



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,

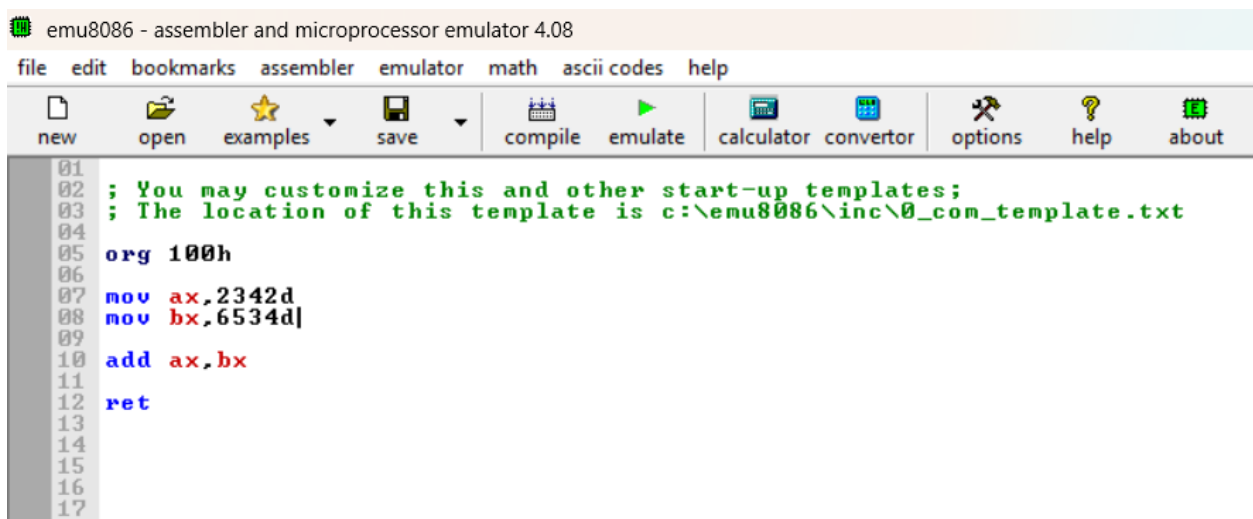
ax=10001010101100

now change it into hexadecimal

we get ax=22AC

By using emulator,

Step #01:



```
01
02 ; You may customize this and other start-up templates;
03 ; The location of this template is c:\emu8086\inc\0_com_template.txt
04
05 org 100h
06
07 mov ax,2342d
08 mov bx,6534d
09
10 add ax,bx
11
12 ret
13
14
15
16
17
```

 original source code

Binary window:

[illegible]

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

```
01 ; You may customize this and other start-up templates;
02 ; The location of this template is c:\emu8086\inc\0_com_template.txt
03
04
05 org 100h
06
07 mov ax,2342d
08 mov bx,6534d
09
10 add ax,bx
11
12 ret
13
14
15
16
17
```

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	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		0700:0108	
07100:	B8 184 1	MOV AX, 00926h	
07101:	26 038 &	MOV BX, 01986h	
07102:	09 009 TAB	ADD AX, BX	
07103:	BB 187 1	RET	
07104:	86 134 1	NOP	
07105:	19 025 1	NOP	
07106:	03 003 1	NOP	
07107:	C3 195 1	NOP	
07108:	C3 195 1	NOP	
07109:	90 144 E	NOP	
0710A:	90 144 E	NOP	
0710B:	90 144 E	NOP	
0710C:	90 144 E	NOP	
0710D:	90 144 E	NOP	
0710E:	90 144 E	NOP	
0710F:	90 144 E	NOP	
07110:	90 144 E	NOP	
07111:	90 144 E	NOP	
07112:	90 144 E	NOP	
07113:	90 144 E	NOP	
07114:	90 144 E	NOP	
07115:	90 144 E	...	

original source c...

```
01 ; You may customize this
02 ; The location of this t
03
04
05 org 100h
06
07 mov ax,2342d
08 mov bx,6534d
09
10 add ax,bx
11
12 ret
13
14
15
16
17
```

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= 00000000101110110

bx= 0000000001000000

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

ax=1111111010001001

now change it into hexadecimal

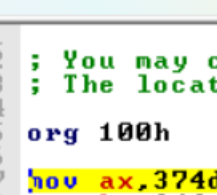
we get ax=FE89

By using emulator,

Step #01:

```
01
02 ; You may customize this and other start-up templates;
03 ; The location of this template is c:\emu8086\inc\0_com_template.txt
04
05 org 100h
06
07 mov ax,374d
08 mov bx,64d
09
10 sub ax,bx
11
12 ret
13
14
15
16
```

Source code:

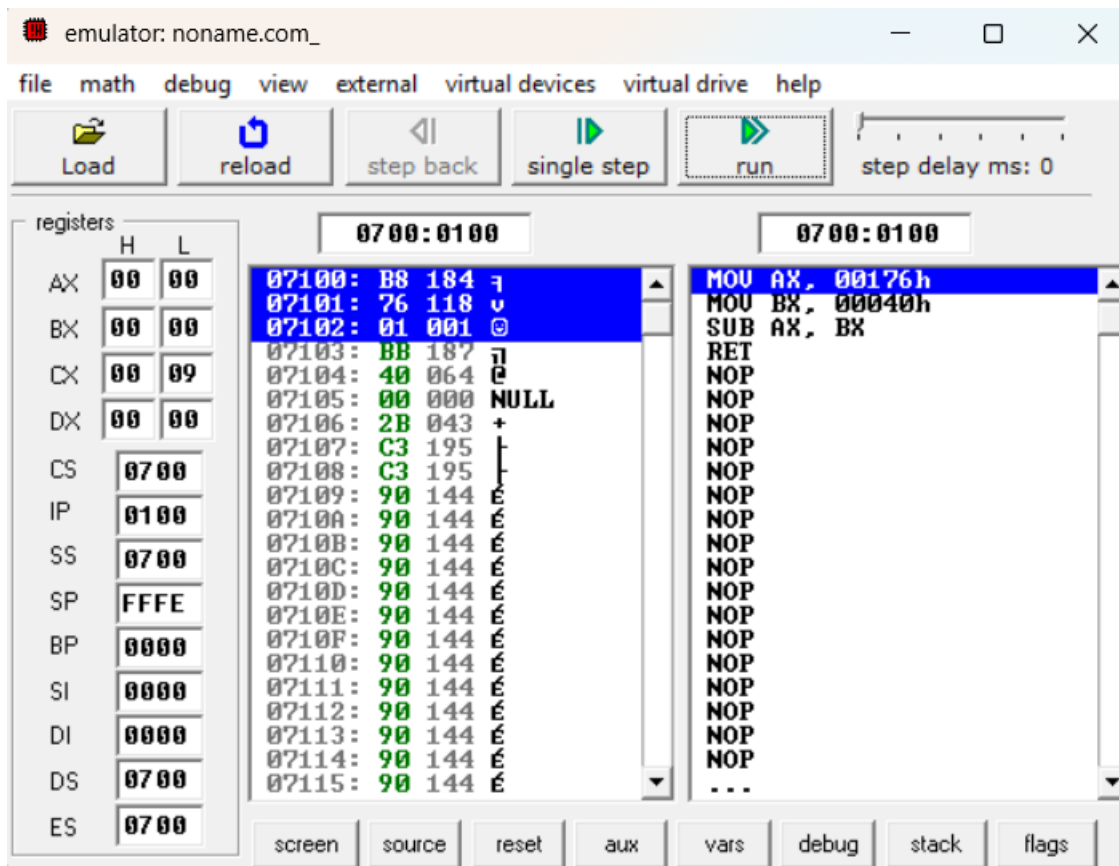


```

01
02 ; You may customize this
03 ; The location of this t
04
05 org 100h
06
07 mov ax, 374d
08 mov bx, 64d
09
10 sub ax, bx
11
12 ret
13
14
15
16
17
18
19
20

```

Binary window:



Step #02:

Ax=

	H	L
AX	01	76

Step #03:

Bx=

	H	L
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

```
01 ; You may customize this and other start-up templates;
02 ; The location of this template is c:\emu8086\inc\0_com_template.txt
03
04 org 100h
05
06
07 mov ax,374d
08 mov bx,64d
09
10 sub ax,bx
11
12 ret
13
14
15
16
17
```

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	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		0700:0108	
07100:	B8 184 3	MOV AX, 00176h	
07101:	76 118 v	MOV BX, 00040h	
07102:	01 001 0	SUB AX, BX	
07103:	BB 187 7	ret	
07104:	40 064 0	NOP	
07105:	00 000 NULL	NOP	
07106:	2B 043 +	NOP	
07107:	C3 195	NOP	
07108:	C3 195	NOP	
07109:	90 144 E	NOP	
0710A:	90 144 E	NOP	
0710B:	90 144 E	NOP	
0710C:	90 144 E	NOP	
0710D:	90 144 E	NOP	
0710E:	90 144 E	NOP	
0710F:	90 144 E	NOP	
07110:	90 144 E	NOP	
07111:	90 144 E	NOP	
07112:	90 144 E	NOP	
07113:	90 144 E	NOP	
07114:	90 144 E	NOP	
07115:	90 144 E	...	

screen source reset aux vars debug stack flags

original source c...

```
01 ; You may customize this
02 ; The location of this t
03
04 org 100h
05
06
07 mov ax,374d
08 mov bx,64d
09
10 sub ax,bx
11
12 ret
13
14
15
16
17
18
19
```

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,

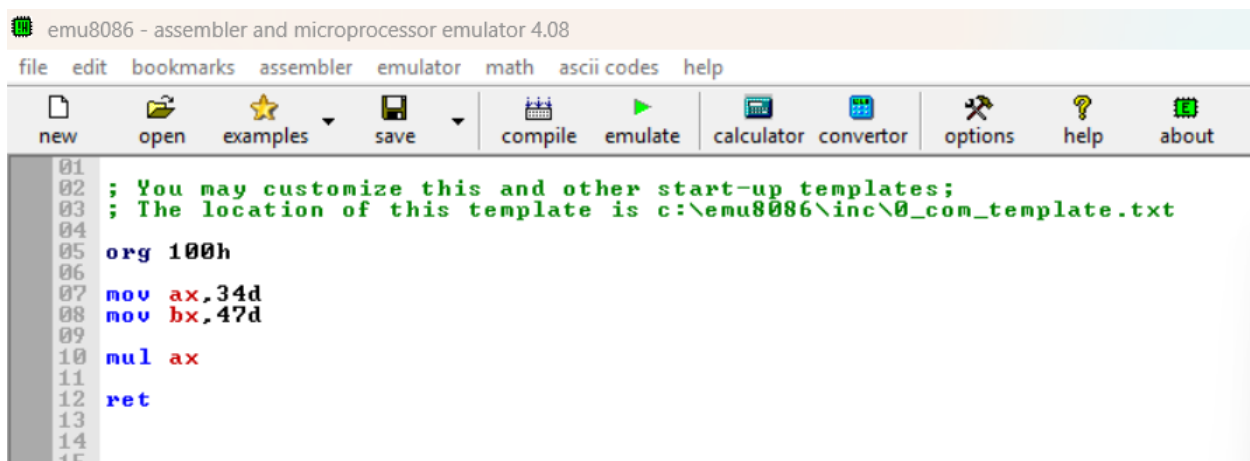
ax=11011101

now change it into hexadecimal

we get ax=DD

By using emulator,

Step #01:



```
01 ; You may customize this and other start-up templates;
02 ; The location of this template is c:\emu8086\inc\0_com_template.txt
03
04 org 100h
05
06 mov ax,34d
07 mov bx,47d
08
09 mul ax
10
11
12 ret
13
14
15
```


Source code:

```
original source c...
01
02 ; You may customize this
03 ; The location of this t
04
05 org 100h
06
07 mov ax, 34d
08 mov bx, 47d
09
10 mul ax
11
12 ret
13
14
15
16
17
18
19
```

Binary window:

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		0700:0100		0700:0100	
	H	L			
AX	00	00	07100: B8 184	MOV AX, 00022h	
BX	00	00	07101: 22 034	MOV BX, 0002Fh	
CX	00	09	07102: 00 000 NULL	MUL AX	
DX	00	00	07103: BB 187	RET	
CS	0700		07104: 2F 047	NOP	
IP	0100		07105: 00 000 NULL	NOP	
SS	0700		07106: F7 247	NOP	
SP	FFFE		07107: E0 224	NOP	
BP	0000		07108: C3 195	NOP	
SI	0000		07109: 90 144	NOP	
DI	0000		0710A: 90 144	NOP	
DS	0700		0710B: 90 144	NOP	
ES	0700		0710C: 90 144	NOP	
			0710D: 90 144	NOP	
			0710E: 90 144	NOP	
			0710F: 90 144	NOP	
			07110: 90 144	NOP	
			07111: 90 144	NOP	
			07112: 90 144	NOP	
			07113: 90 144	NOP	
			07114: 90 144	NOP	
			07115: 90 144	...	

screen source reset aux vars debug stack flags

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

```
01 ; You may customize this and other start-up templates;
02 ; The location of this template is c:\emu8086\inc\0_com_template.txt
03
04 org 100h
05
06 mov ax,34d
07 mov bx,47d
08
09 mul ax
10
11 ret
12
13
14
15
16
17
```

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	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 0700:0108

Address	Hex	Dec	Comment
07100:	B8	184	MOV AX, 00022h
07101:	22	034	MOV BX, 0002Fh
07102:	00	000	MUL AX
07103:	BB	187	RET
07104:	2F	047	NOP
07105:	00	000	NOP
07106:	F7	247	NOP
07107:	E0	224	NOP
07108:	C3	195	NOP
07109:	90	144	NOP
0710A:	90	144	NOP
0710B:	90	144	NOP
0710C:	90	144	NOP
0710D:	90	144	NOP
0710E:	90	144	NOP
0710F:	90	144	NOP
07110:	90	144	NOP
07111:	90	144	NOP
07112:	90	144	NOP
07113:	90	144	NOP
07114:	90	144	NOP
07115:	90	144	NOP

original source c...

```
01 ; You may customize this
02 ; The location of this t
03
04 org 100h
05
06 mov ax,34d
07 mov bx,47d
08
09 mul ax
10
11 ret
12
13
14
15
16
17
```

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,

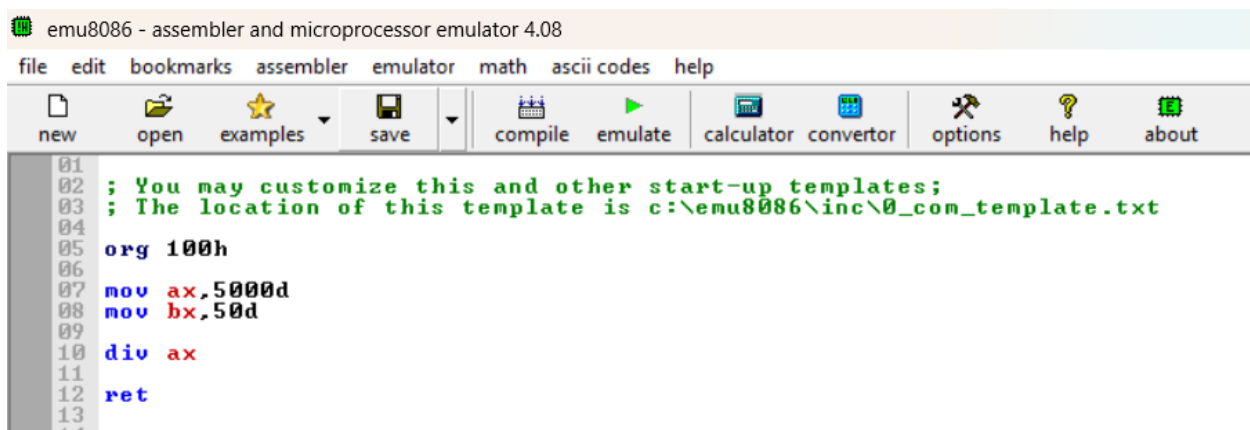
ax=0011110010101111

now change it into hexadecimal

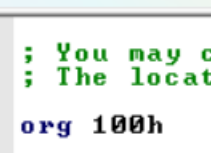
we get ax=3CAF

By using emulator,

Step #01:



```
01 ; You may customize this and other start-up templates;
02 ; The location of this template is c:\emu8086\inc\0_com_template.txt
03
04 org 100h
05
06
07 mov ax,5000d
08 mov bx,50d
09
10 div ax
11
12 ret
13
14
```



```

01
02 ; You may customize this
03 ; The location of this t
04
05 org 100h
06
07 mov ax, 5000d
08 mov bx, 50d
09
10 div ax
11
12 ret
13
14
15
16
17
18
19
20

```

The screenshot shows the x86-64 emulator interface. At the top, the title bar reads "emulator: noname.com_". Below it is a menu bar with "file", "math", "debug", "view", "external", "virtual devices", "virtual drive", and "help". The toolbar contains icons for "Load", "reload", "step back", "single step", "run", and a "step delay ms: 0" slider.

The "registers" section on the left displays the following values:

	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	

The main display area is divided into two panes. The left pane shows memory addresses from 07100 to 07115 with their corresponding hex values and ASCII representations:

Address	Hex	ASCII
07100	B8 184	j
07101	88 136	e
07102	13 019	!!
07103	BB 187	ï
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

The right pane shows the instruction stream:

Address	Instruction
07100	MOV AX, 01388h
07101	MOV BX, 00032h
07102	DIU AX
07103	RET
07104	NOP
07105	NOP
07106	NOP
07107	NOP
07108	NOP
07109	NOP
0710A	NOP
0710B	NOP
0710C	NOP
0710D	NOP
0710E	NOP
0710F	NOP
07110	NOP
07111	NOP
07112	NOP
07113	NOP
07114	NOP
07115	...

At the bottom, there is a row of buttons: "screen", "source", "reset", "aux", "vars", "debug", "stack", and "flags".

Step #02:

$$Ax =$$

	H	L
AX	13	88

Step #03:

$$Bx =$$
BX 00 32

Step #04:

Result:

AX 00 01

