

UNIVERSITY OF ASIA PACIFIC

**Department of Computer Science & Engineering**

**LAB ASSIGNMENT-02**

**Course Title :** Microprocessors and Assembly LanguageLab

**Course Code :** CSE 312

**Date of Submission:** 29/08/2022

**Submitted by: Submitted To: Name :** Sheikh Nafez Sadnan Faria Zarin Subah **Reg. No. :** 20101106 Lecturer

**Roll No. :** 106 Department of CSE

**Section :** B (2) 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

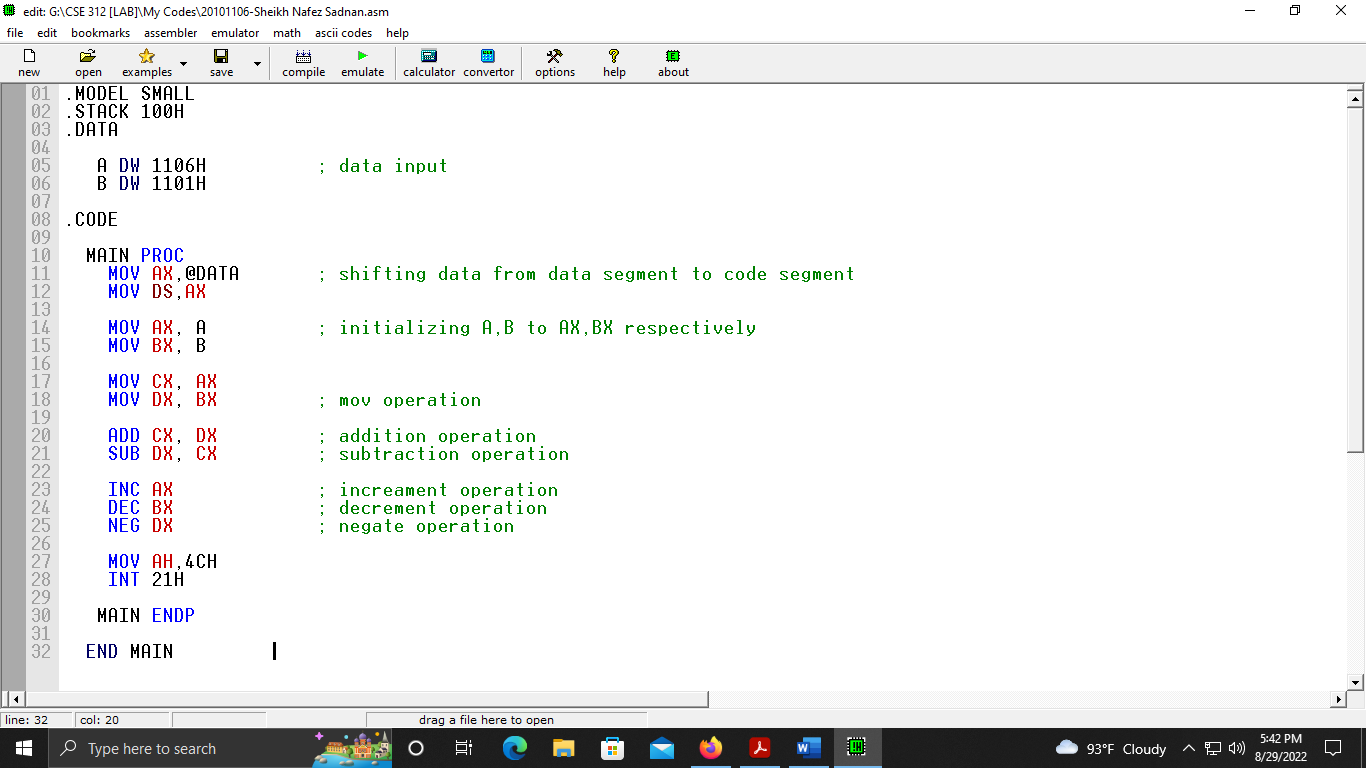
And, BX = last 4 digits of your best friend’s ID

According to this,

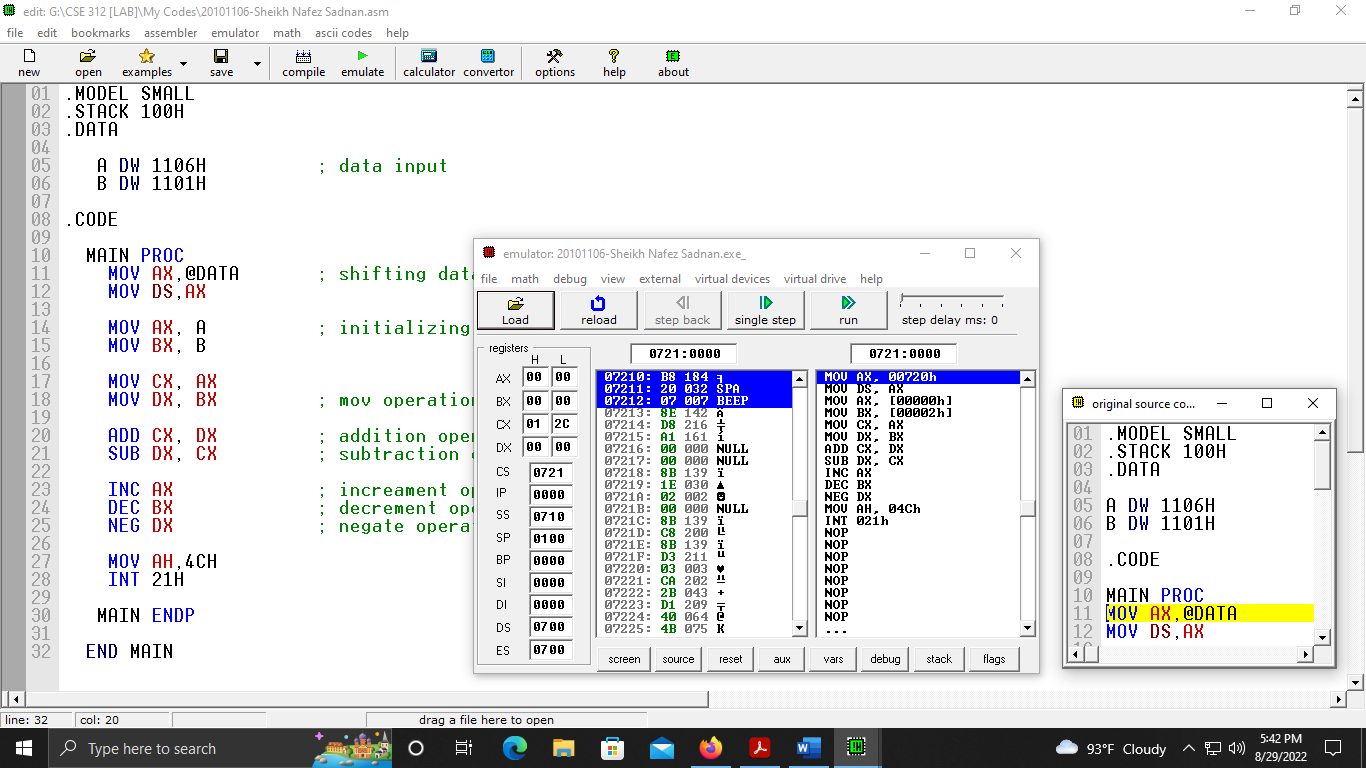
AX= 1106 (My UAP ID- 20101106)

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

**Assembly Code Screenshot:**

****

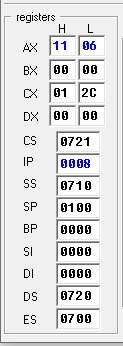
**Assembly Code Screen Shot (While Emulating):**

****

**Analysis of changes to the Register:**

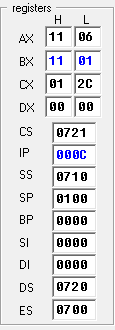
Analysis of changes in the register can be found after emulating the code. Here, step by step operation and changes has been shown in the below figures as screen shots-

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



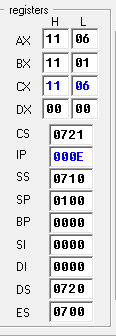
After the operation, value of A (1106H) is transferred from source (A) to destination(AX).

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



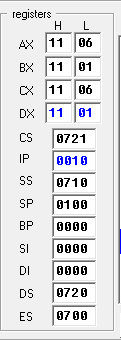
After the operation, value of B (1101H) is transferred from source (B) to destination(BX).

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



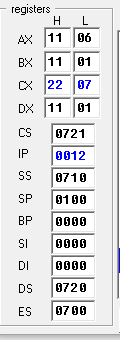
After the operation, value of AX (1106H) is transferred from source (AX) to destination(CX).

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



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)



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

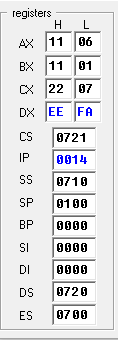
(DX+CX)

=(1101H+1106H)

=2207H

So, CX=2207H

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



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

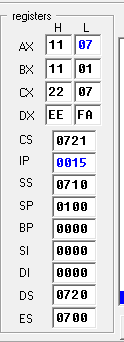
(DX-CX)

=(1101H-2207H)

=EEFAH

So, DX=EEFAH

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



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

Analyzing the above screen shot,

We know,

AX=1106H

Increment Value=0001H

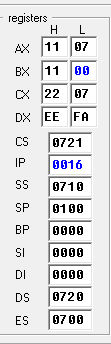
(AX+ Increment Value)

=(1106H-0001H)

=1107H

So, AX=1107H

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



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

Analyzing the above screen shot,

We know,

BX=1101H

Decrement Value=0001H

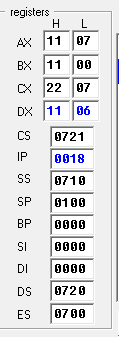
(BX- Decrement Value)

=(1101H-0001H)

=1100H

So, BX=1100H

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



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

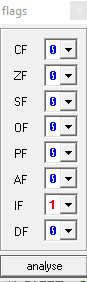
**Formula of Physical Address:**

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

**Flag Status**

**For “ADD” Operation:**

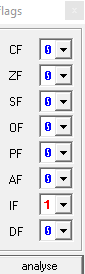
Example- ADD CX, DX



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

**For “SUB” Operation:**

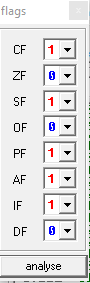
Example- SUB DX, CX



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

**For “INC” Operation:**

Example- INC AX



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

****

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



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