



UNIVERSITY OF ASIA PACIFIC

Department of Computer Science & Engineering

LAB ASSIGNMENT-02

Course Title : Microprocessors and Assembly Language Lab

Course Code : CSE 312

Date of Submission: 29/08/2022

Submitted by:

Name : Sheikh Nafez Sadnan

Reg. No. : 20101106

Roll No. : 106

Section : B₍₂₎

Submitted To:

Faria Zarin Subah

Lecturer

Department of CSE

University of Asia Pacific

Problem Statement 1:

Perform the following operations using emu8086:

- a) MOV CX, AX
- b) MOV DX, BX
- c) ADD CX, DX
- d) SUB DX, CX
- e) INC AX
- f) DEC BX
- g) NEG DX

Initially, AX = last 4 digits of your ID

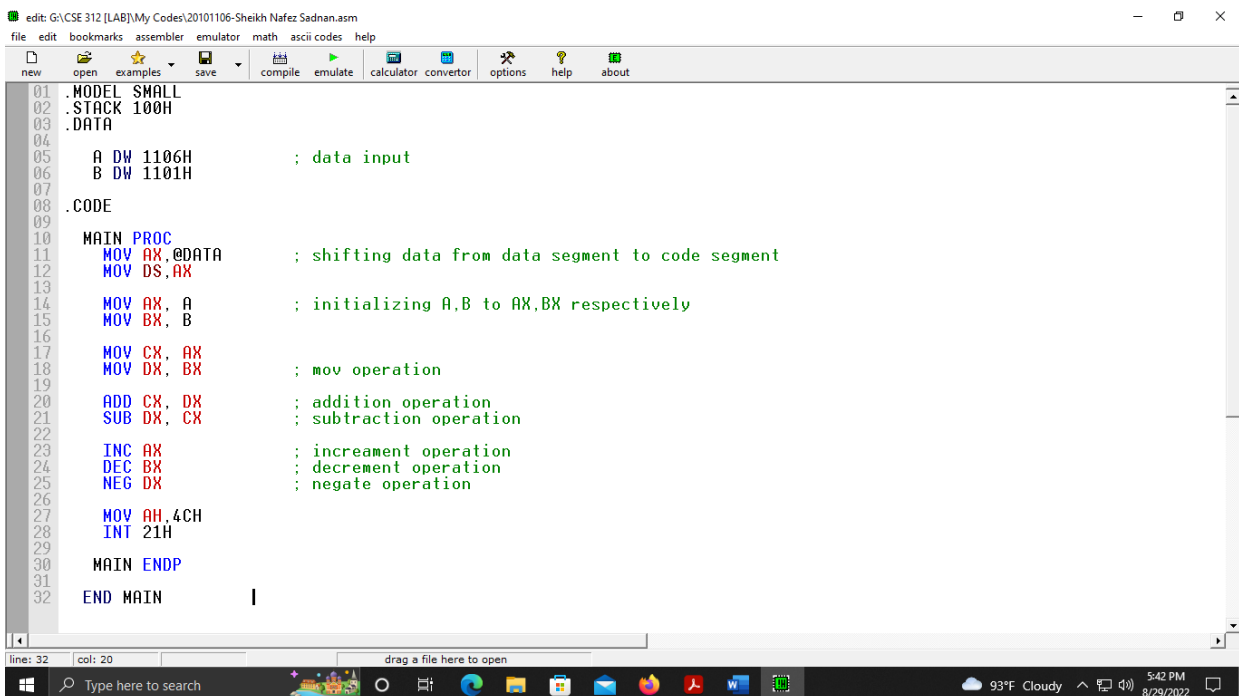
And, BX = last 4 digits of your best friend's ID

According to this,

AX= 1106 (My UAP ID- 20101106)

BX= 1101 (Md. Asadujjaman Noor's UAP ID- 20101101)

Assembly Code Screenshot:

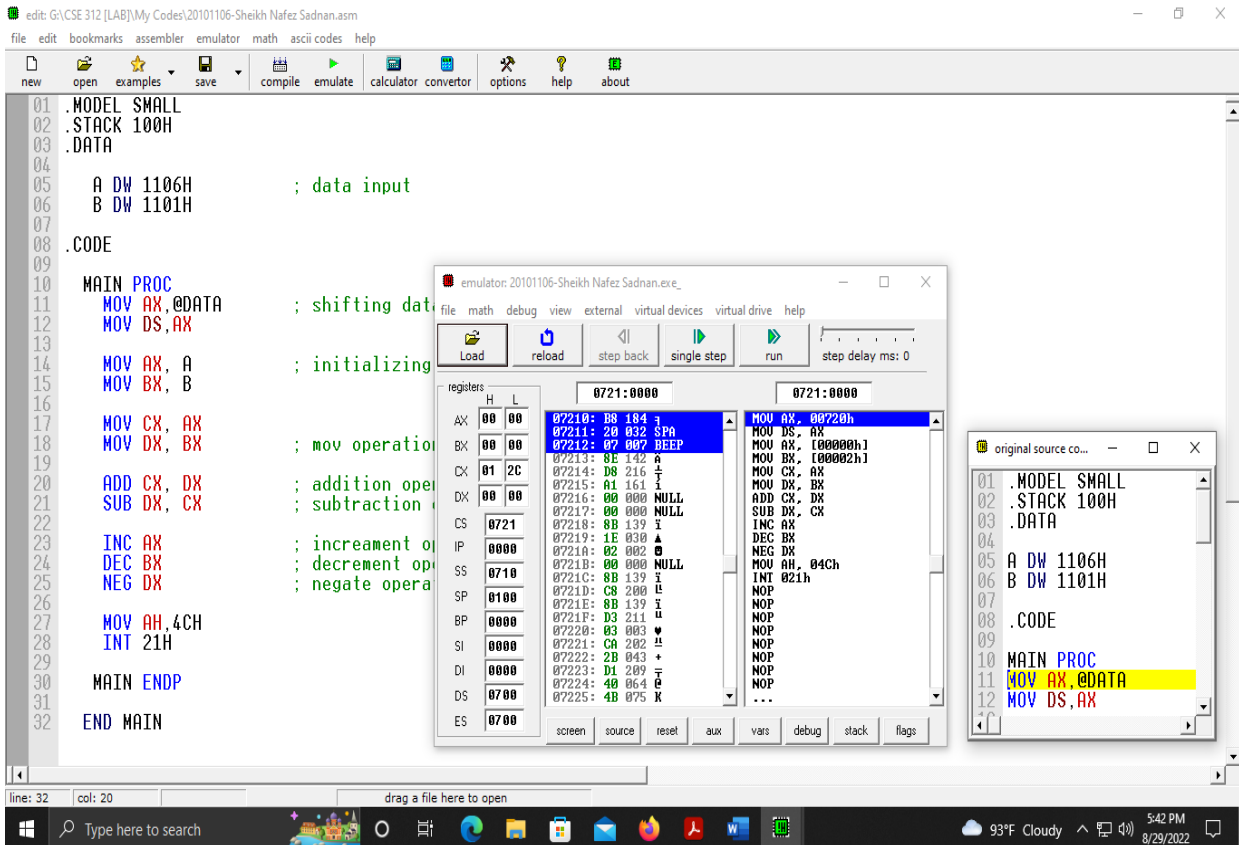


```
edit: G:\CSE 312 [LAB]\My Codes\20101106-Sheikh Nafez Sadnan.asm
file  edit  bookmarks  assembler  emulator  math  ascii codes  help

new  open  examples  save  compile  emulate  calculator  convertor  options  help  about

01 .MODEL SMALL
02 .STACK 100H
03 .DATA
04
05     A DW 1106H           ; data input
06     B DW 1101H
07
08 .CODE
09
10 MAIN PROC
11     MOV AX, @DATA        ; shifting data from data segment to code segment
12     MOV DS, AX
13
14     MOV AX, A             ; initializing A,B to AX,BX respectively
15     MOV BX, B
16
17     MOV CX, AX
18     MOV DX, BX           ; mov operation
19
20     ADD CX, DX            ; addition operation
21     SUB DX, CX           ; subtraction operation
22
23     INC AX               ; increment operation
24     DEC BX               ; decrement operation
25     NEG DX               ; negate operation
26
27     MOV AH, 4CH
28     INT 21H
29
30 MAIN ENDP
31
32 END MAIN
```

Assembly Code Screen Shot (While Emulating):



Analysis of changes to the Register:

Analysis of changes in the register can be found after emulating the code. Here, step by step operation and changes has been shown in the below figures as screen shots-

1st Operation: MOV AX, A (Where, A=1106H)

registers		H	L
AX		11	06
BX		00	00
CX		01	2C
DX		00	00
CS		0721	
IP		0008	
SS		0710	
SP		0100	
BP		0000	
SI		0000	
DI		0000	
DS		0720	
ES		0700	

After the operation, value of A (1106H) is transferred from source (A) to destination (AX).

2nd Operation: MOV BX, B (Where, B=1101H)

registers		H	L
AX		11	06
BX		11	01
CX		01	2C
DX		00	00
CS		0721	
IP		000C	
SS		0710	
SP		0100	
BP		0000	
SI		0000	
DI		0000	
DS		0720	
ES		0700	

After the operation, value of B (1101H) is transferred from source (B) to destination (BX).

3rd Operation: MOV CX, AX (Where, AX=1106H)

registers		H	L
AX		11	06
BX		11	01
CX		11	06
DX		00	00
CS		0721	
IP		000E	
SS		0710	
SP		0100	
BP		0000	
SI		0000	
DI		0000	
DS		0720	
ES		0700	

After the operation, value of AX (1106H) is transferred from source (AX) to destination(CX).

4th Operation: MOV DX, BX (Where, BX=1101H)

registers		H	L
AX		11	06
BX		11	01
CX		11	06
DX		11	01
CS		0721	
IP		0010	
SS		0710	
SP		0100	
BP		0000	
SI		0000	
DI		0000	
DS		0720	
ES		0700	

After the operation, value of BX (1101H) is transferred from source (BX) to destination(DX).

5th Operation: ADD CX, DX (Where, CX=1106H, DX=1101H)

registers

	H	L
AX	11	06
BX	11	01
CX	22	07
DX	11	01
CS	0721	
IP	0012	
SS	0710	
SP	0100	
BP	0000	
SI	0000	
DI	0000	
DS	0720	
ES	0700	

After the operation, value of DX (1101H) is added with CX (1106H)
Sum is stored in CX

Analyzing the above screen shot,
We know,
DX=1101H
CX=1106H

$$\begin{aligned} & (DX+CX) \\ &= (1101H+1106H) \\ &= 2207H \end{aligned}$$

So, CX=2207H

6th Operation: SUB DX, CX (Where, DX=1101H, CX=2207H)

registers		
	H	L
AX	11	06
BX	11	01
CX	22	07
DX	EE	FA
CS	0721	
IP	0014	
SS	0710	
SP	0100	
BP	0000	
SI	0000	
DI	0000	
DS	0720	
ES	0700	

After the operation, value of CX (2207H) is subtracted from DX (1101H)
Subtracted value is stored in DX

Analyzing the above screen shot,

We know,

DX=1101H

CX=2207H

$$\begin{aligned} & (DX-CX) \\ & = (1101H-2207H) \\ & = EEFAH \end{aligned}$$

So, DX=EEFAH

7th Operation: INC AX (Where, AX=1106H)

registers		
	H	L
AX	11	07
BX	11	01
CX	22	07
DX	EE	FA
CS	0721	
IP	0015	
SS	0710	
SP	0100	
BP	0000	
SI	0000	
DI	0000	
DS	0720	
ES	0700	

After the operation, value of AX (1106) is incremented to 1107H

Analyzing the above screen shot,

We know,

AX=1106H

Increment Value=0001H

(AX+ Increment Value)

=(1106H-0001H)

=1107H

So, AX=1107H

8th Operation: DEC BX (Where, BX=1101H)

registers		H	L
AX		11	07
BX		11	00
CX		22	07
DX		EE	FA
CS		0721	
IP		0016	
SS		0710	
SP		0100	
BP		0000	
SI		0000	
DI		0000	
DS		0720	
ES		0700	

After the operation, value of BX (1101) is decremented to 1100H

Analyzing the above screen shot,

We know,

BX=1101H

Decrement Value=0001H

(BX- Decrement Value)

=(1101H-0001H)

=1100H

So, BX=1100H

9th Operation: NEG DX (Where, DX=EEFAH)

registers		
	H	L
AX	11	07
BX	11	00
CX	22	07
DX	11	06
CS	0721	
IP	0018	
SS	0710	
SP	0100	
BP	0000	
SI	0000	
DI	0000	
DS	0720	
ES	0700	

After the operation, value of DX (EEFAH) has formed a negative value.

Analyzing the above screen shot,

Here,

15's complement = $FFFFH - EEFAH = 1105H$

16's complement = $1105H + 0001H = 1106H$

So,

DX = 1106H.

Tabular Form:

Instructions	Opcode	Decimal Equivalent	Logical Addresses	Physical Address
MOV AX, A	A1 00 00	161 000 000	0721:0005	07215
MOV BX, B	8B 1E 02 00	139 030 002 000	0721:0008	07218
MOV CX, AX	8B C8	139 200	0721:000C	0721D
MOV DX, BX	8B D3	139 211	0721:000E	0721F
ADD CX, DX	03 CA	003 202	0721:0010	07721
SUB DX, CX	2B D1	043 209	0721:0012	07223
INC AX	40	064	0721:0014	07224
DEC BX	4B	075	0721:0015	07225
NEG DX	F7 DA	247 218	0721:0016	07227

Formula of Physical Address:

*Physical Address = Segment Address*10 + Offset Address*

Flag Status

For “ADD” Operation:

Example- ADD CX, DX

flags

CF	0
ZF	0
SF	0
OF	0
PF	0
AF	0
IF	1
DF	0

analyse

Here, after performing ADD operation, the value of IF flag is 1 and all other flags are 0.

For “SUB” Operation:

Example- SUB DX, CX

flags

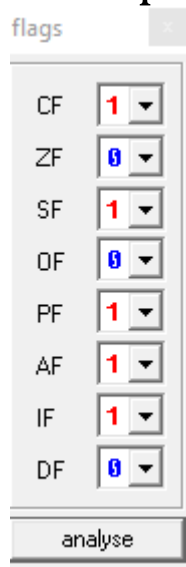
CF	0
ZF	0
SF	0
OF	0
PF	0
AF	0
IF	1
DF	0

analyse

Here, after performing SUB operation, the value of IF flag is 1 and all other flags are 0.

For “INC” Operation:

Example- INC AX



The 'flags' window displays the following flag values:

Flag	Value
CF	1
ZF	0
SF	1
OF	0
PF	1
AF	1
IF	1
DF	0

An 'analyse' button is located at the bottom of the window.

Here, after performing INC operation, the value of CF,SF,PF,AF and IF flags are 1 and all other flags are 0.

For “DEC” Operation:

Example- DEC BX



The 'flags' window displays the following flag values:

Flag	Value
CF	1
ZF	0
SF	0
OF	0
PF	0
AF	0
IF	1
DF	0

An 'analyse' button is located at the bottom of the window.

Here, after performing DEC operation, the value of CF and IF flags are 1 and all other flags are 0.

For “NEG” Operation:

Example-NEG DX

flags

CF	1
ZF	0
SF	0
OF	0
PF	1
AF	0
IF	1
DF	0

analyse

Here, after performing NEG operation, the value of CF,PF and IF flags are 1 and all other flags are 0.