VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

Microprocessors and Interfacing

(CSE – 3002)

LAB TASK – 1

Name: **Vibhu Kumar Singh**

Reg. No: **19BCE0215**

Teacher: **Mr. Konguvel E.**

Reg. No – 19BCE0215

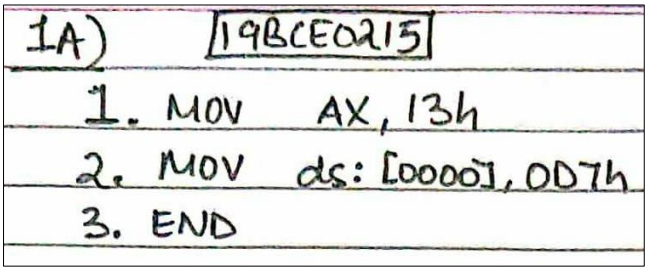
DATA 1 : 19d = 13h **DATA 2 : 0215d = D7h**

1A: Perform two data transfer function:

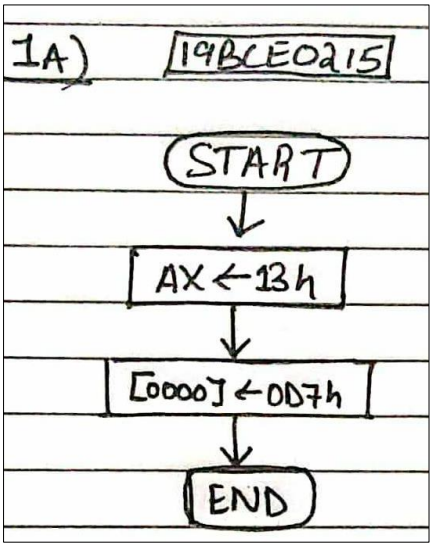
DATA 1 to General Purpose Registers
DATA 2 to Any memory location

Ans 1A:

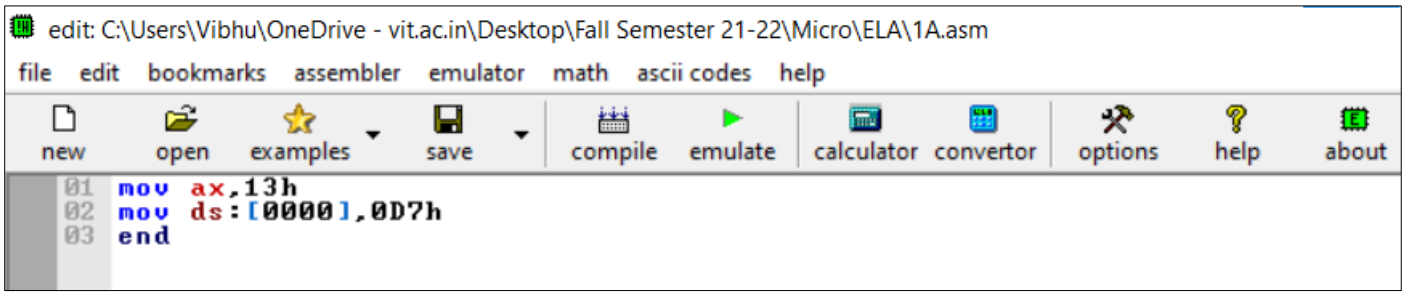
1. Handwritten Assembly Language Program (ALP):



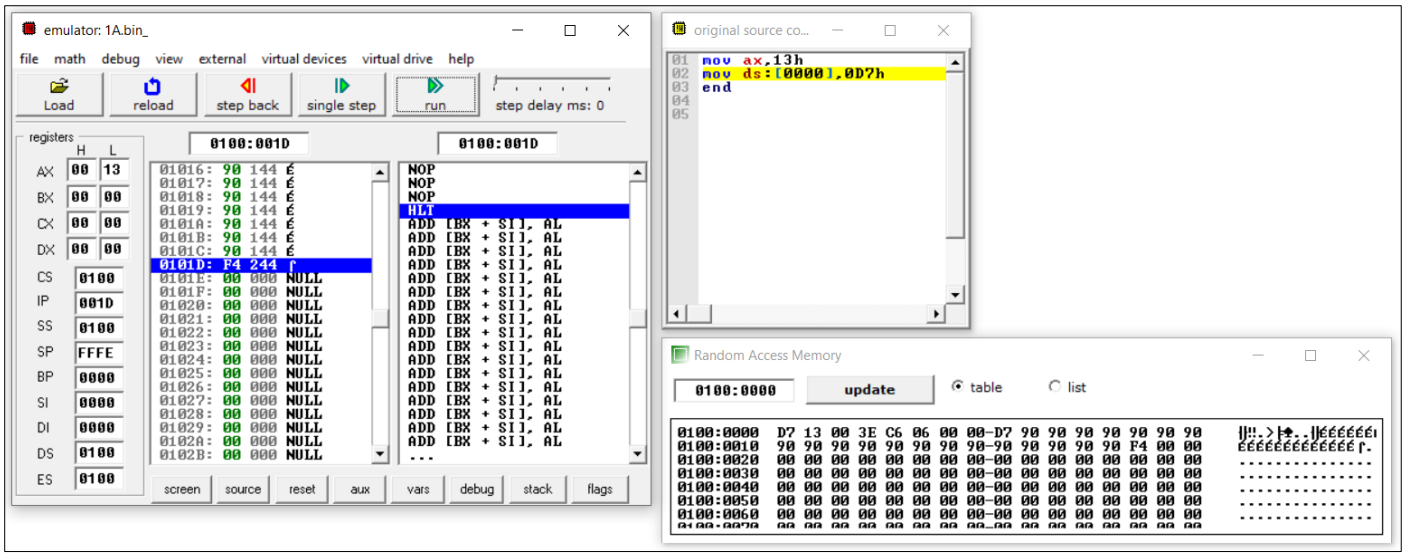
2. Flow Chart:



3. Handwritten Calculations for Verification: **(N/A)**
4. Snapshot of ALP:



5. Snapshot of Output (Registers/Memory/Stack) and status of Flag registers.



Reg. No – 19BCE0215

DATA 1 : 19d = 13h **DATA 2 : 0215d = D7h**

1B: Perform any two Arithmetic operations using DATA 1 and DATA 2: (ADD/SUB/MUL/DIV)

Store the result in General Purpose Registers

Check the status of Flag register

Ans 1B:

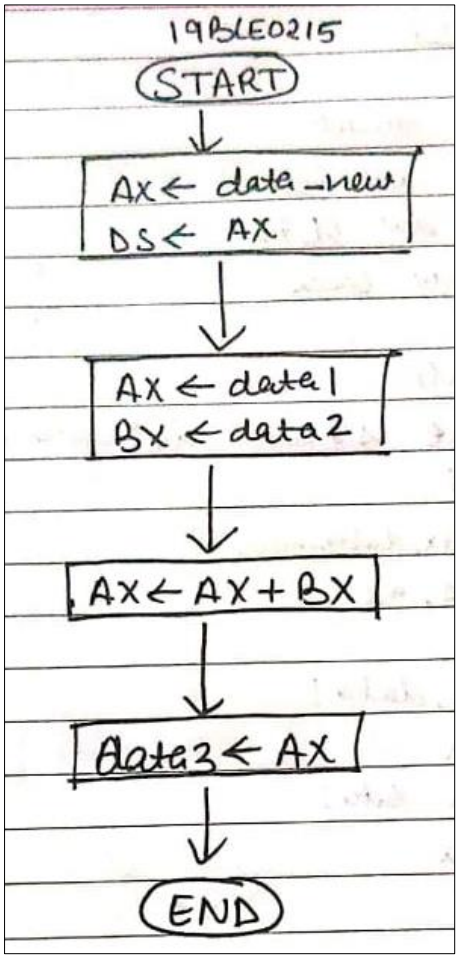
[Addition]

1. Handwritten Assembly Language Program (ALP):

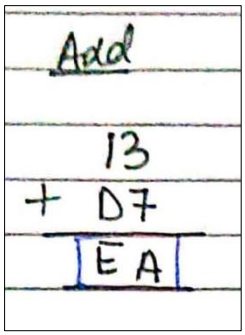
19BLE0215

1.
```
data_new segment
        data1 dw 13h
        data2 dw 0D7h
        data3 dw 1 dup(?)
        data_new ends

code segment
        assume cs: code, ds: data_new
        start:
        mov ax, data_new
        mov ds, ax

        mov ax, data1
        mov bx, data2
        add ax, bx
        mov data3, ax

        code ends
end start
end.
```

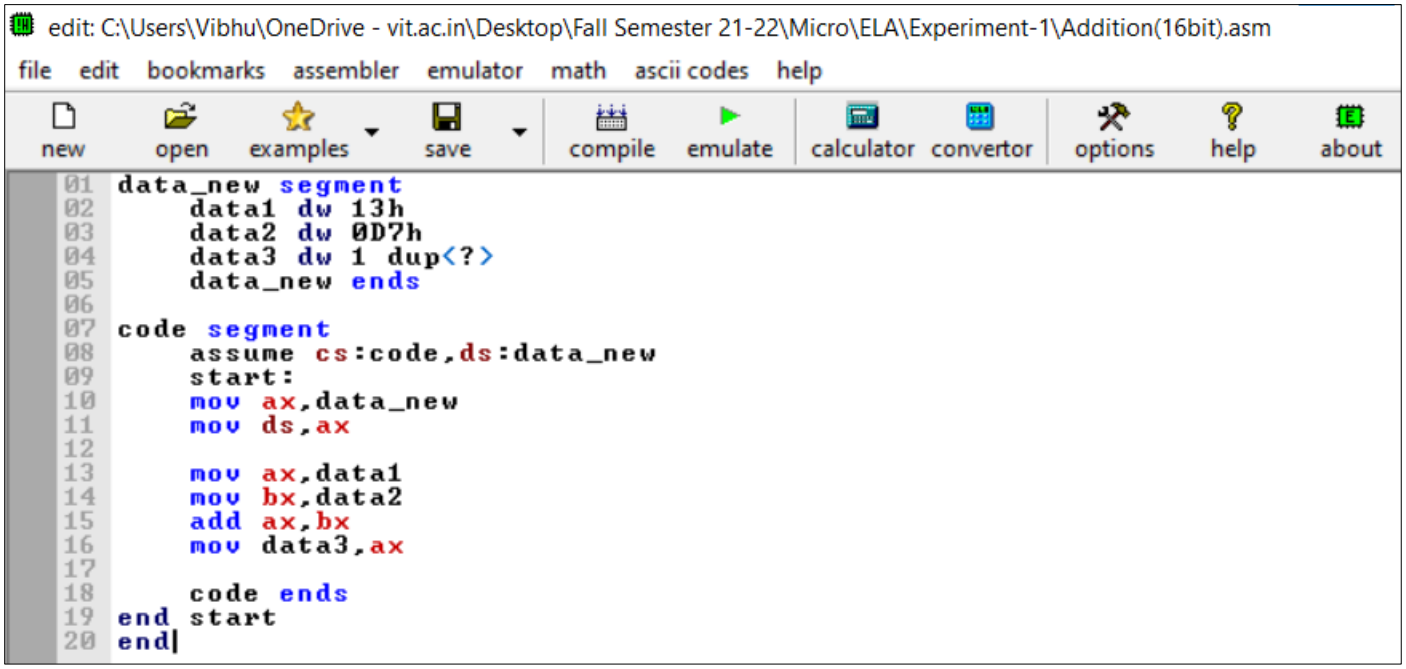
2. Flow Chart:



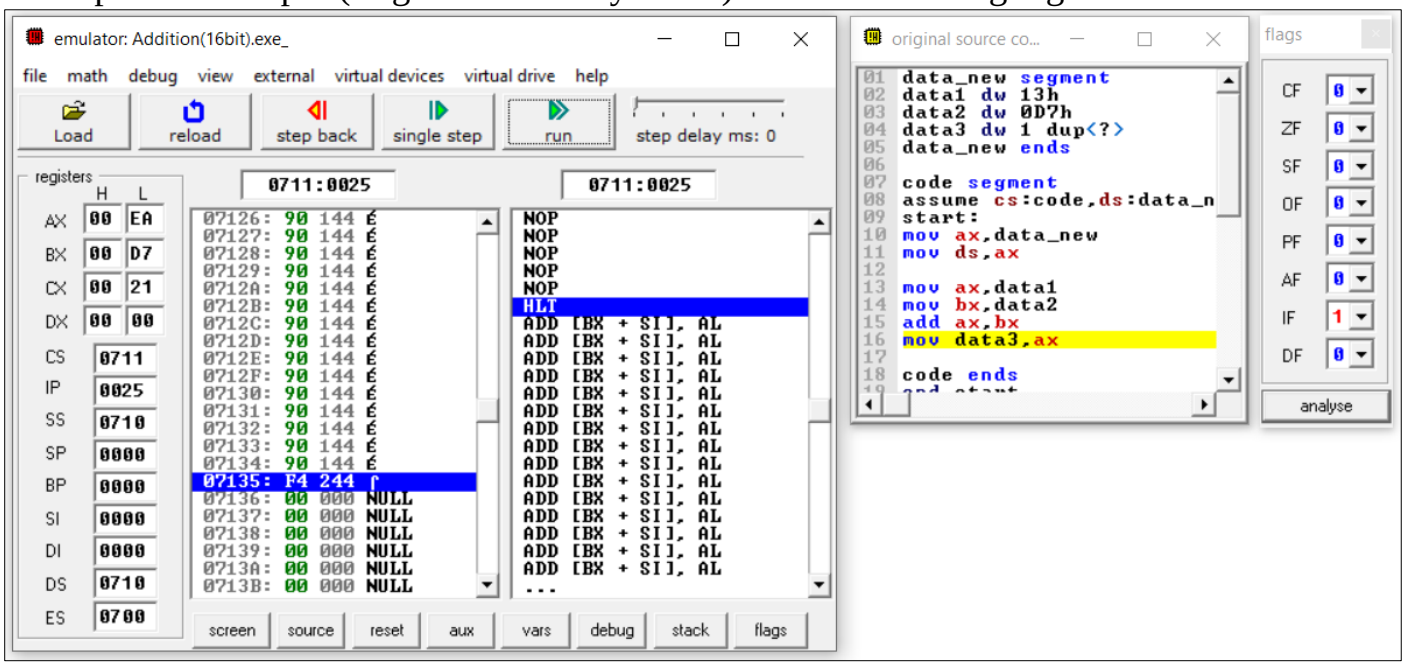
3. Handwritten Calculations for Verification:



4. Snapshot of ALP:



5. Snapshot of Output (Registers/Memory/Stack) and status of Flag registers.

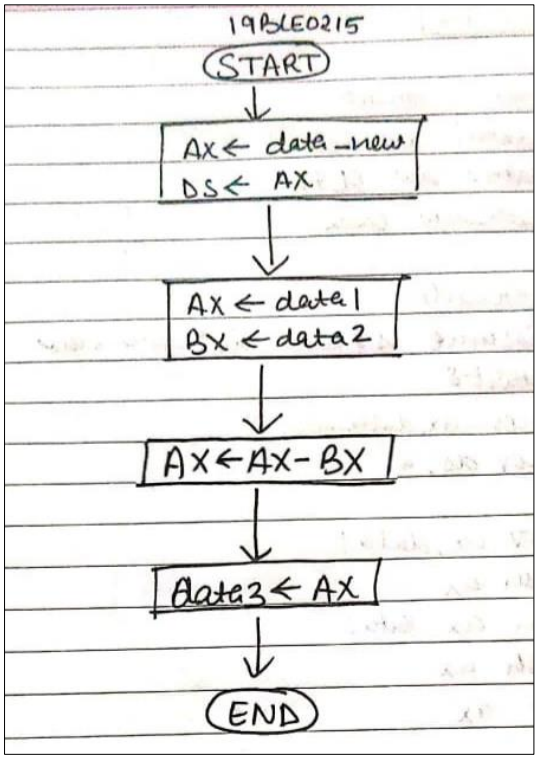


[SUBTRACT]

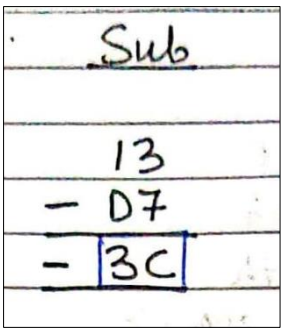
1. Handwritten Assembly Language Program (ALP):

```
19BLE0215

1. data_new segment
        data1 dw 13h
        data2 dw 007h
        data3 dw 1 dup(?)
        data_new ends

code segment
        assume cs: code, ds: data_new
        start:
        mov ax, data_new
        mov ds, ax

        mov ax, data1
        mov bx, data2
        sub ax, bx
        mov data3, ax

        code ends
end start
end.
```

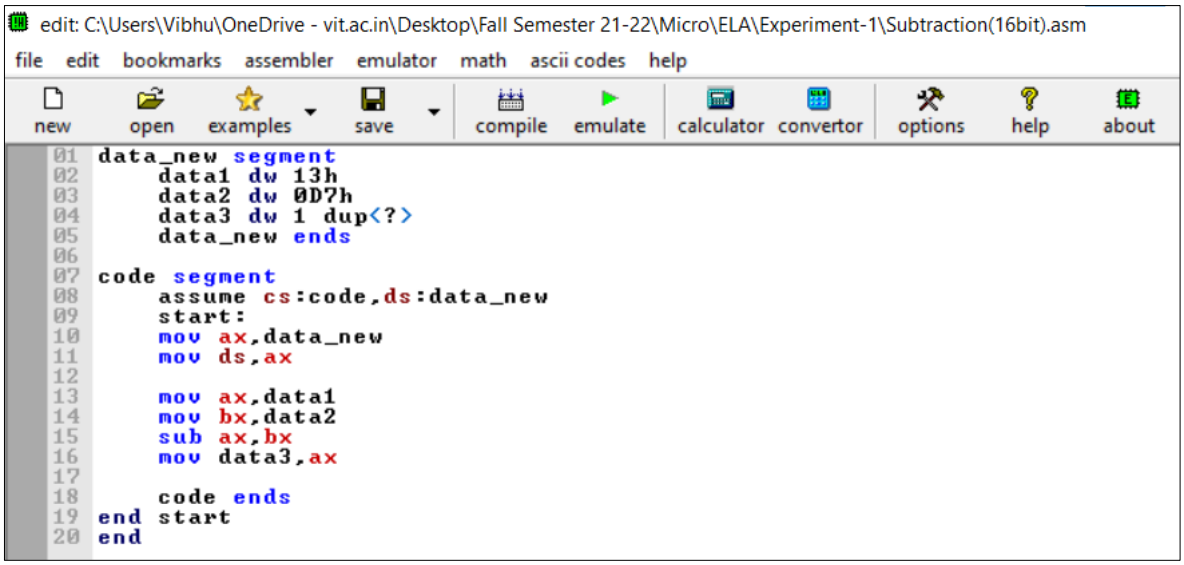
2. Flow Chart:



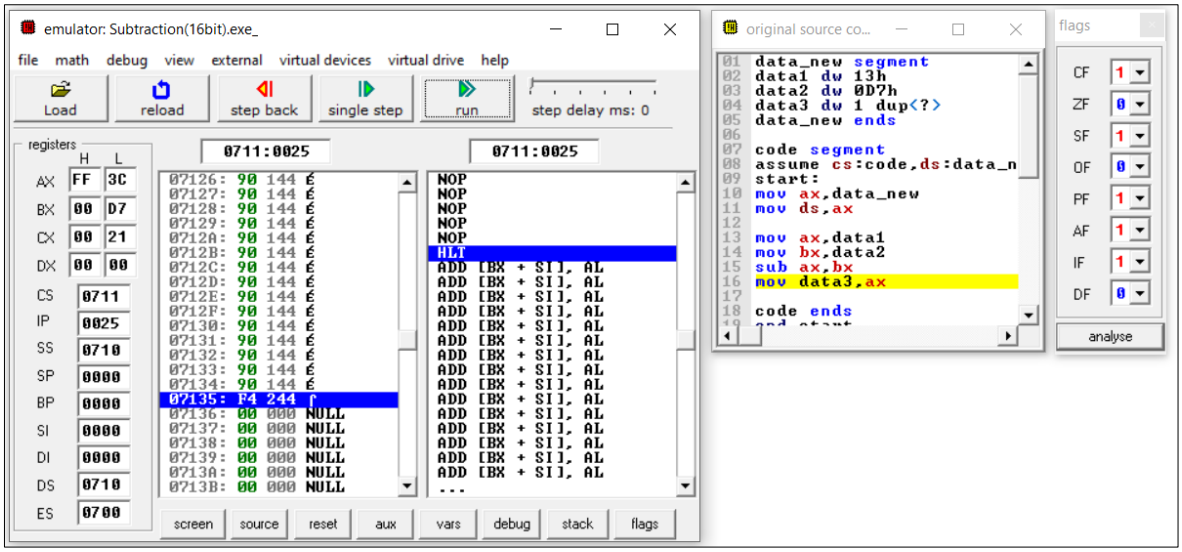
3. Handwritten Calculations for Verification:



4. Snapshot of ALP:



5. Snapshot of Output (Registers/Memory/Stack) and status of Flag registers.



Reg. No – 19BCE0215

DATA 1 : 19d = 13h **DATA 2 : 0215d = D7h**

1C: Perform any two Logical operations using DATA 1 and DATA 2: (AND/OR/XOR)

Store the result in General Purpose Registers

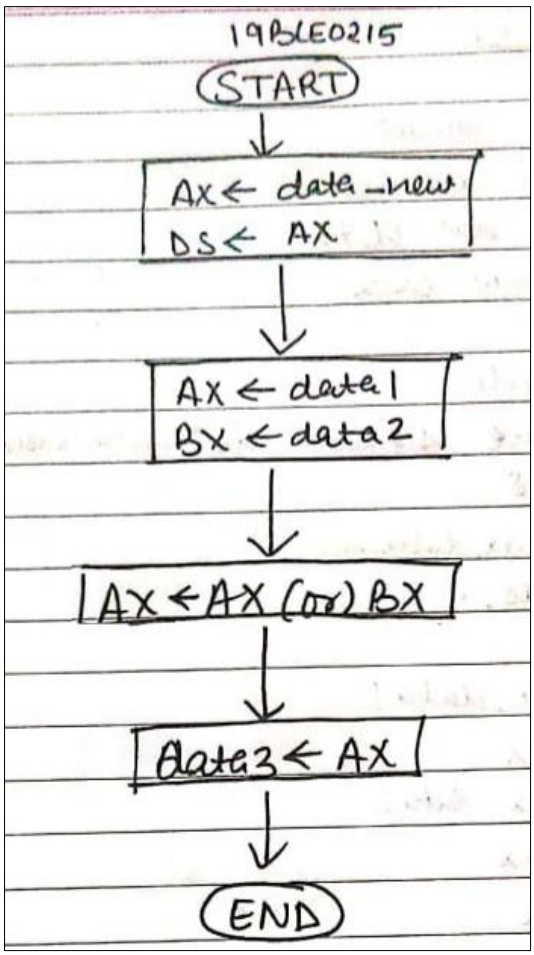
Check the status of Flag register

Ans 1C:

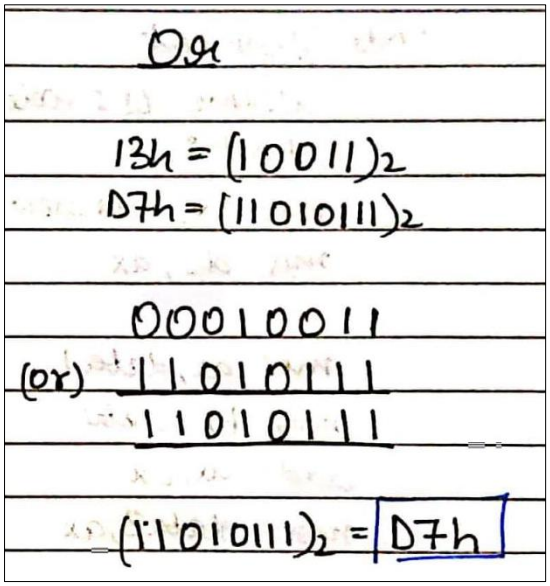
1. Handwritten Assembly Language Program (ALP):

```
19BLE0215

1. data_new segment
        data1 dw 13h
        data2 dw 0D7h
        data3 dw 1 dup(?)
        data_new ends

code segment
        assume cs: code, ds: data_new
        start:
        mov ax, data_new
        mov ds, ax

        mov ax, data1
        mov bx, data2
        or  ax, bx
        mov data3, ax

        code ends
end start
end.
```

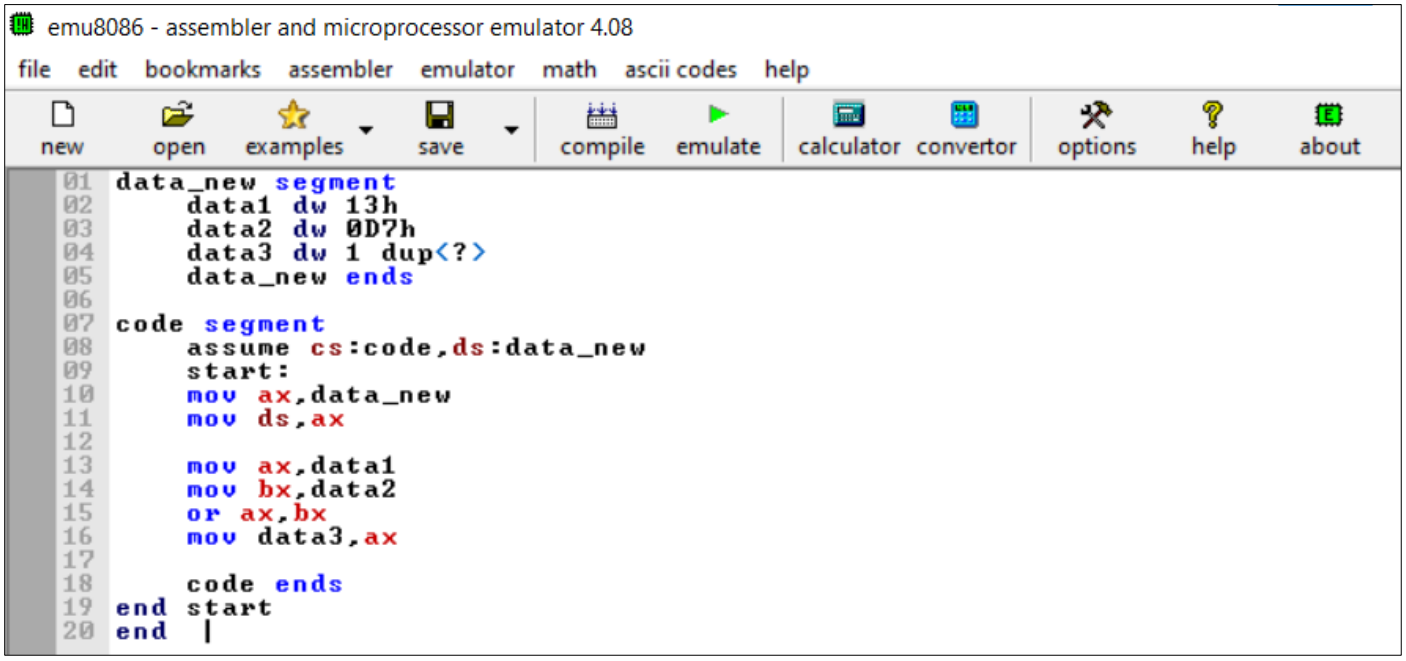
2. Flow Chart:



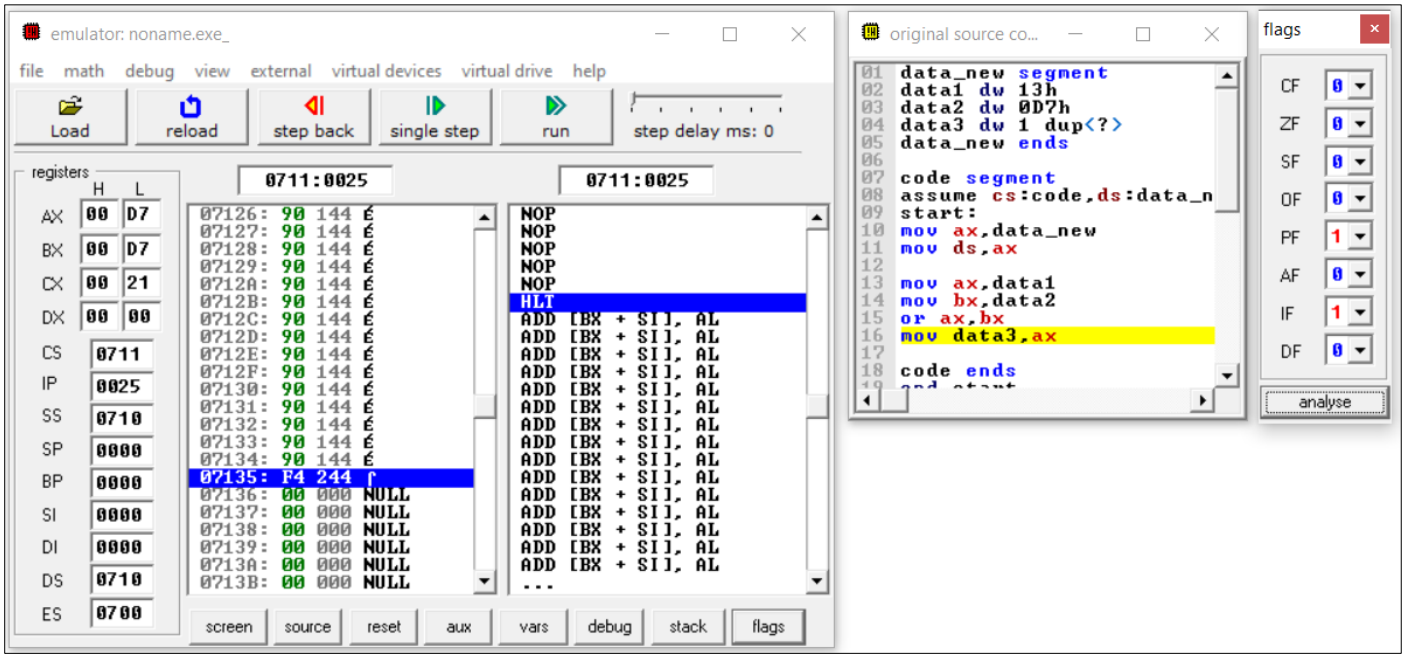
3. Handwritten Calculations for Verification:



4. Snapshot of ALP:



5. Snapshot of Output (Registers/Memory/Stack) and status of Flag registers.

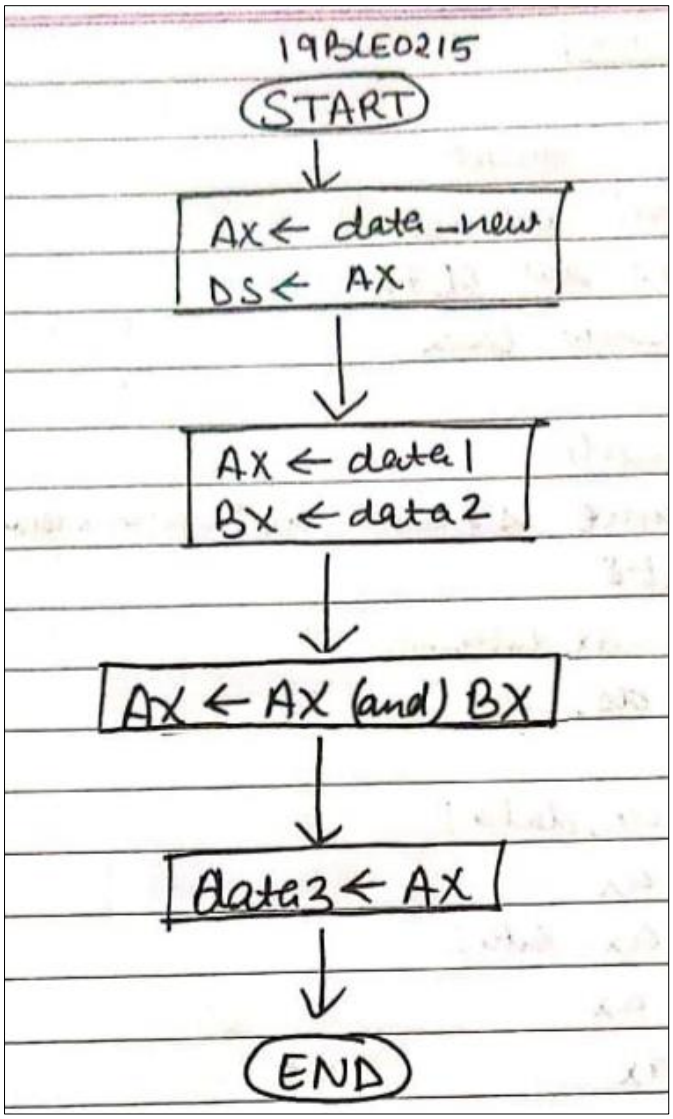


[AND]

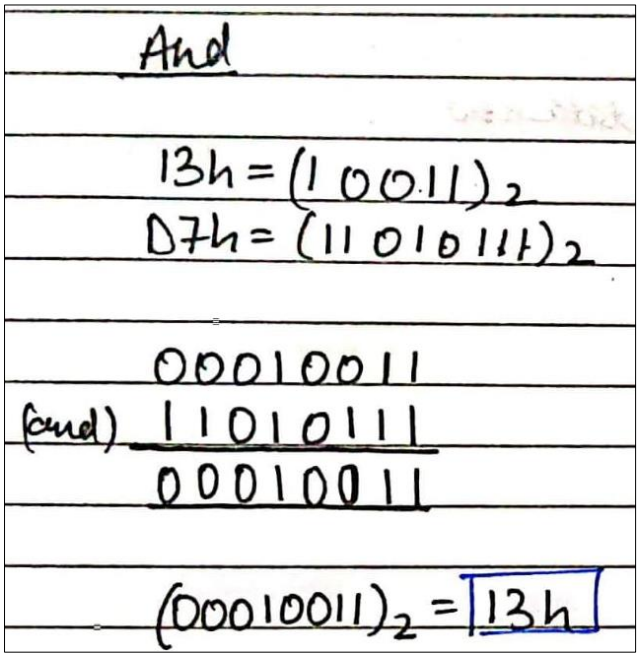
1. Handwritten Assembly Language Program (ALP):

```
19BLE0215

1. data_new segment
        data1 dw 13h
        data2 dw 007h
        data3 dw 1 dup(?)
        data_new ends

code segment
        assume cs: code, ds: data_new
        start:
        mov ax, data_new
        mov ds, ax

        mov ax, data1
        mov bx, data2
        and ax, bx
        mov data3, ax

        code ends
end start
end.
```

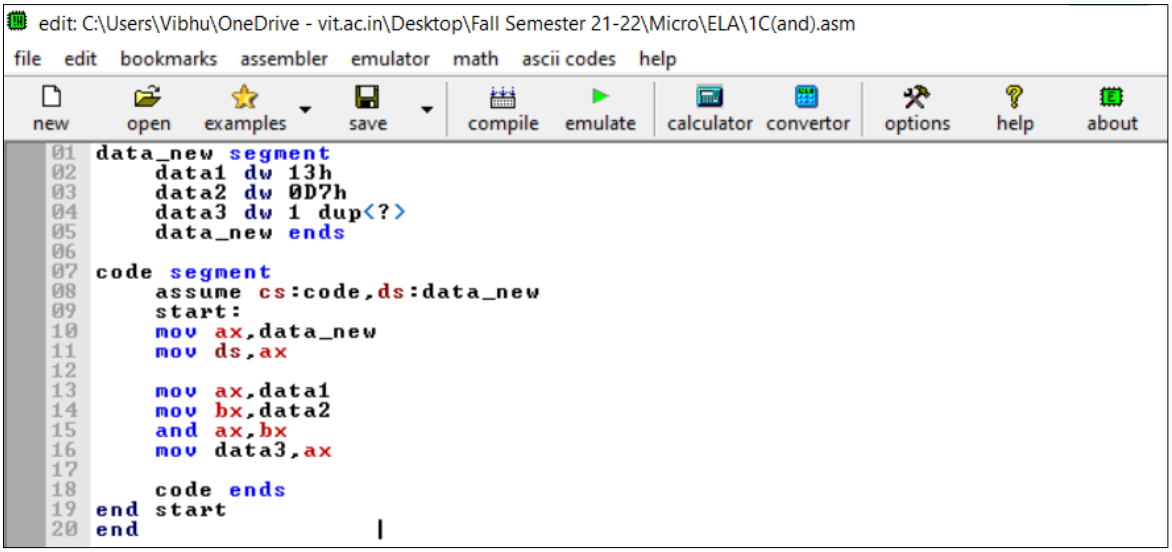
2. Flow Chart:



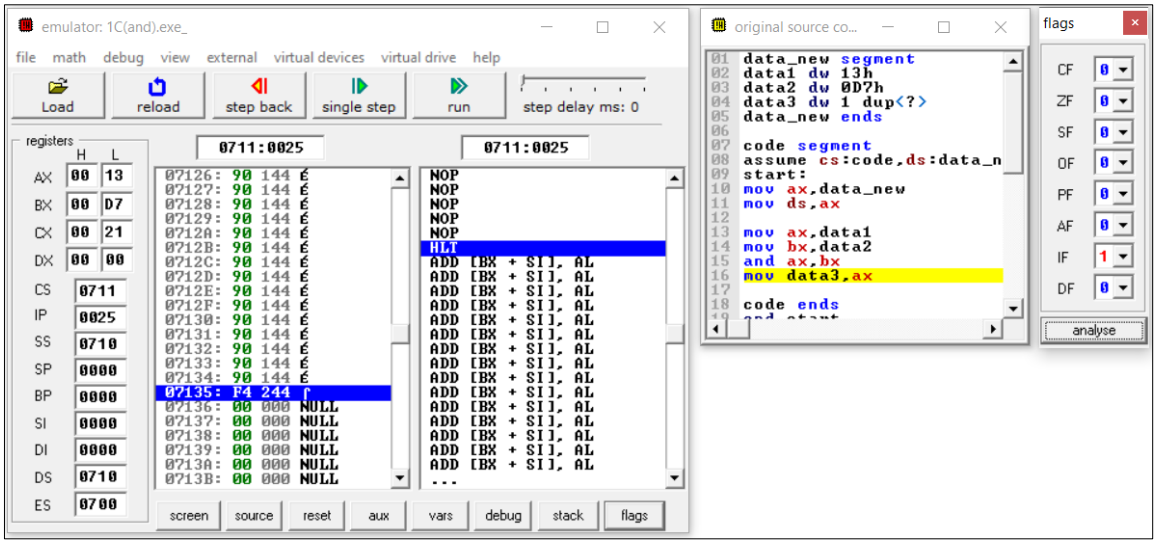
3. Handwritten Calculations for Verification:



4. Snapshot of ALP:



5. Snapshot of Output (Registers/Memory/Stack) and status of Flag registers.



Reg. No – 19BCE0215

DATA 1 : 19d = 13h **DATA 2 : 0215d = D7h**

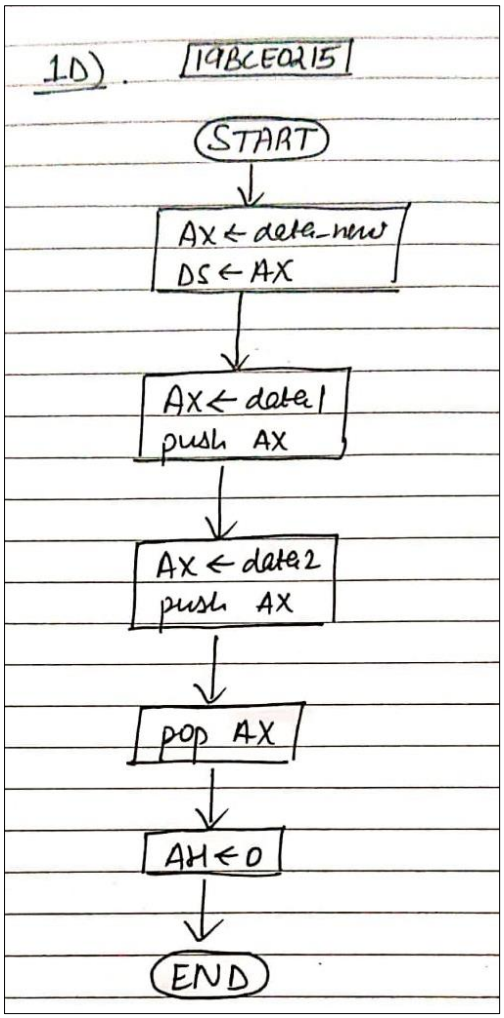
1D: Push the DATA 1, DATA 2 to Stack and Pop it from Stack.

Ans 1D:

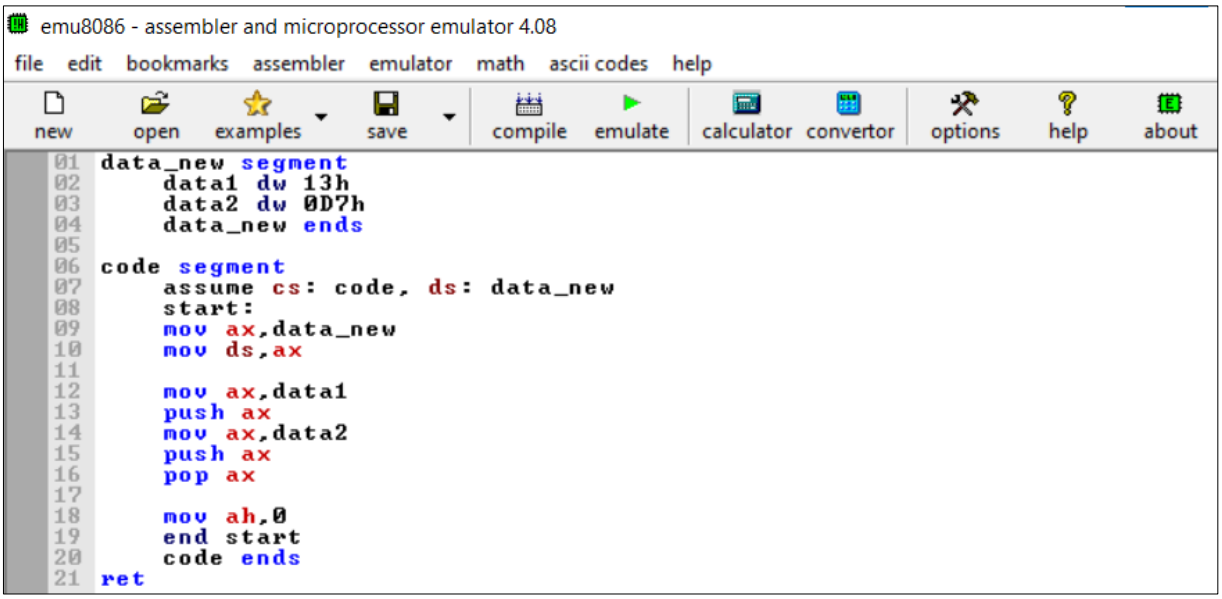
1. Handwritten Assembly Language Program (ALP):

```
1D)  [19BCE0215]

data_new segment
        data1 dw 13h
        data2 dw 0D7h
        data_new ends

code segments
        assume cs: code, ds: data_new
        start:
        mov ax, data_new
        mov ds, ax

        mov ax, data1
        push ax
        mov ax, data2
        push ax
        pop ax

        mov ah, 0
        end start
        code ends
end.
```

2. Flow Chart:

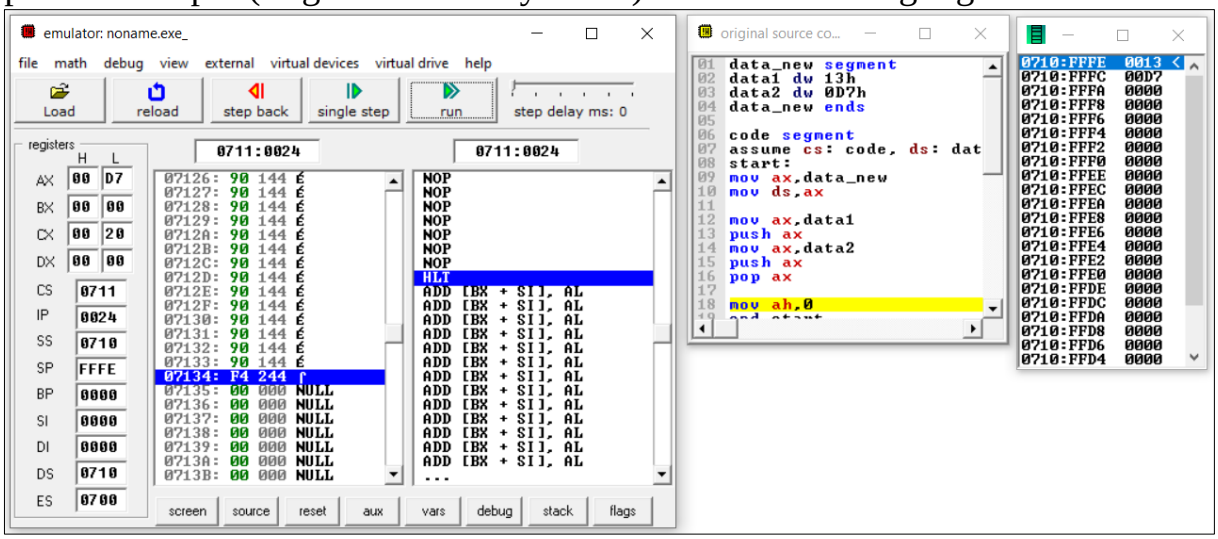


3. Handwritten Calculations for Verification: **(N/A)**
4. Snapshot of ALP:




```
data_new segment
    data1 dw 13h
    data2 dw 0D7h
    data_new ends

code segment
    assume cs: code, ds: data_new
    start:
    mov ax,data_new
    mov ds,ax

    mov ax,data1
    push ax
    mov ax,data2
    push ax
    pop ax

    mov ah,0
    end start
    code ends
ret
```

5. Snapshot of Output (Registers/Memory/Stack) and status of Flag registers.



Reg. No – 19BCE0215

DATA 1 : 19d = 13h **DATA 2 : 0215d = D7h**

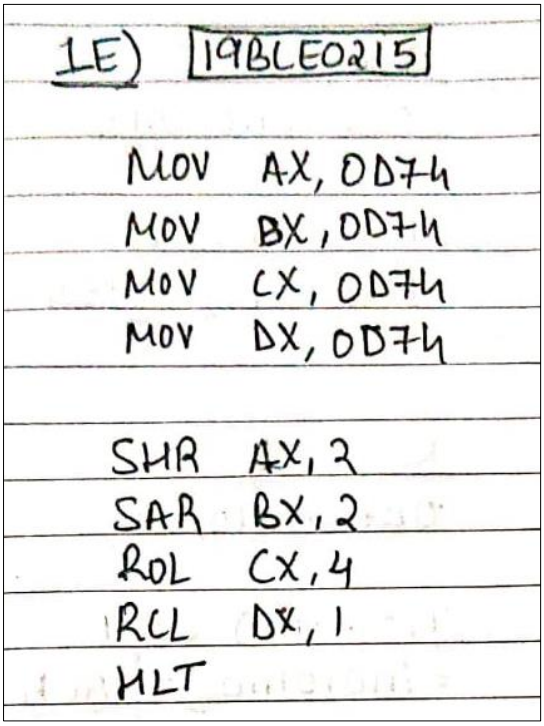
1E: Perform SHR, SAR, ROL, RCL Bit manipulation operations using DATA 2.

Store the result in General Purpose Registers

Check the status of Flag register

Ans 1E:

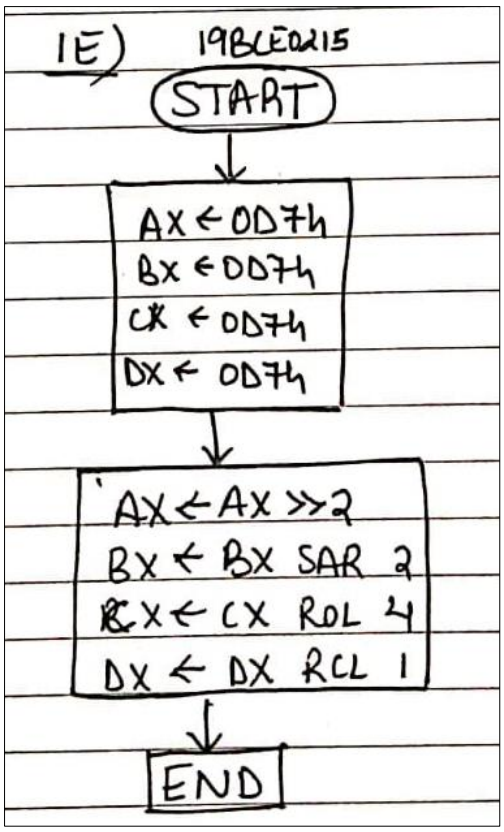
1. Handwritten Assembly Language Program (ALP):



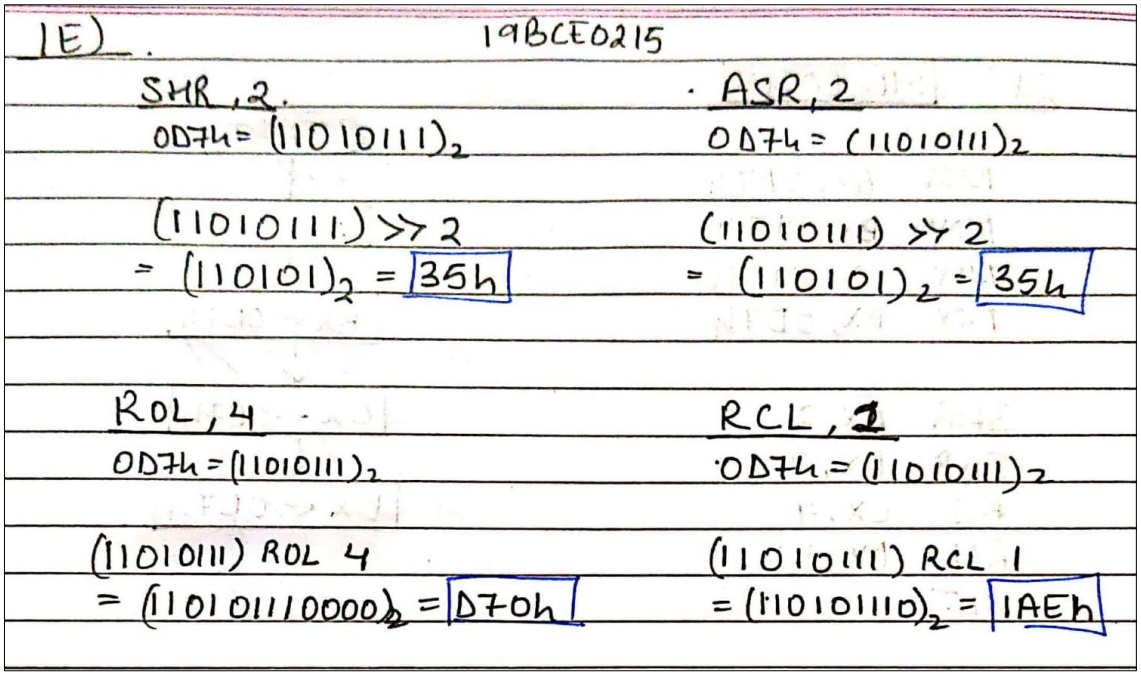
1E) 19BCE0215

```
MOV AX, 0D7h
MOV BX, 0D7h
MOV CX, 0D7h
MOV DX, 0D7h

SHR AX, 2
SAR BX, 2
ROL CX, 4
RCL DX, 1
HLT
```

2. Flow Chart:



3. Handwritten Calculations for Verification:



1E) 19BCE0215

SHR, 2

0D7h = $(11010111)_2$

$(11010111) >> 2$

$= (110101)_2 = \boxed{35h}$

ASR, 2

0D7h = $(11010111)_2$

$(11010111) >> 2$

$= (110101)_2 = \boxed{35h}$

ROL, 4

0D7h = $(11010111)_2$

(11010111) ROL 4

$= (110101110000)_2 = \boxed{D70h}$

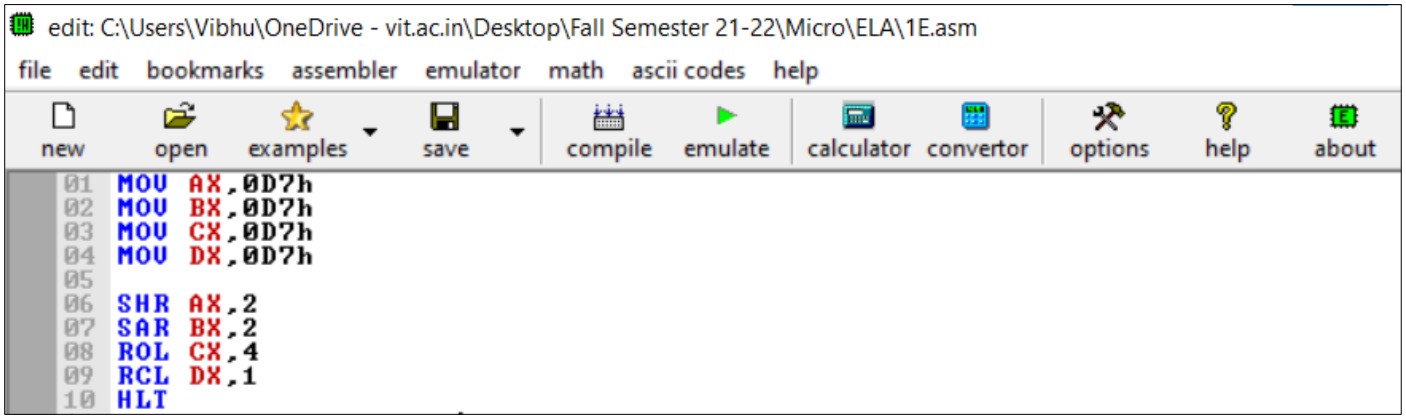
RCL, 1

0D7h = $(11010111)_2$

(11010111) RCL 1

$= (110101110)_2 = \boxed{1AEh}$

4. Snapshot of ALP:



5. Snapshot of Output (Registers/Memory/Stack) and status of Flag registers.