The University of Azad Jammu and Kashmir, Muzaffarabad

Name	Kamal Ali Akmal
Course Name	Computer Architecture and Logic Design
Submitted to	Engr. Sidra Rafique
Semester	2nd
Session	2024-2028
Roll No	2024-SE-38
Lab No	08
Submission date	28 August 2025

Addition

Solution:

Decimal form

a) mov ax,2342d

mov bx, 6534d

add ax, bx

Solution

manually,

We have ax=2342d and bx =6534d

First, we have to change the given decimal into binary form binary form,

ax=0000100100100110

bx=0001100110000110

adding both we get, also I used 2's compliment here,

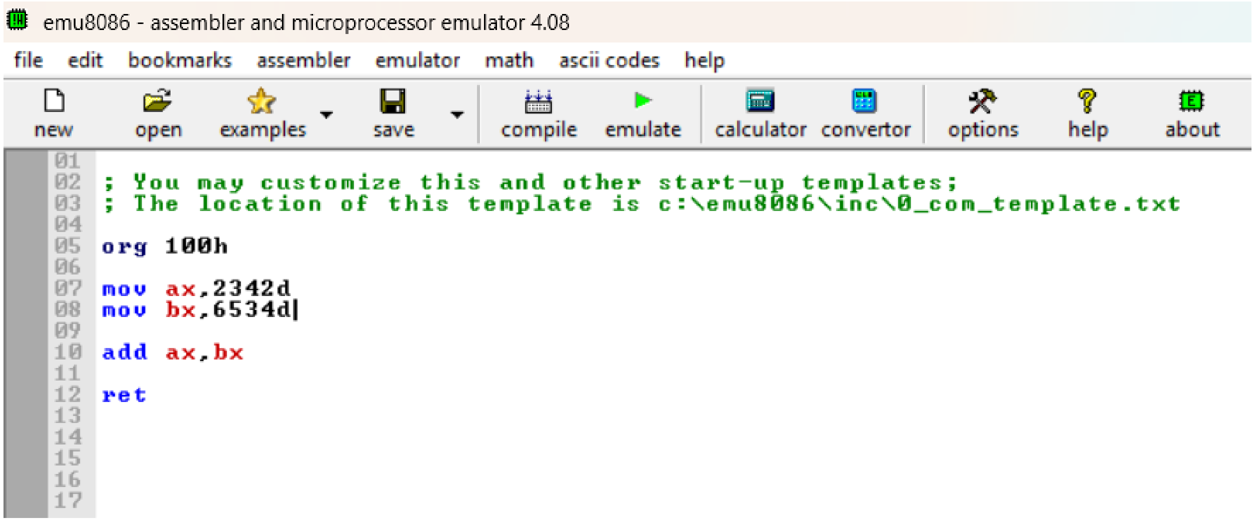
ax=10001010101100

now change it into hexadecimal

we get ax=22AC

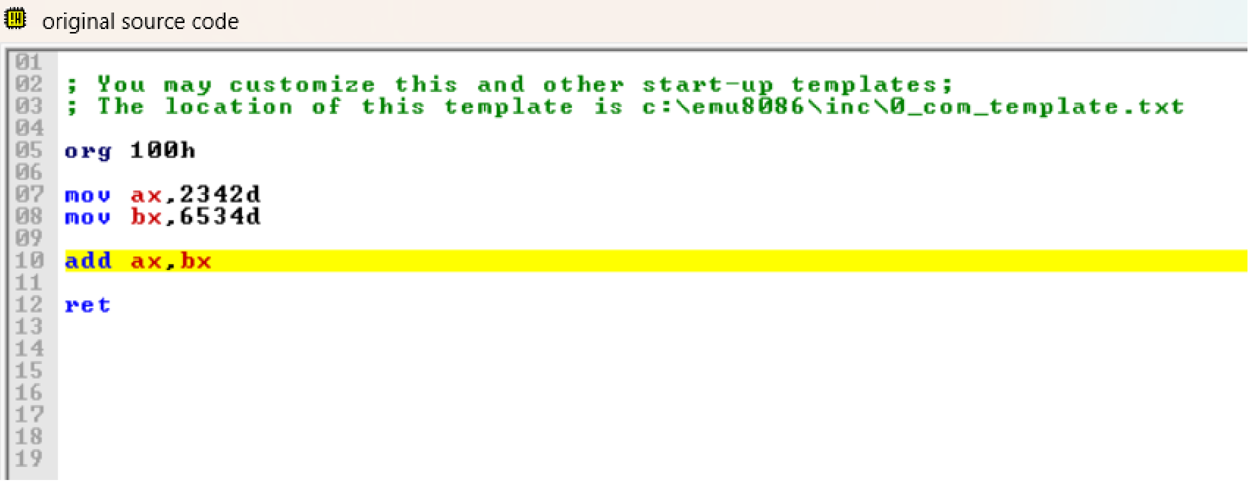
By using emulator,

Step #01:



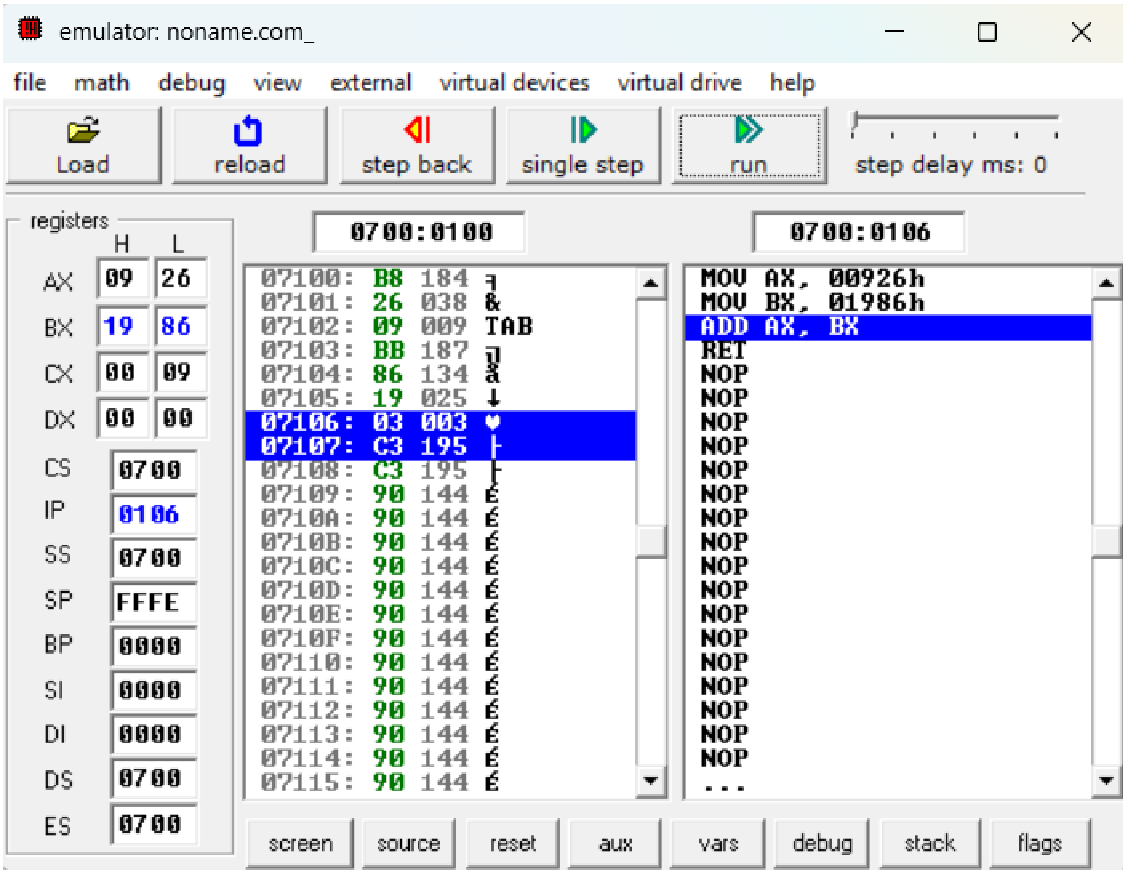
```

; You may customize this and other start-up templates;
; The location of this template is c:\emu8086\inc\0_com_template.txt

org 100h

mov ax,2342d
mov bx,6534d

add ax,bx

ret
```

Source code:



```
; You may customize this and other start-up templates;
; The location of this template is c:\emu8086\inc\0_com_template.txt

org 100h

mov ax,2342d
mov bx,6534d

add ax,bx

ret
```

Binary window:



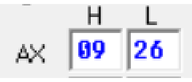
```
registers
     H   L
AX   09  26
BX   19  86
CX   00  09
DX   00  00
CS   0700
IP   0106
SS   0700
SP   FFFE
BP   0000
SI   0000
DI   0000
DS   0700
ES   0700
```

```
0700:0100

07100: B8 184 ╕
07101: 26 038 &
07102: 09 009 TAB
07103: BB 187 ╗
07104: 86 134 å
07105: 19 025 ↓
07106: 03 003 ♥
07107: C3 195 ├
07108: C3 195 ├
07109: 90 144 É
0710A: 90 144 É
0710B: 90 144 É
0710C: 90 144 É
0710D: 90 144 É
0710E: 90 144 É
0710F: 90 144 É
07110: 90 144 É
07111: 90 144 É
07112: 90 144 É
07113: 90 144 É
07114: 90 144 É
07115: 90 144 É
```

```
0700:0106

MOV  AX, 00926h
MOV  BX, 01986h
ADD  AX, BX
RET
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
```

Step #02:

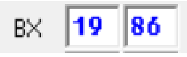
Ax=



```
    H   L
AX  09  26
```

Step #03:

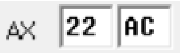
Bx=



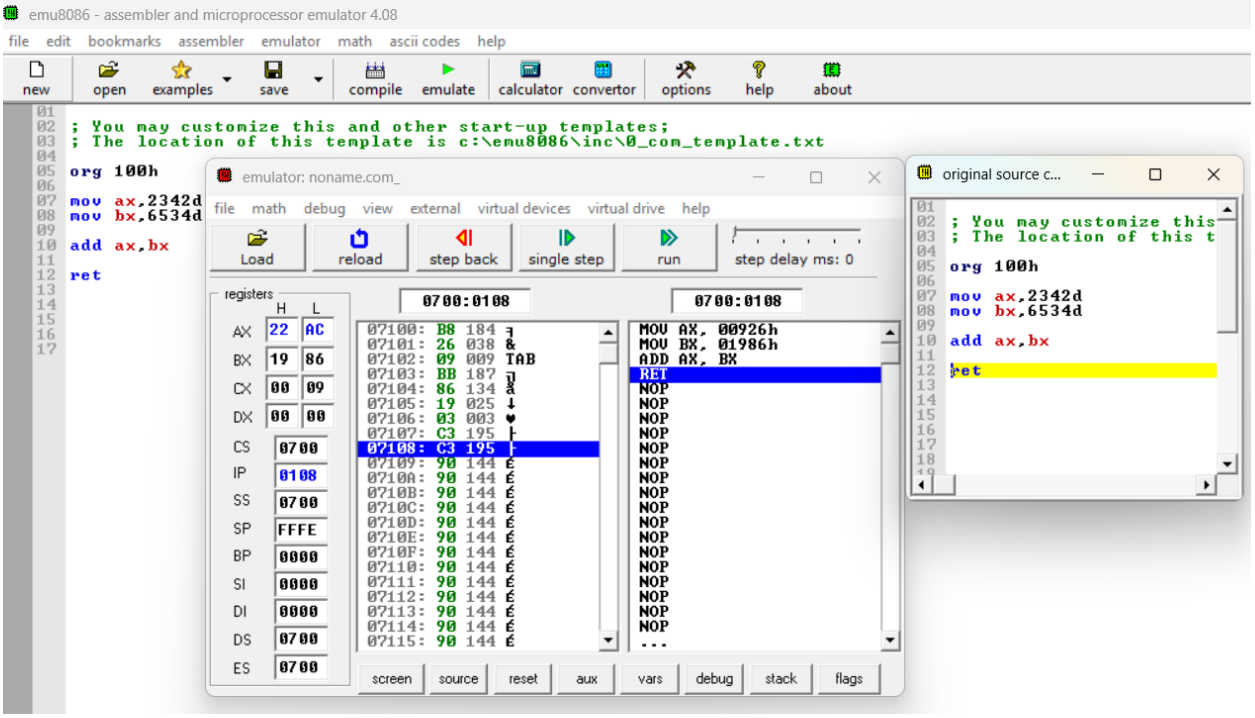
```
BX  19  86
```

Step #04:

Result:



```
AX  22  AC
```



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 help about
emulator: noname.com_
file math debug view external virtual devices virtual drive help
Load reload step back single step run step delay ms: 0
registers
screen source reset aux vars debug stack flags
original source c...

```
; You may customize this and other start-up templates;
; The location of this template is c:\emu8086\inc\0_com_template.txt

org 100h

mov ax,2342d
mov bx,6534d

add ax,bx

ret
```

```
    H   L
AX  22  AC
BX  19  86
CX  00  09
DX  00  00
CS  0700
IP  0108
SS  0700
SP  FFFE
BP  0000
SI  0000
DI  0000
DS  0700
ES  0700
```

```
0700:0108

07100: B8 184 ╕
07101: 26 038 &
07102: 09 009 TAB
07103: BB 187 ╗
07104: 86 134 å
07105: 19 025 ↓
07106: 03 003 ♥
07107: C3 195 ├
07108: C3 195 ├
07109: 90 144 É
0710A: 90 144 É
0710B: 90 144 É
0710C: 90 144 É
0710D: 90 144 É
0710E: 90 144 É
0710F: 90 144 É
07110: 90 144 É
07111: 90 144 É
07112: 90 144 É
07113: 90 144 É
07114: 90 144 É
07115: 90 144 É
```

```
0700:0108

MOV AX, 00926h
MOV BX, 01986h
ADD AX, BX
RET
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
```

```
; You may customize this
; The location of this t

org 100h

mov ax,2342d
mov bx,6534d

add ax,bx

ret
```

Subtraction

Solution:

Decimal form

a) mov ax,374d

mov bx, 64d

add ax, bx

Solution

manually,

We have ax=374d and bx =64d

First, we have to change the given decimal into binary form binary form,

ax= 0000000101110110

bx= 0000000001000000

adding both we get, also I used 2's compliment here,

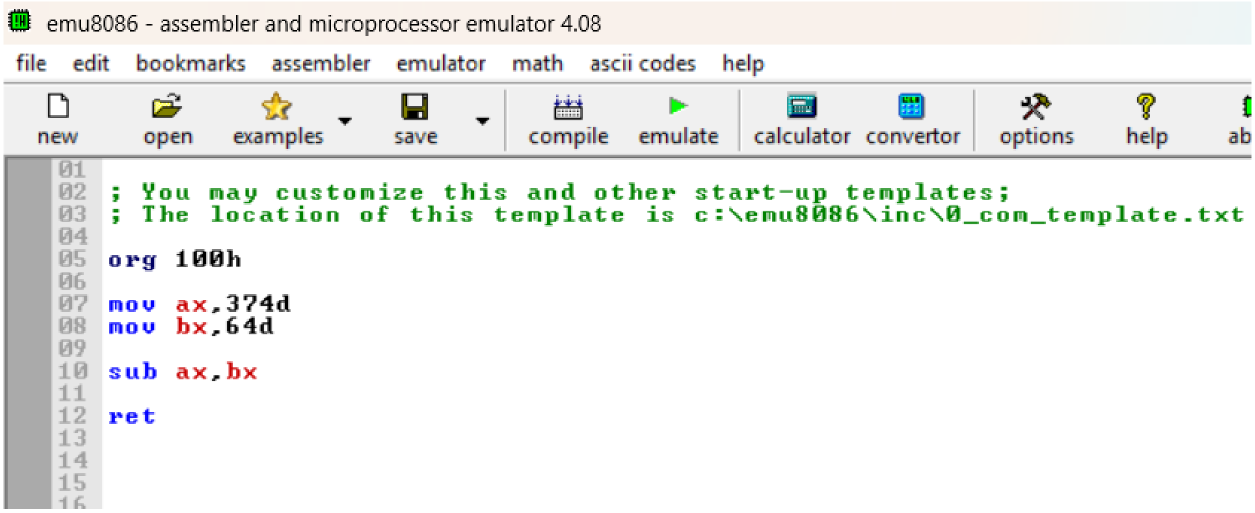
ax=1111111010001001

now change it into hexadecimal

we get ax=FE89

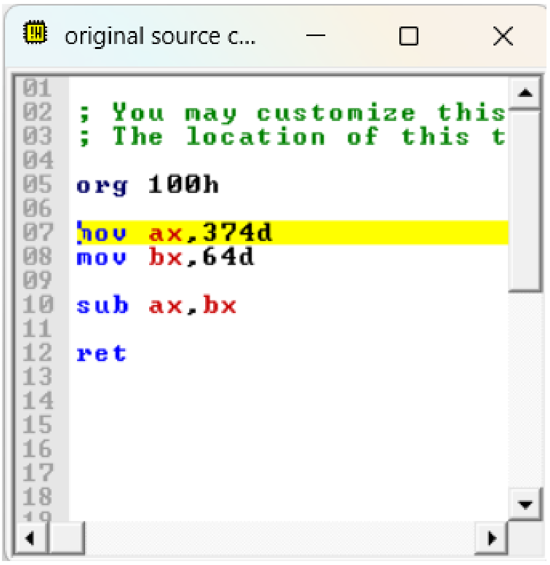
By using emulator,

Step #01:



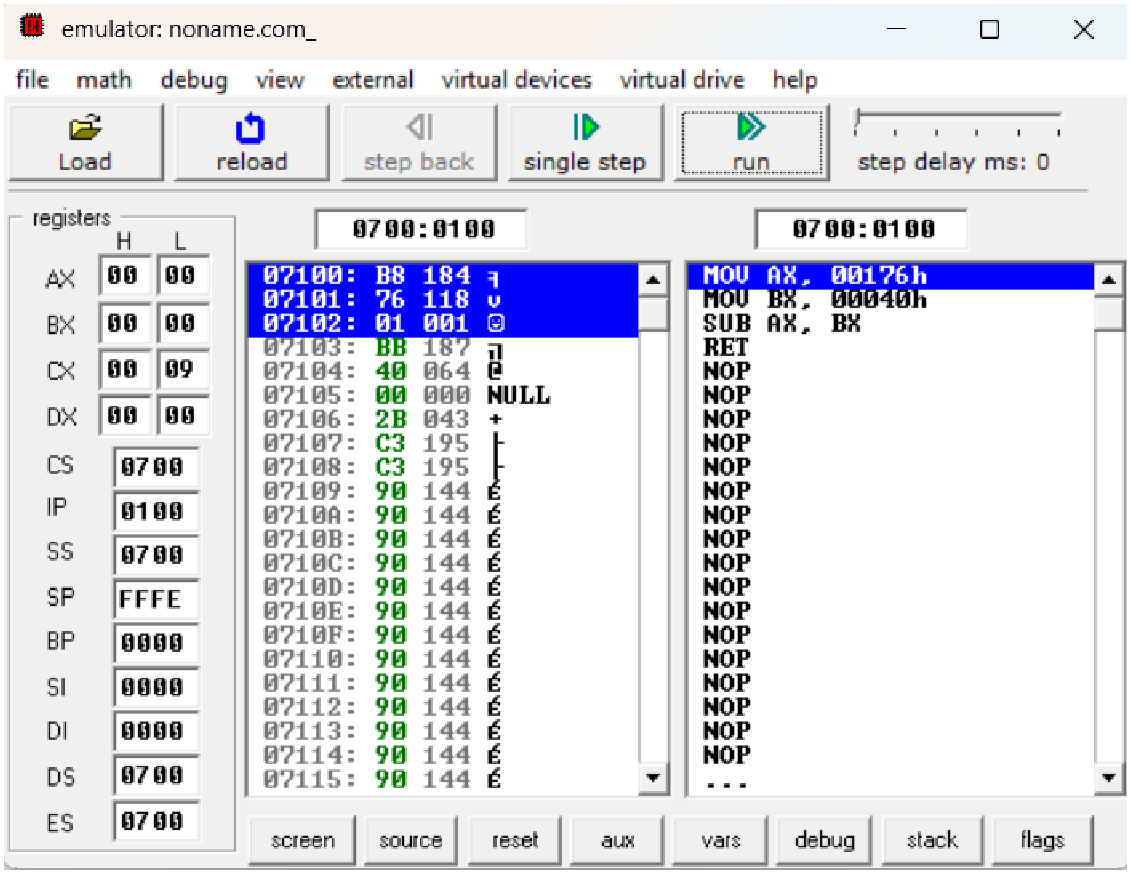
```

; You may customize this and other start-up templates;
; The location of this template is c:\emu8086\inc\0_com_template.txt

org 100h

mov ax,374d
mov bx,64d

sub ax,bx

ret
```

Source code:



```

; You may customize this
; The location of this t

org 100h

mov ax,374d
mov bx,64d

sub ax,bx

ret
```

Binary window:



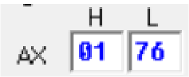
```
registers
     H   L
AX   00  00
BX   00  00
CX   00  09
DX   00  00
CS   0700
IP   0100
SS   0700
SP   FFFE
BP   0000
SI   0000
DI   0000
DS   0700
ES   0700
```

```
0700:0100

07100: B8 184 ╕
07101: 76 118 v
07102: 01 001 ☺
07103: BB 187 ╗
07104: 40 064 @
07105: 00 000 NULL
07106: 2B 043 +
07107: C3 195 ├
07108: C3 195 ├
07109: 90 144 É
0710A: 90 144 É
0710B: 90 144 É
0710C: 90 144 É
0710D: 90 144 É
0710E: 90 144 É
0710F: 90 144 É
07110: 90 144 É
07111: 90 144 É
07112: 90 144 É
07113: 90 144 É
07114: 90 144 É
07115: 90 144 É
```

```
0700:0100

MOV  AX, 00176h
MOV  BX, 00040h
SUB  AX, BX
RET
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
```

Step #02:

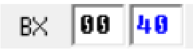
Ax=



	H	L
AX	01	76

Step #03:

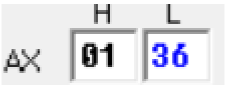
Bx=



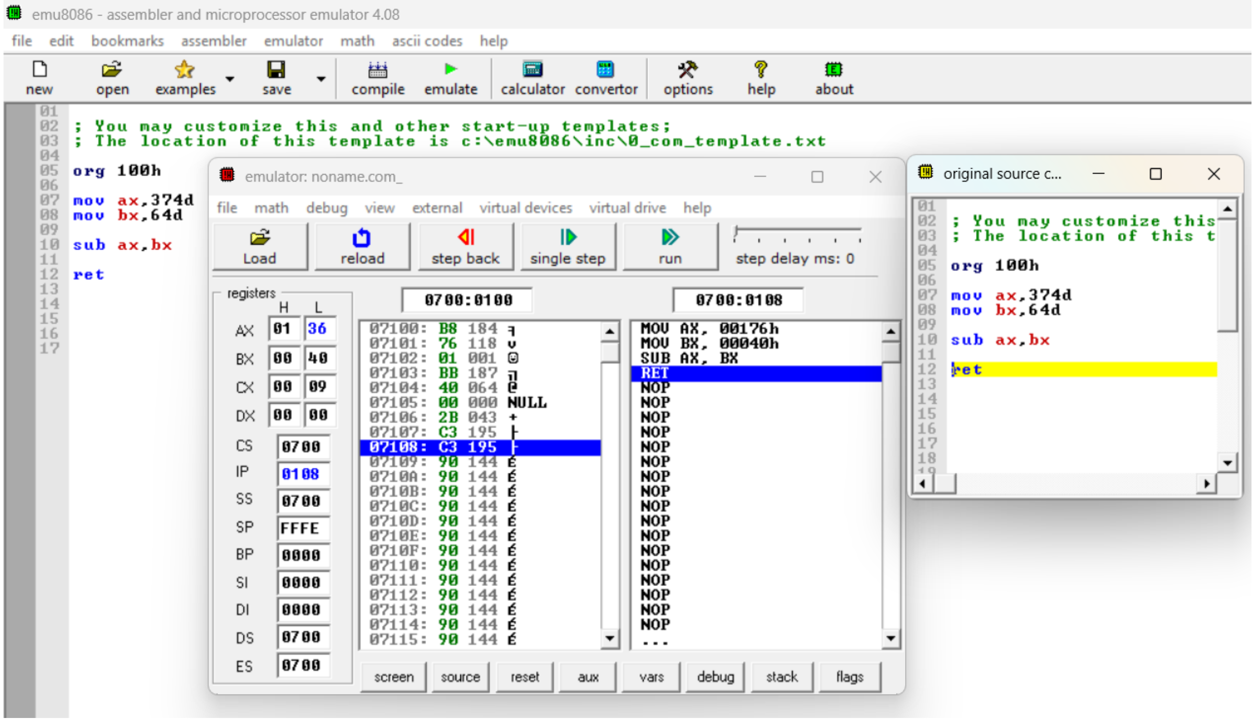
BX	00	40

Step #04:

Result:



	H	L
AX	01	36



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 help about
emulator: noname.com_
file math debug view external virtual devices virtual drive help
Load reload step back single step run step delay ms: 0
screen source reset aux vars debug stack flags
original source c...

```
; You may customize this and other start-up templates;
; The location of this template is c:\emu8086\inc\0_com_template.txt

org 100h

mov ax,374d
mov bx,64d

sub ax,bx

ret
```

registers

	H	L
AX	01	36
BX	00	40
CX	00	09
DX	00	00

CS	0700
IP	0108
SS	0700
SP	FFFE
BP	0000
SI	0000
DI	0000
DS	0700
ES	0700

```
0700:0100

07100: B8 184 ╕
07101: 76 118 v
07102: 01 001 ☺
07103: BB 187 ╗
07104: 40 064 @
07105: 00 000 NULL
07106: 2B 043 +
07107: C3 195 ├
07108: C3 195 ├
07109: 90 144 É
0710A: 90 144 É
0710B: 90 144 É
0710C: 90 144 É
0710D: 90 144 É
0710E: 90 144 É
0710F: 90 144 É
07110: 90 144 É
07111: 90 144 É
07112: 90 144 É
07113: 90 144 É
07114: 90 144 É
07115: 90 144 É
```

```
0700:0108

MOV AX, 00176h
MOV BX, 00040h
SUB AX, BX
RET
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
```

Multiplication

Solution:

Decimal form

a) mov ax,34d

mov bx, 47d

add ax, bx

Solution

manually,

We have ax=34d and bx =47d

First, we have to change the given decimal into binary form binary form,

ax=00100010

bx=00101111

adding both we get, also I used 2's compliment here,

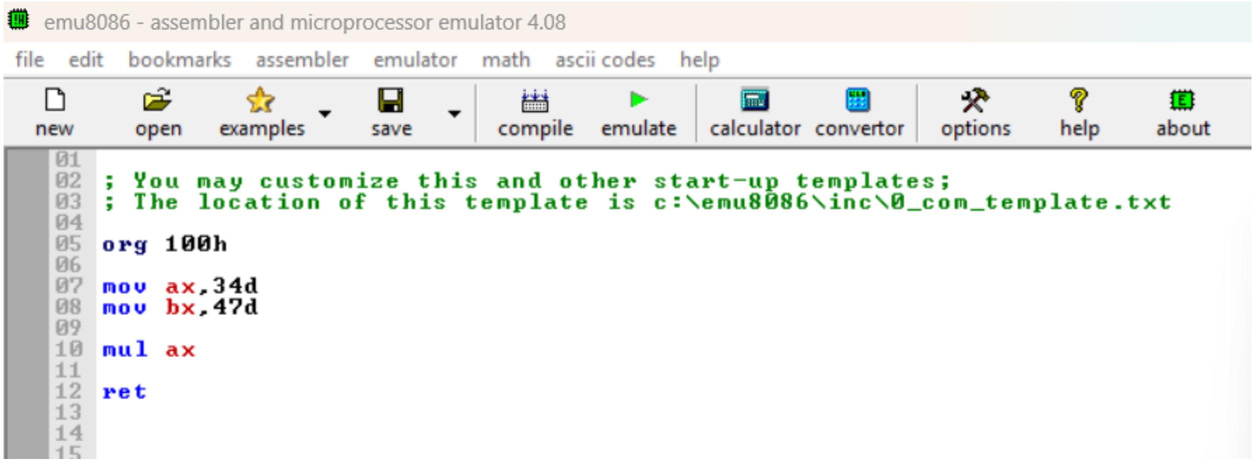
ax=11011101

now change it into hexadecimal

we get ax=DD

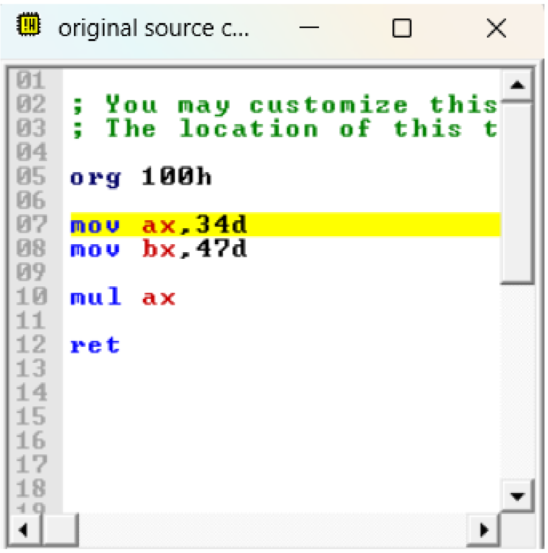
By using emulator,

Step #01:



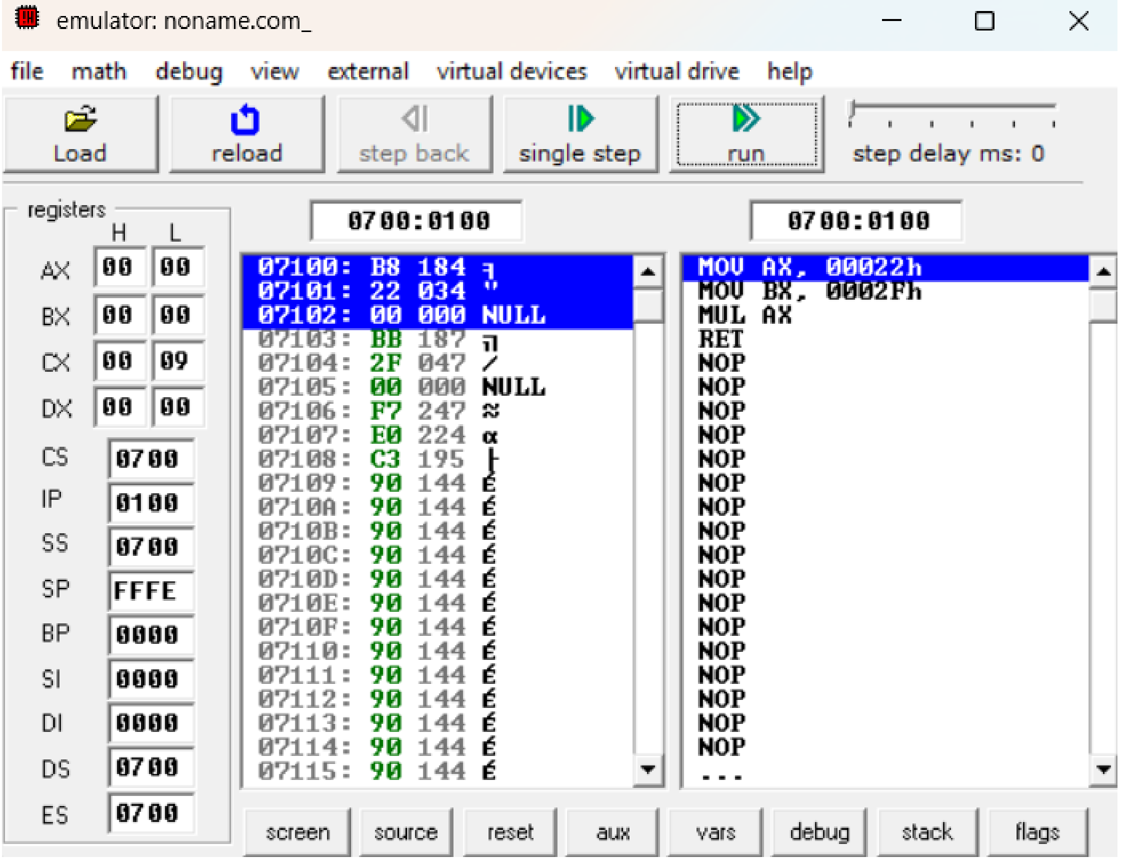
```
; You may customize this and other start-up templates;
; The location of this template is c:\emu8086\inc\0_com_template.txt

org 100h

mov ax,34d
mov bx,47d

mul ax

ret
```

Source code:



```
; You may customize this
; The location of this t

org 100h

mov  ax,34d
mov  bx,47d

mul  ax

ret
```

Binary window:



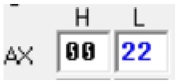
```
registers
      H    L
AX    00   00
BX    00   00
CX    00   09
DX    00   00
CS    0700
IP    0100
SS    0700
SP    FFFE
BP    0000
SI    0000
DI    0000
DS    0700
ES    0700
```

```
0700:0100

07100:  B8 184 ╕
07101:  22 034 "
07102:  00 000 NULL
07103:  BB 187 ╗
07104:  2F 047 /
07105:  00 000 NULL
07106:  F7 247 ≈
07107:  E0 224 α
07108:  C3 195 ├
07109:  90 144 É
0710A:  90 144 É
0710B:  90 144 É
0710C:  90 144 É
0710D:  90 144 É
0710E:  90 144 É
0710F:  90 144 É
07110:  90 144 É
07111:  90 144 É
07112:  90 144 É
07113:  90 144 É
07114:  90 144 É
07115:  90 144 É
```

```
0700:0100

MOV  AX, 00022h
MOV  BX, 0002Fh
MUL  AX
RET
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
```

Step #02:

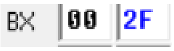
Ax=



```
     H    L
AX  00   22
```

Step #03:

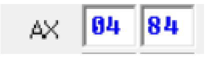
Bx=



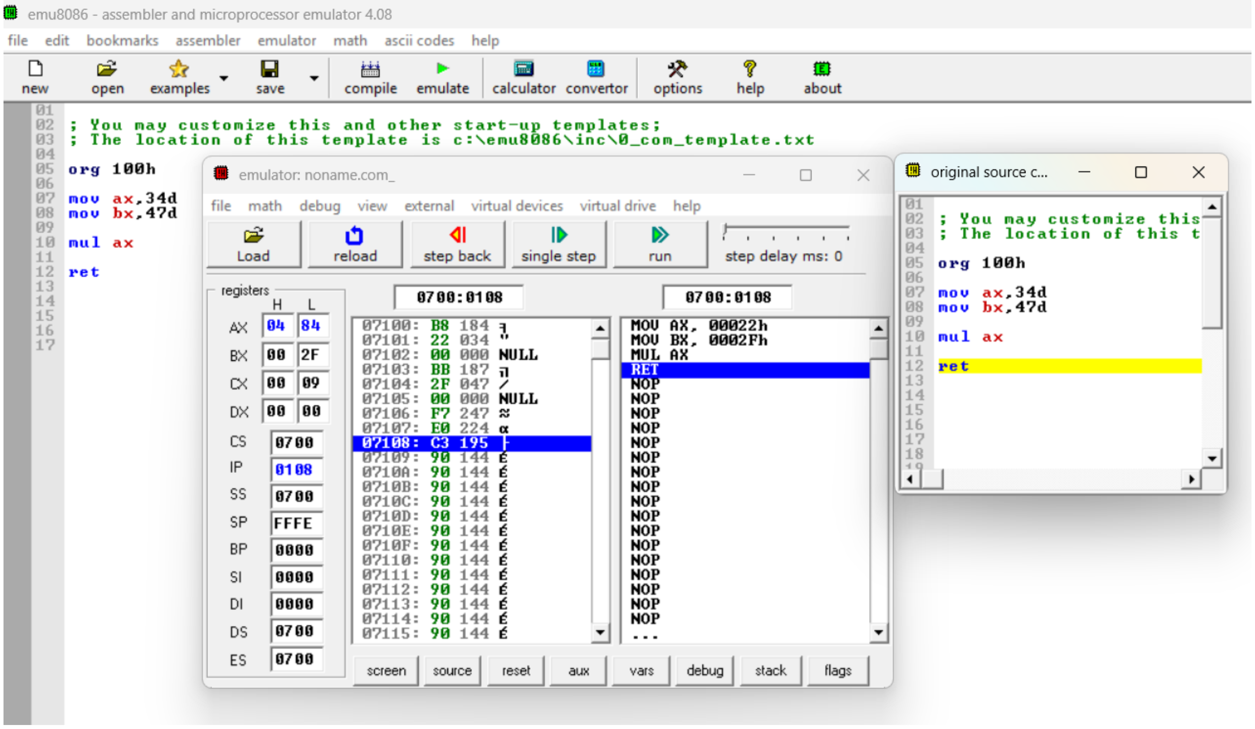
```
BX  00   2F
```

Step #04:

Result:



```
AX  04   84
```



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 help about
emulator: noname.com_
file math debug view external virtual devices virtual drive help
Load reload step back single step run step delay ms: 0
original source c...
screen source reset aux vars debug stack flags

```
; You may customize this and other start-up templates;
; The location of this template is c:\emu8086\inc\0_com_template.txt

org 100h

mov ax,34d
mov bx,47d

mul ax

ret
```

```
registers
     H   L
AX  04  84
BX  00  2F
CX  00  09
DX  00  00
CS  0700
IP  0108
SS  0700
SP  FFFE
BP  0000
SI  0000
DI  0000
DS  0700
ES  0700
```

```
0700:0108

07100: B8 184 ╕
07101: 22 034 "
07102: 00 000 NULL
07103: BB 187 ╗
07104: 2F 047 /
07105: 00 000 NULL
07106: F7 247 ≈
07107: E0 224 α
07108: C3 195 ├
07109: 90 144 É
0710A: 90 144 É
0710B: 90 144 É
0710C: 90 144 É
0710D: 90 144 É
0710E: 90 144 É
0710F: 90 144 É
07110: 90 144 É
07111: 90 144 É
07112: 90 144 É
07113: 90 144 É
07114: 90 144 É
07115: 90 144 É
```

```
0700:0108

MOV AX, 00022h
MOV BX, 0002Fh
MUL AX
RET
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
```

```
; You may customize this
; The location of this t

org 100h

mov ax,34d
mov bx,47d

mul ax

ret
```

Division

Solution:

Decimal form

a) mov ax,5000d

mov bx, 50d

add ax, bx

Solution

manually,

We have ax=5000d and bx =50d

First, we have to change the given decimal into binary form binary form,

ax=1100001101010000

bx=0000000000110010

adding both we get, also I used 2's compliment here,

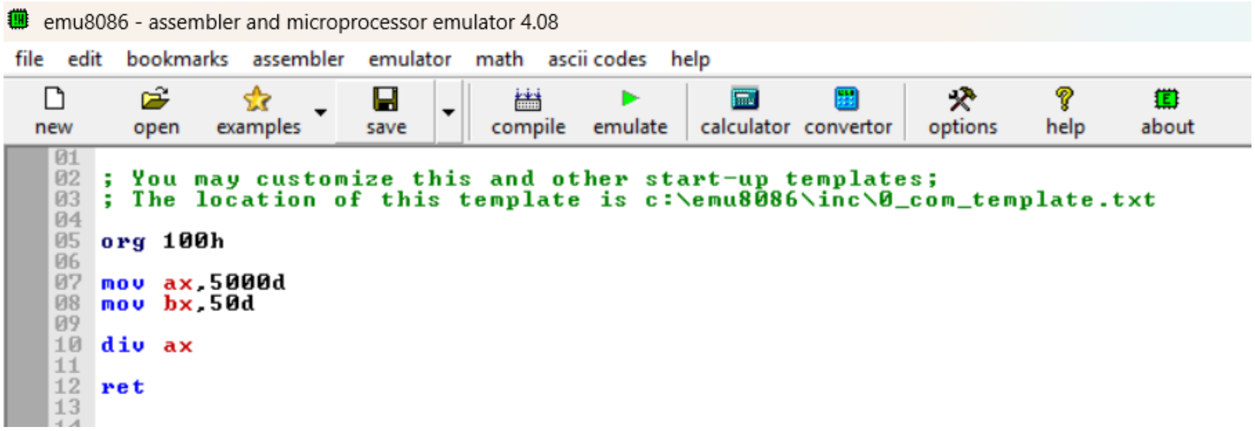
ax=0011110010101111

now change it into hexadecimal

we get ax=3CAF

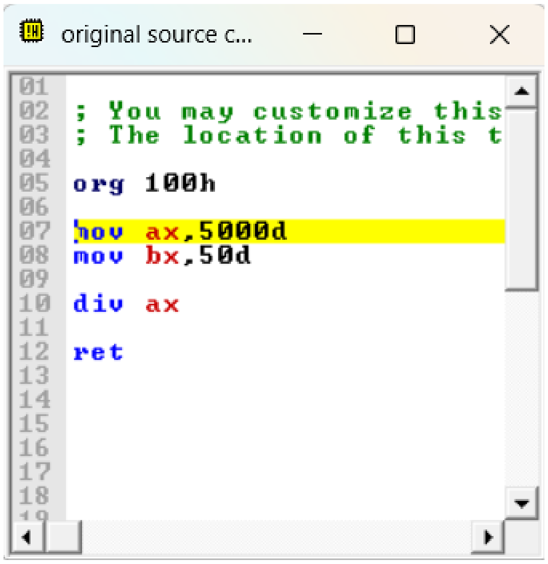
By using emulator,

Step #01:



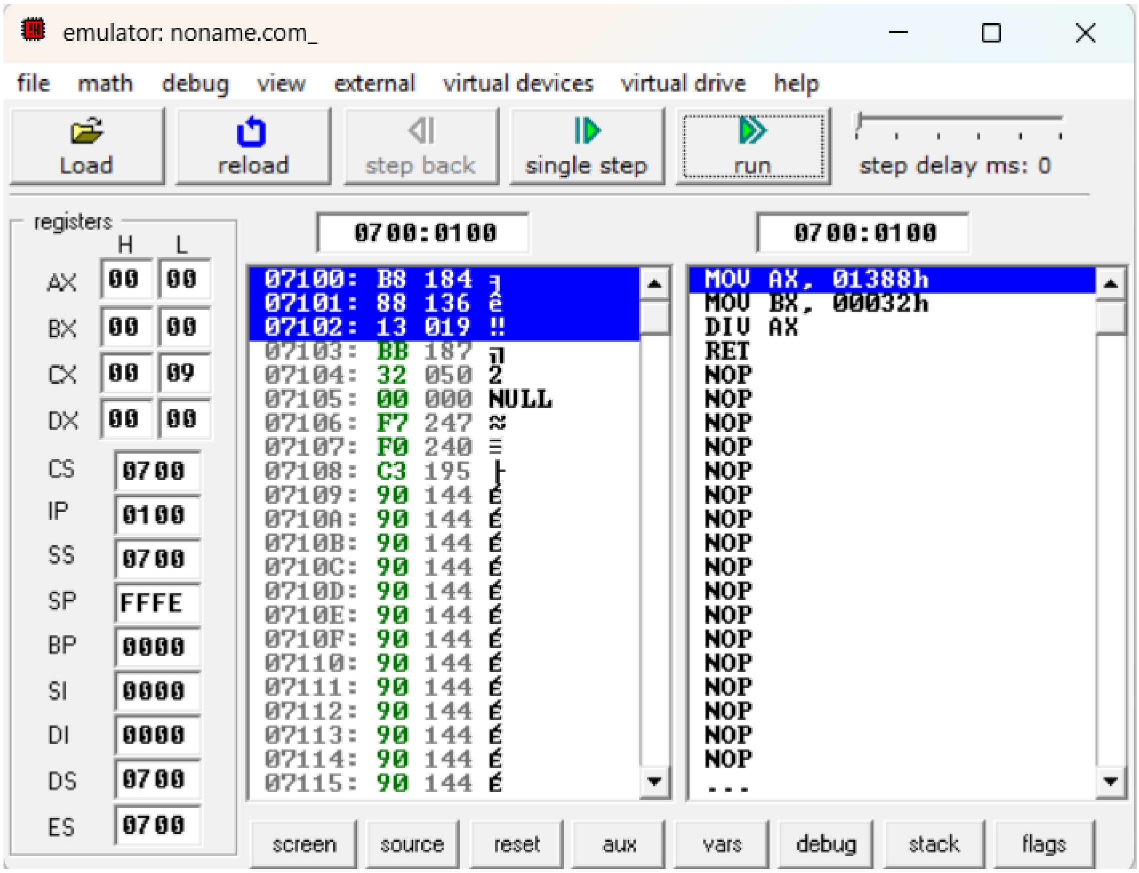
```
; You may customize this and other start-up templates;
; The location of this template is c:\emu8086\inc\0_com_template.txt

org 100h

mov  ax,5000d
mov  bx,50d

div  ax

ret
```

Source code:



```

; You may customize this
; The location of this t

org 100h

mov ax,5000d
mov bx,50d

div ax

ret
```

Binary window:



registers

	H	L
AX	00	00
BX	00	00
CX	00	09
DX	00	00

CS	0700
IP	0100
SS	0700
SP	FFFE
BP	0000
SI	0000
DI	0000
DS	0700
ES	0700

0700:0100

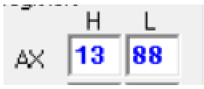
```
07100: B8 184 ╕
07101: 88 136 ê
07102: 13 019 ‼
07103: BB 187 ╗
07104: 32 050 2
07105: 00 000 NULL
07106: F7 247 ≈
07107: F0 240 ≡
07108: C3 195 ├
07109: 90 144 É
0710A: 90 144 É
0710B: 90 144 É
0710C: 90 144 É
0710D: 90 144 É
0710E: 90 144 É
0710F: 90 144 É
07110: 90 144 É
07111: 90 144 É
07112: 90 144 É
07113: 90 144 É
07114: 90 144 É
07115: 90 144 É
```

0700:0100

```
MOV  AX, 01388h
MOV  BX, 00032h
DIV  AX
RET
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
```

Step #02:

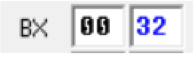
Ax=



```
    H   L
AX  13  88
```

Step #03:

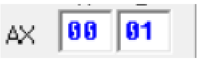
Bx=



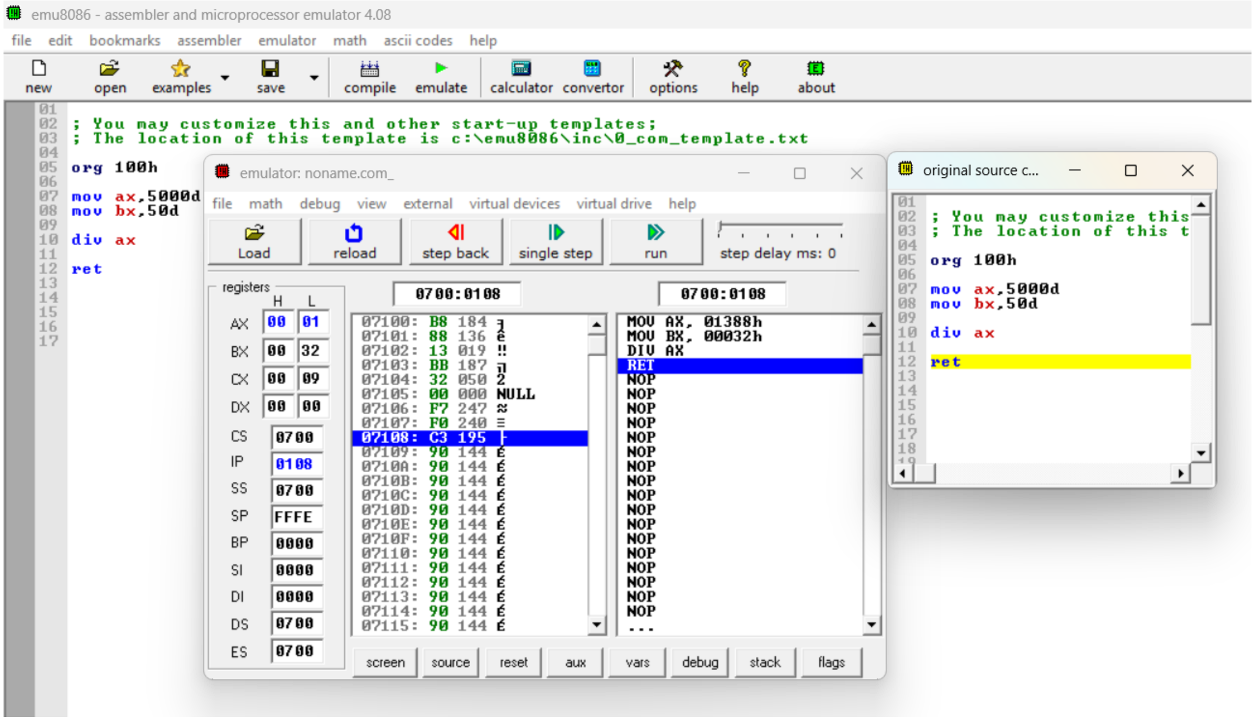
```
BX  00  32
```

Step #04:

Result:



```
AX  00  01
```



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 help about

```
; You may customize this and other start-up templates;
; The location of this template is c:\emu8086\inc\0_com_template.txt

org 100h

mov ax,5000d
mov bx,50d

div ax

ret
```

emulator: noname.com_
file math debug view external virtual devices virtual drive help
Load reload step back single step run step delay ms: 0
registers
screen source reset aux vars debug stack flags

```
registers
    H   L
AX  00  01
BX  00  32
CX  00  09
DX  00  00
CS  0700
IP  0108
SS  0700
SP  FFFE
BP  0000
SI  0000
DI  0000
DS  0700
ES  0700
```

```
0700:0108

07100: B8 184 ╕
07101: 88 136 ê
07102: 13 019 ‼
07103: BB 187 ╗
07104: 32 050 2
07105: 00 000 NULL
07106: F7 247 ≈
07107: F0 240 ≡
07108: C3 195 ├
07109: 90 144 É
0710A: 90 144 É
0710B: 90 144 É
0710C: 90 144 É
0710D: 90 144 É
0710E: 90 144 É
0710F: 90 144 É
07110: 90 144 É
07111: 90 144 É
07112: 90 144 É
07113: 90 144 É
07114: 90 144 É
07115: 90 144 É
```

```
0700:0108

MOV AX, 01388h
MOV BX, 00032h
DIV AX
RET
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
```

```
; You may customize this
; The location of this t

org 100h

mov ax,5000d
mov bx,50d

div ax

ret
```