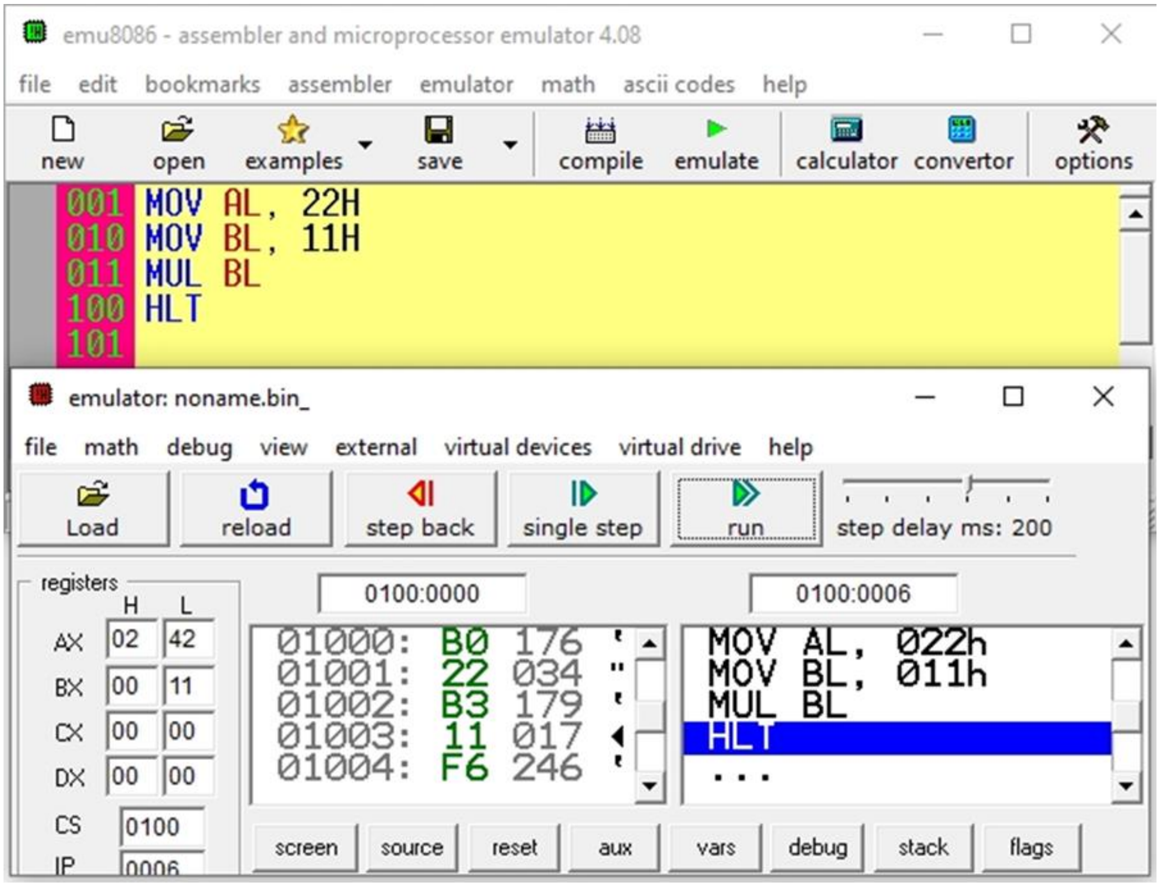
ملاحظة(2): عند ضرب رقمين يجب وضع احد الرقمين في AL ونتيجة الضرب في AX أي ان السجل AL يكون مخفي داخل ايعاز الـ MUL .

Example (1): Write a program in assembly language to find result of

11H X 22H

Solution:

```
MOV AL, 22H
MOV BL, 11H
MUL BL
HLT
AX=0242H
```


emu8086 - assembler and microprocessor emulator 4.08
file edit bookmarks assembler emulator math ascii codes help
new open examples save compile emulate calculator convertor options
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
screen source reset aux vars debug stack flags

```
MOV AL, 22H
MOV BL, 11H
MUL BL
HLT
```

	H	L
AX	02	42
BX	00	11
CX	00	00
DX	00	00
CS	0100	
IP	0006	

```
0100:0000
01000: B0 176 ░
01001: 22 034 "
01002: B3 179 ░
01003: 11 017 ◄
01004: F6 246 ÷
```

```
0100:0006
MOV AL, 022h
MOV BL, 011h
MUL BL
HLT
...
```

Example (2):

Write instructions to multiply the content of BL by CL, Load BL with 5 & load CL with 10h. Store the result in DX register.

Answer)

MOV BL, 5

MOV CL,10h

MOV AL, CL

MUL BL

MOV DX, AX

HLT

Example (3): Find the result of this program:

MOV AL,11H

MOV BL, 22H

MUL AL

HLT

Solution) AX=121H

Example (4): What's result for this program?

MOV [100H],32H

MOV AL, 13H

MUL [100H]

HLT

Solution) AX=03B6H

b) 16-bit Multiplication

□ AX contains the multiplicand instead of AL.

□ The 32-bit product appears in DX-AX instead of AX.

Ex (1): MUL CX ;AX is multiplied by CX and the unsigned product is store in DX & AX registers.

Ex (2): MOV CX, 54H
MOV AX,367H
MUL AX
HLT

EX (3): Write a program to multiply 1111h x 22h , using arithmetic instructions then find the result?

Answer:

MOV BX, 1111H

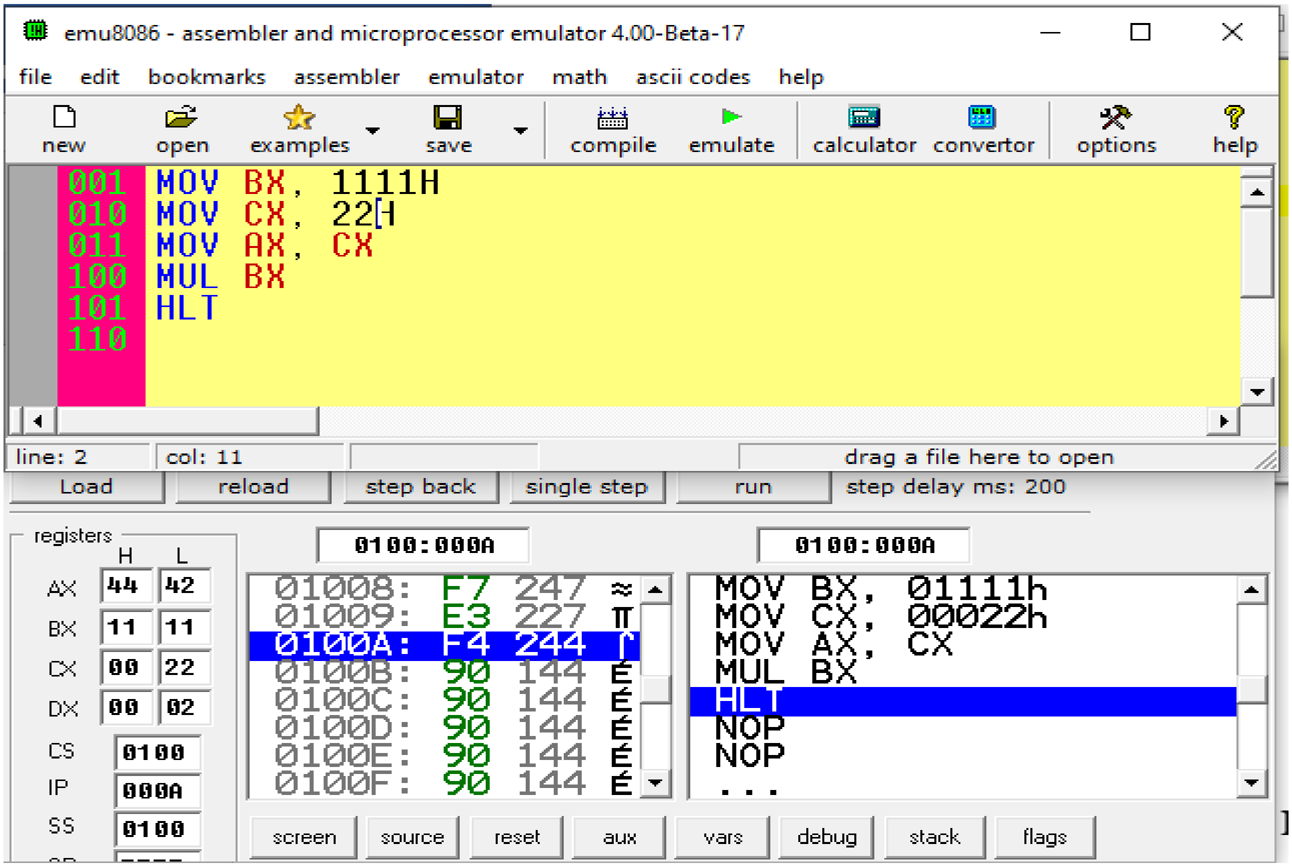
MOV CX, 22H

MOV AX, CX

MUL BX

HLT

AX=4442H, DX=02


emu8086 - assembler and microprocessor emulator 4.00-Beta-17
file edit bookmarks assembler emulator math ascii codes help
new open examples save compile emulate calculator convertor options help
Load reload step back single step run
drag a file here to open
screen source reset aux vars debug stack flags

```
MOV BX, 1111H
MOV CX, 22H
MOV AX, CX
MUL BX
HLT
```

line: 2 col: 11

step delay ms: 200

registers

	H	L
AX	44	42
BX	11	11
CX	00	22
DX	00	02
CS	0100	
IP	000A	
SS	0100	

```
0100:000A
01008: F7 247 ≈
01009: E3 227 π
0100A: F4 244 ⌠
0100B: 90 144 É
0100C: 90 144 É
0100D: 90 144 É
0100E: 90 144 É
0100F: 90 144 É
```

```
0100:000A
MOV BX, 01111h
MOV CX, 00022h
MOV AX, CX
MUL BX
HLT
NOP
NOP
...
```

2. Division Instructions:

Divide (DIV) performs an unsigned division, while Integer Divide (IDIV) performs a signed division.

ان وظيفة الايعاز DIV هو القسمة الاعداد بدون اشارة بينما IDIV يقسم الارقام مع الاشارة.

The form is:

- **DIV Source**
- **IDIV Source**

ممكن ان يكون المصدر(source) عبارة عن memory او register ولايمكن وضع أي بيانات داخل الـ source .

They are two types of division:

a. 8-bit Division:

- An 8-bit division uses the AX register to store the dividend that is divided by the contents of any 8-bit register or memory location.
- The result moves into AL while AH containing a whole number remainder.

ملاحظة (1): نفس الشروط التي تنطبق بالضرب تنطبق على عملية القسمة.

ملاحظة (2): عند قسمة byte على byte يجب وضع المقسوم في سجل AL والمقسوم عليه اما ان يكون في سجل او ذاكرة وناتج القسمة يكون في السجل AL وباقي القسمة في السجل AH.

<u>Examples of division without sign:</u>

- DIV [SI]

في هذا المثال يتم قسمة محتوى السجل AL الذي يمثل المقسوم والموجود ضمنيا داخل ايعاز القسمة على محتوى موقع الذاكرة المتمثل بالعنوان الموجود داخل السجل SI والذي يمثل المقسوم عليه.

- DIV BL

هنا يتم قسمة محتوى السجل AL والذي يمثل المقسوم على محتوى السجل BL الذي يمثل المقسوم عليه.

- MOV AL, 30H

```
DIV 23H       ;False
    The correction is:
    MOV AL, 30H
    MOV AH, 23H
    DIV AH
```

Example (1): What's this program print?

```
MOV AL, 95H
MOV CL, 10H
DIV CL
HLT
```

Solution) AL=9, AH=5

توضيح (1): الرقم 95 يمثل المقسوم والرقم 10 يمثل المقسوم عليه.

توضيح (2): AL= 9 يمثل ناتج القسمة و AH = 5 يمثل باقي القسمة.

Example (2): Write a program in assembly language to divide 37H/3 and find the result?

Solution:

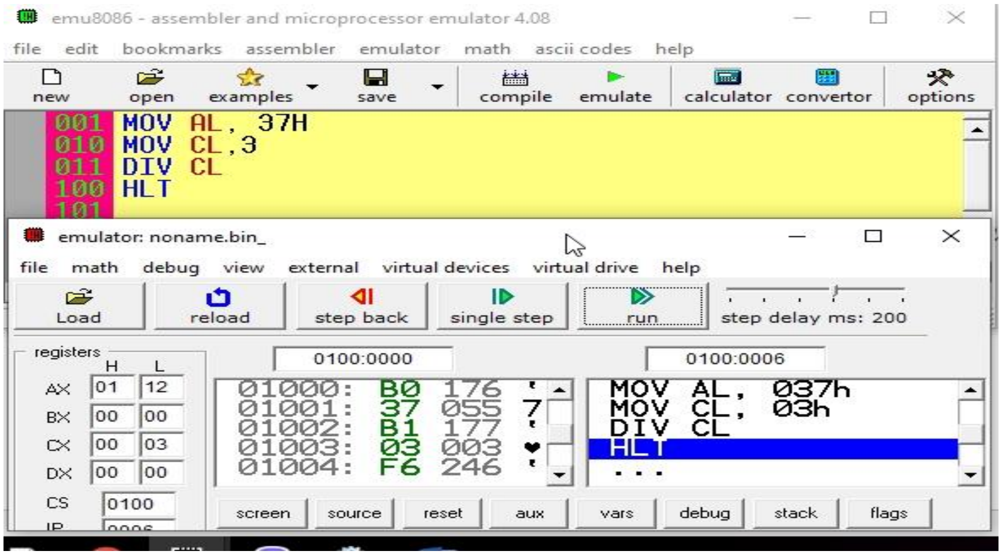
MOV AL, 37H

MOV CL,3

DIV CL

HLT

AL= 12H, AH=1


emu8086 - assembler and microprocessor emulator 4.08
file edit bookmarks assembler emulator math ascii codes help
new open examples save compile emulate calculator convertor options
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
screen source reset aux vars debug stack flags

```
MOV AL, 37H
MOV CL,3
DIV CL
HLT
```

	H	L
AX	01	12
BX	00	00
CX	00	03
DX	00	00
CS	0100	

0100:0000

```
01000: B0 176
01001: 37 055 7
01002: B1 177
01003: 03 003 ❤
01004: F6 246
```

0100:0006

```
MOV AL, 037h
MOV CL, 03h
DIV CL
HLT
...
```

Example (3): Write a program in assembly language to divide 29H/2 and find the result?

Solution:

MOV AL,29H

MOV BH, 2

DIV BH

HLT

AL=14H (ناتج القسمة), AH=1(باقي القسمة)

Note:

Division can result in two different types of errors:

- Divide by zero.
- Divide overflow.

b. 16-bit Division:

Ex) DIV CX ; AX is divided by CX and the unsigned result is store in AX and unsigned remainder is in DX.

Example (1) Write a program in assembly language to divide AX/CX. If AX = -100 and CX= 9?

Solution:

```
MOV AX, -100H
MOV CX, 9
CWD     ; CWD (it means convert word to double word )
IDIV CX

 HLT
```

Example (4): Write a program to divide AL/[SI], if AL=8, SI=300H and [SI]=5.

Solution:

MOV AL,8

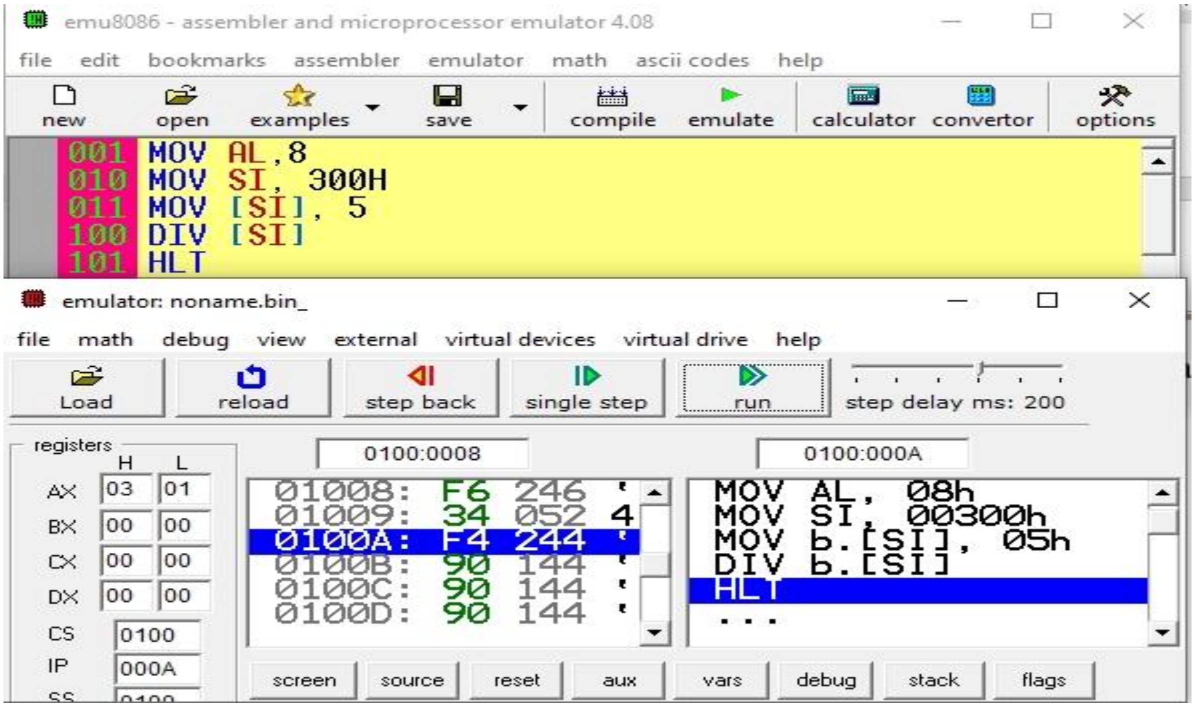
MOV SI, 300H

MOV [SI], 5

DIV [SI]

HLT

AL=1 (ناتج القسمة), AH= 3 (باقي القسمة)


emu8086 - assembler and microprocessor emulator 4.08
file edit bookmarks assembler emulator math ascii codes help
new open examples save compile emulate calculator convertor options
emulator: noname.bin_
file math debug view external virtual devices virtual drive help
Load reload step back single step run step delay ms: 200
screen source reset aux vars debug stack flags

```
MOV AL,8
MOV SI, 300H
MOV [SI], 5
DIV [SI]
HLT
```

registers

	H	L
AX	03	01
BX	00	00
CX	00	00
DX	00	00

CS 0100

IP 000A

SS 0100

0100:0008

```
01008:  F6  246  '
01009:  34  052  4
0100A:  F4  244  '
0100B:  90  144  '
0100C:  90  144  '
0100D:  90  144  '
```

0100:000A

```
MOV  AL, 08h
MOV  SI, 00300h
MOV  b.[SI], 05h
DIV  b.[SI]
HLT
...
```

Example (5): Write a program in assembly language to divide AL /contain of memory location 1800H, then put the result in BX, if [1800H] = 7 & AL=56H (using arithmetic instructions).

Solution)

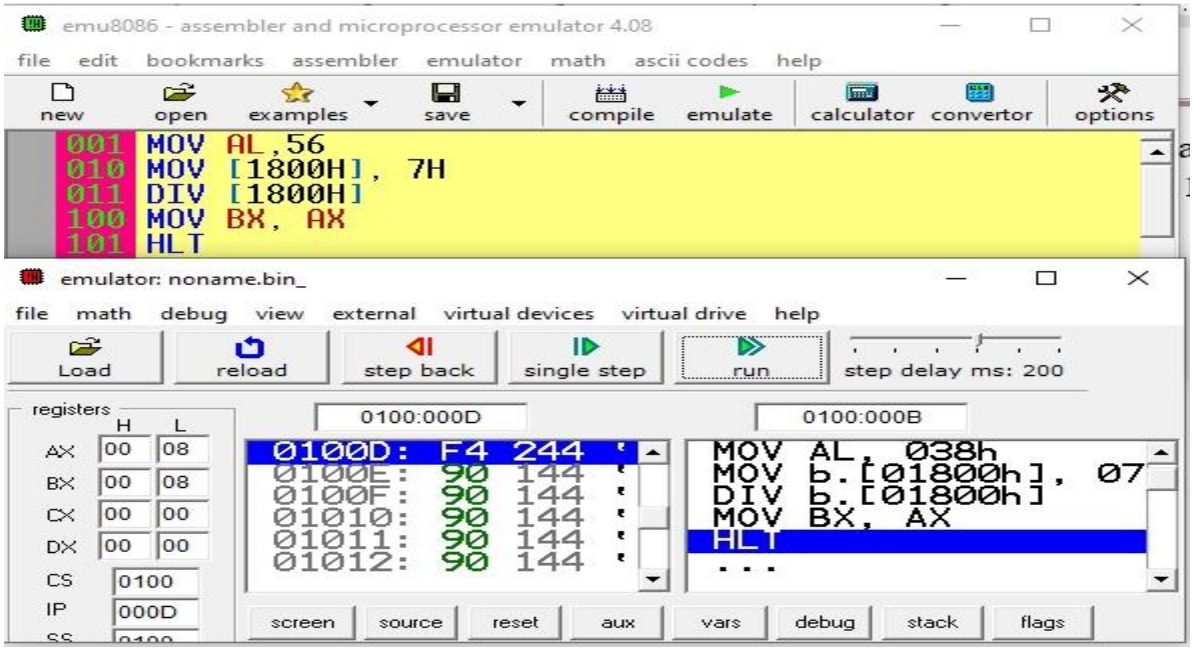
MOV AL, 56H

MOV [1800H], 7

DIV [1800H]

MOV BX, AX

HLT



```
MOV AL,56
MOV [1800H], 7H
DIV [1800H]
MOV BX, AX
HLT
```

step delay ms: 200

registers

	H	L
AX	00	08
BX	00	08
CX	00	00
DX	00	00

CS 0100
IP 000D
SS 0100

0100:000D

```
0100D: F4 244
0100E: 90 144
0100F: 90 144
01010: 90 144
01011: 90 144
01012: 90 144
```

0100:000B

```
MOV AL, 038h
MOV b.[01800h], 07
DIV b.[01800h]
MOV BX, AX
HLT
...
```