Name: Jigar Siddhpura **SAPID:** 60004200155

DIV: C/C2 **Branch:** Computer Engineering

POA EXPERIMENT 5

Jigar Siddhpura
60004210155

Experiment 5: Addition & Subtraction in 8086

Aim: To implement assembly program for 16-bit addition / subtraction using direct, immediate & Register addressing mode.

Theory: 8086 Microprocessor works on instructions in assembly language (a low level language). Here, with the help of 8086, we perform arithmetic operations using various addressing mode.

Addition: First, we used direct addressing mode to add 2 numbers at location [1000h] & [1002h], the result is stored in [1004h]. The carry flag is examined & if set 1 it's stored in [1006h]. Here instruction – MOV, ADD, INC, JNC, HLT are used.

Secondly we used immediate addressing mode where we add 2 immediate values – 0ffffh & 00099h. The result is stored in DX using register addressing mode. Carry flag is checked & carry count is stored in [1006h] using direct addressing.

Subtraction: Thirdly, we perform subtraction using direct addressing, so 2 no.s from memory location [1000h] & [1002h] are subtracted & result is stored in register DX using register addressing mode. Borrow flag is examined to determine if borrowing occured during subtraction. If so, borrow count

is stored in [1006h].

Lastly, immediate addressing is used for subtraction i.e. immediate values are subtracted & result is stored in register DX. Again, if borrow exists, it's stored in [1006h].

For subtraction, instructions are same as addition, except we have SUB in replacement of ADD.

Conclusion: So we used various addressing modes in 8086 to perform arithmetic operations (addition / subtraction), where direct addressing mode simplifies access, while immediate proves valuable for handling constants.

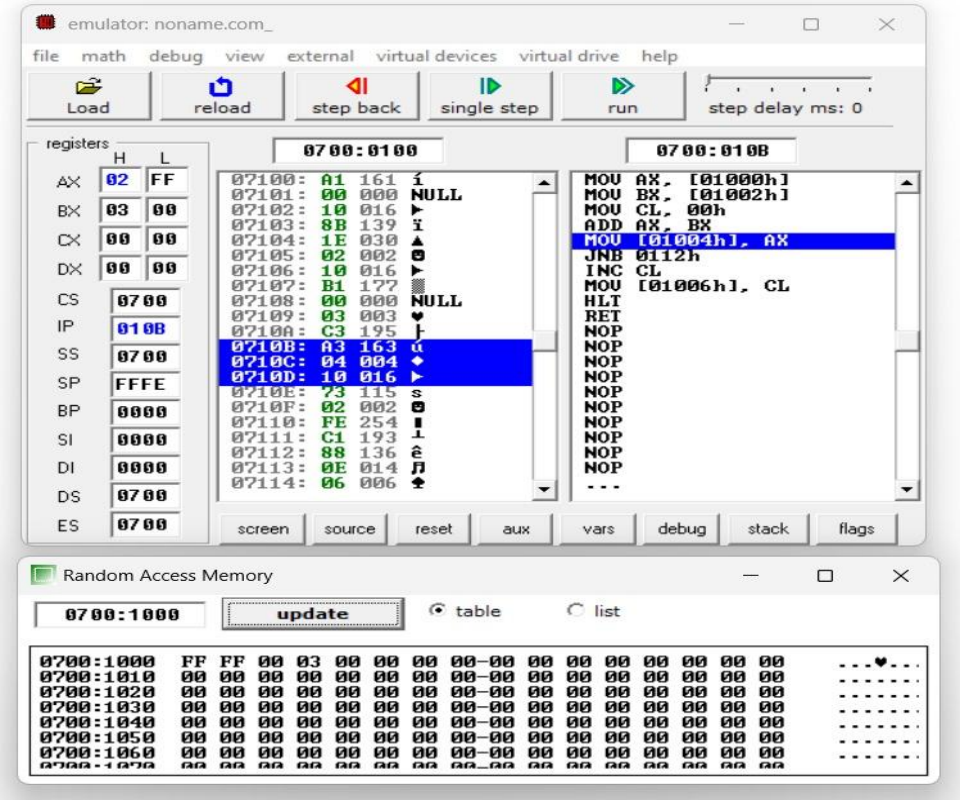
Code :

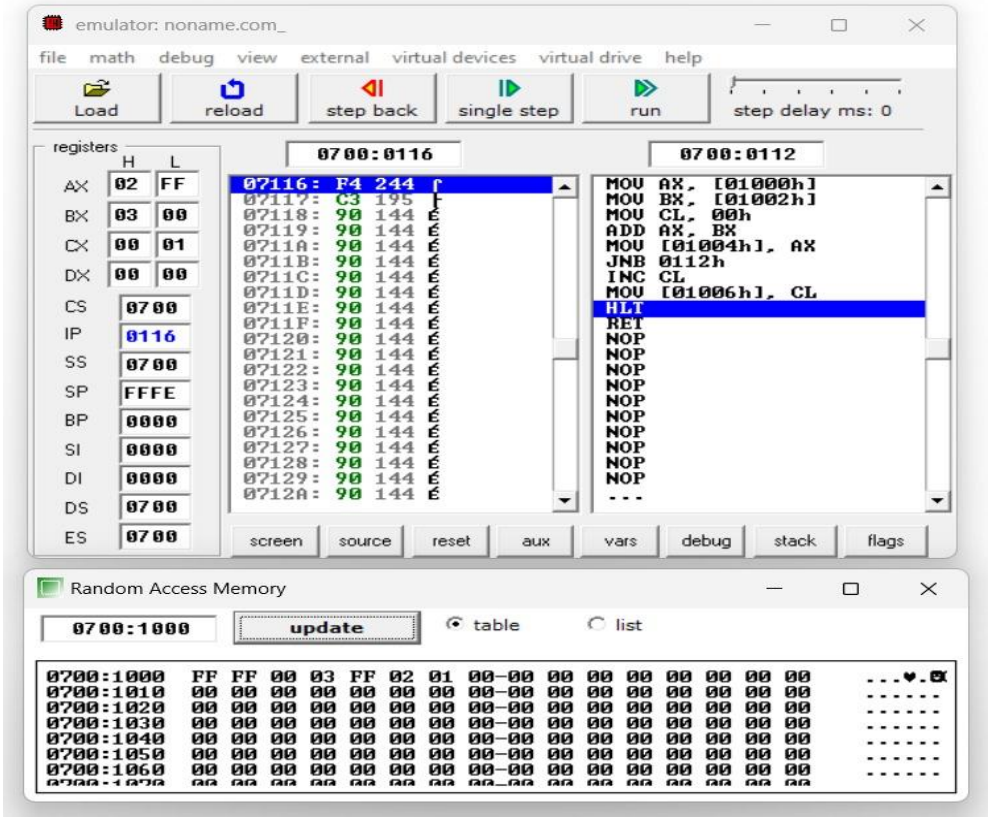
1. **Addition using Direct Addressing mode** :

```
org 100h

MOV AX, [1000h]
MOV BX, [1002h]
MOV CL,00h
ADD AX,BX
MOV [1004h],AX
JNC carry
INC CL
carry:
MOV [1006h],CL
HLT
ret
```

- Addition of ffffh + 0900h which gives output has 08ff with carry has 01 where sum can be seen at addresses 1005h,1004h and carry at 1006h



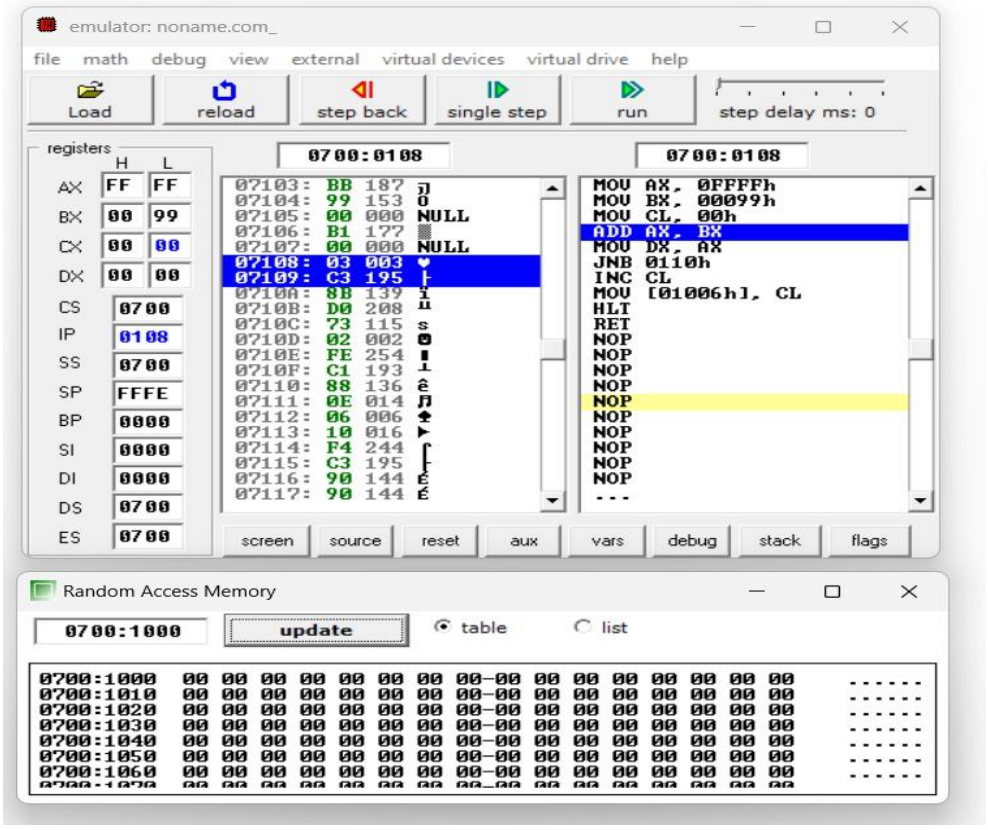


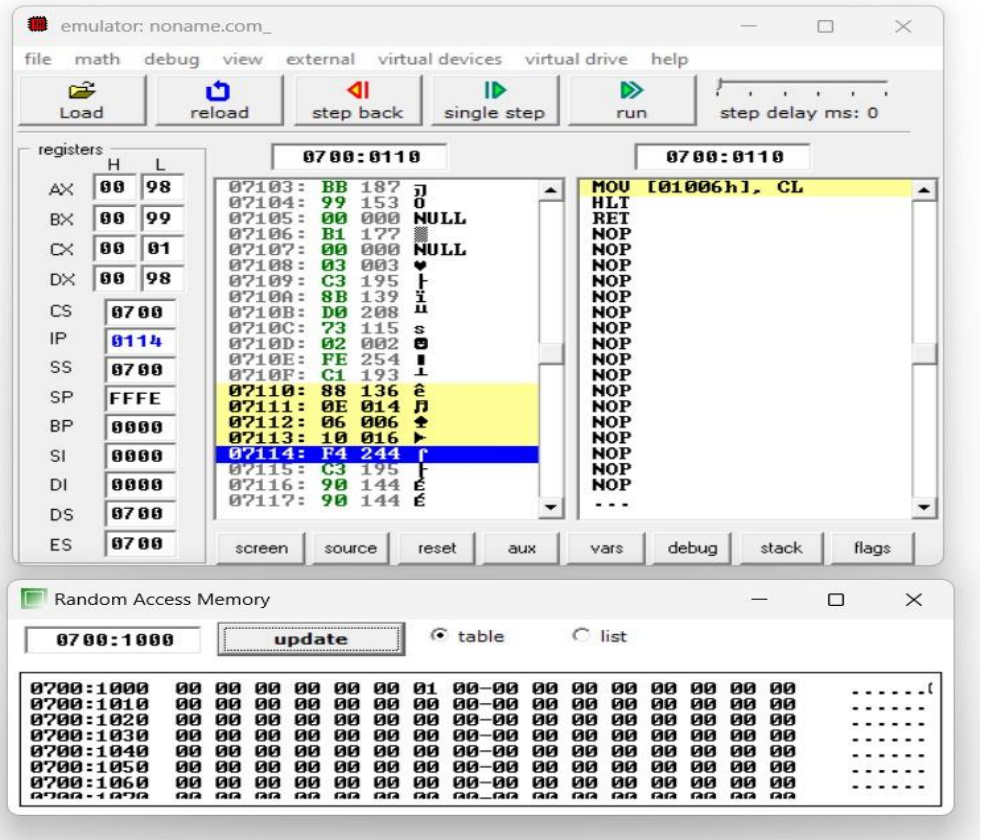
2. **Addition using Immediate Addressing mode** :

```
org 100h

MOV AX, 0ffffh
MOV BX, 00099h
MOV CL,00h
ADD AX,BX
MOV DX,AX
JNC carry
INC CL
carry:
MOV [1006h],CL
HLT
ret
```

- Addition of ffffh + 0099h which gives output has 0098 with carry has 01 where sum can be seen at register DX and carry at 1006h



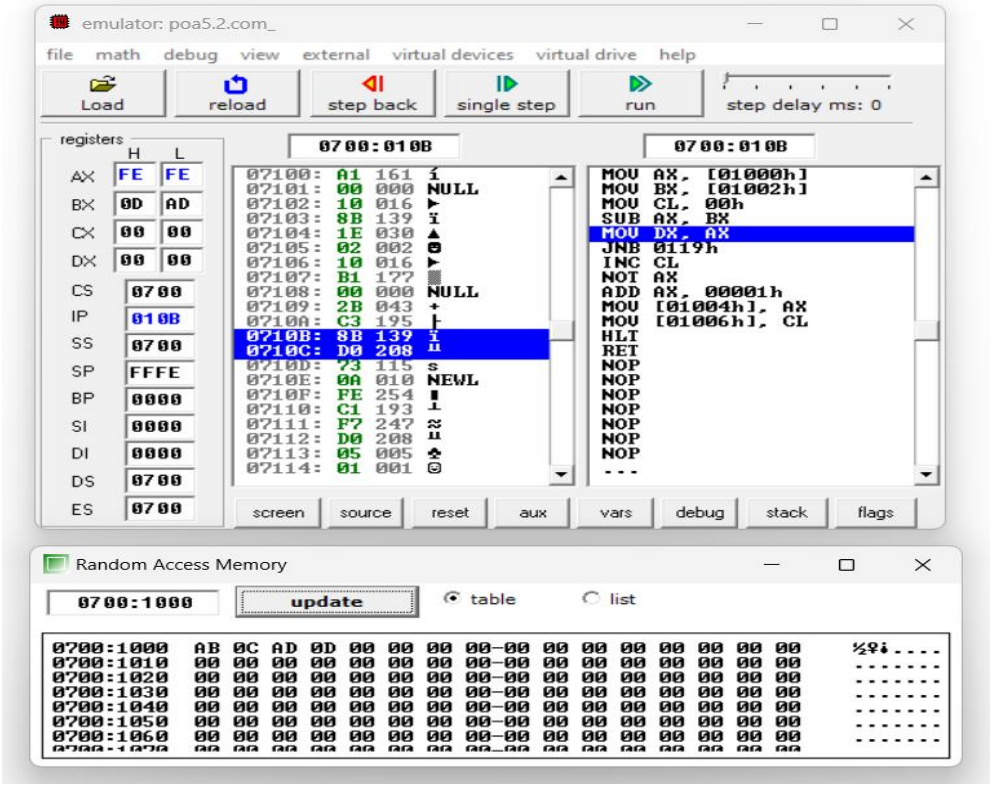


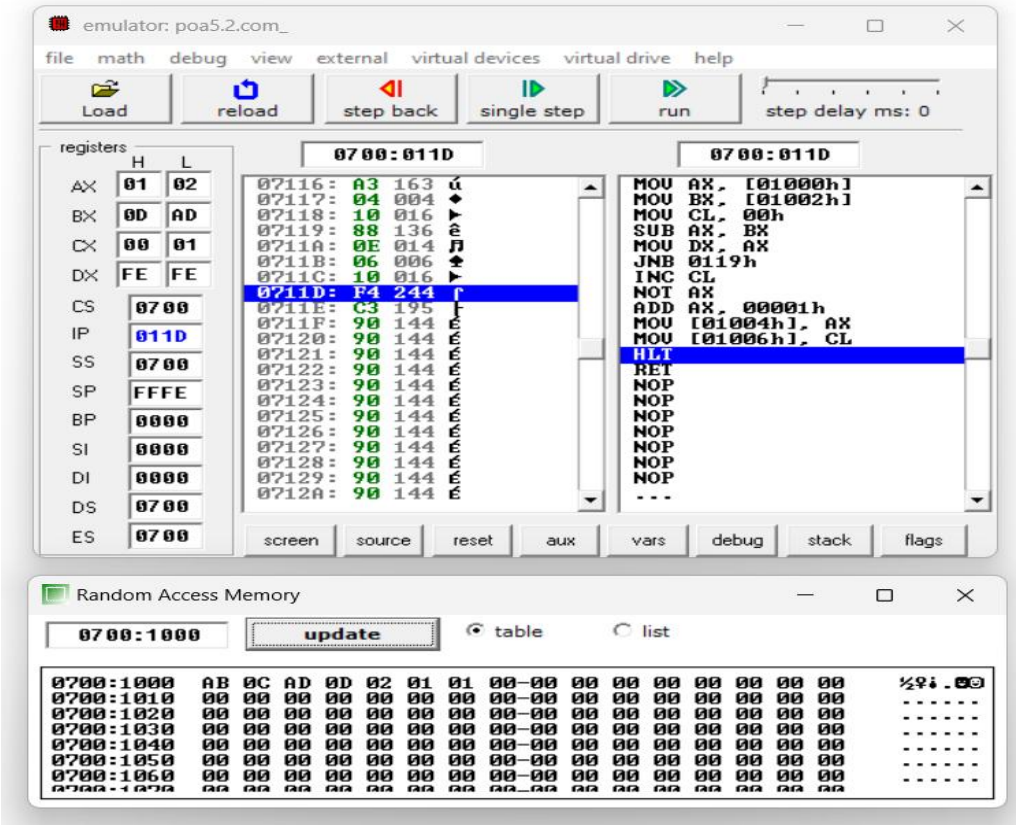
3. **Subtraction using Direct Addressing mode** :

```
org 100h

MOV AX, 0ffffh
MOV BX, 00099h
MOV CL,00h
SUB AX,BX
MOV DX,AX
JNC borrow
INC CL
NOT AX
ADD AX,0001h
MOV [1004h],AX
borrow:
MOV [1006h],CL
HLT
ret
```

- Subtraction of CABh - DADh which gives -102 has the answer which can be seen at address 1004h,1003h





4. **Subtraction using Immediate Addressing mode** :

```
org 100h

MOV AX, 0dadH
MOV BX, 0cabH
MOV CL,00h
SUB AX,BX
MOV DX,AX
JNC borrow
INC CL
NOT AX
ADD AX,0001h
MOV DX,AX
borrow:
MOV [1006h],CL
HLT
ret
```

- Subtraction of DADh - CABh which gives 102 has the answer which can be seen at register DX