**Computer Organization and Architecture (EET2211)**

**LAB I: Examine & Analyze Different Addressing Modes of 8086 Microprocessor**

**Siksha ‘O’ Anusandhan Deemed to be University, Bhubaneswar**

|  |  |  |  |
| --- | --- | --- | --- |
| **Branch: Section:** | | | |
| **S. No.** | **Name** | **Registration No.** | **Signature** |
| **30.** | **Aishwary singh gaharwar** | **1941017093** | **47e2c397-28dd-4f88-878e-8cd4bd5007b3** |

**Marks: \_\_\_\_\_\_/10**

**Remarks:**

**Teacher’s Signature**

**I. OBJECTIVE:**

1. Addition of two 16bit numbers using immediate addressing mode.
2. Addition of two 16bit numbers using direct addressing mode.
3. Addition of two 16bit numbers using indirect addressing mode.
4. Addition of two 16bit numbers using index addressing mode.
5. Addition of two 16bit numbers using base index addressing mode.

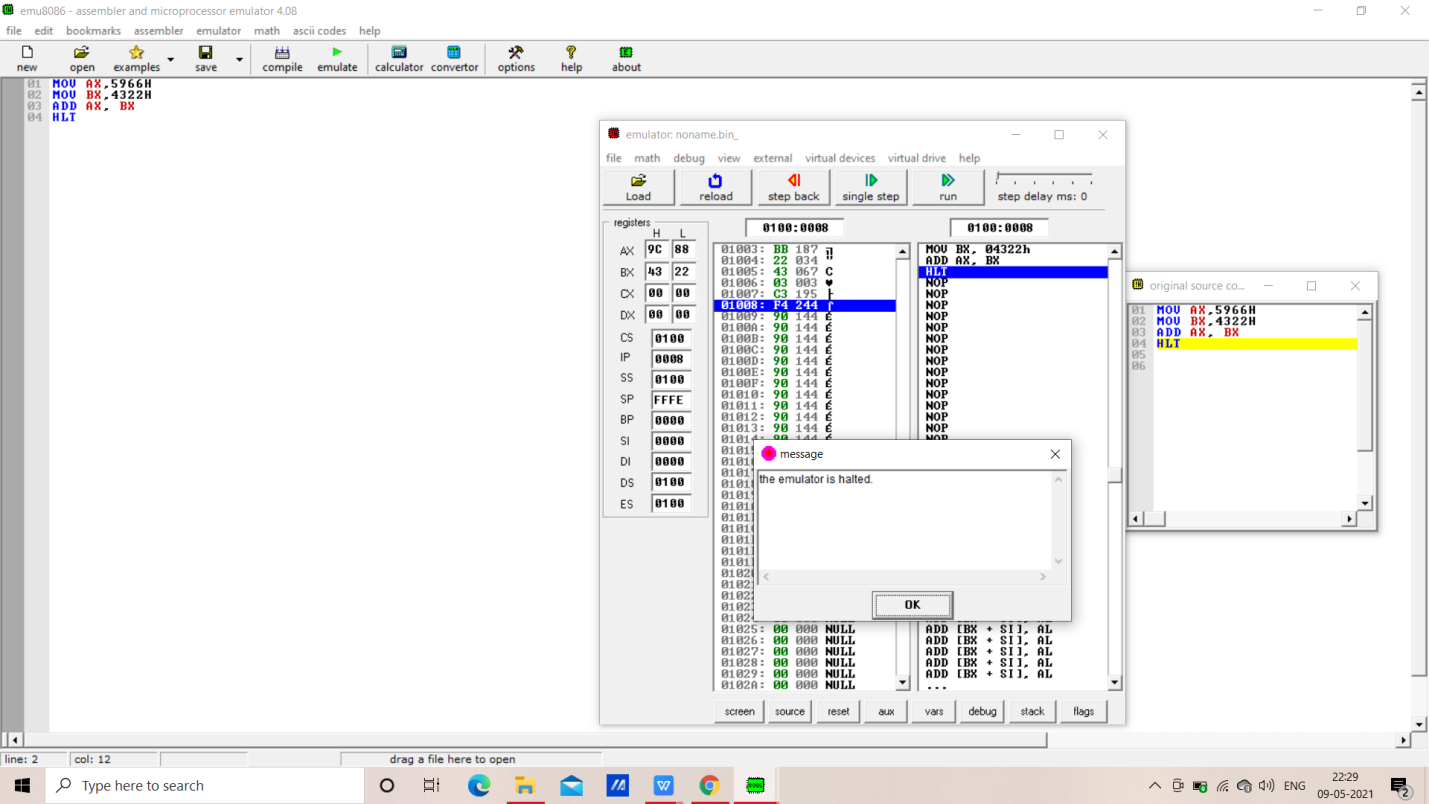
**II. PRE-LAB**

**For Obj. 1:**

a. Explain immediate addressing mode briefly - The mode in which the data operand is a part of the instruction itself is known as immediate addressing mode. Ex- MOV AX, 0005h.

b. Examine & analyze the output obtained from addition of two 16 bit numbers- The Addressing mode In which the data operand is a part of the instruction itself.

EX-MOV AX,0005H(16 bit)



c. Write the assembly code -

MOV AX,5966H

MOV BX,4322H

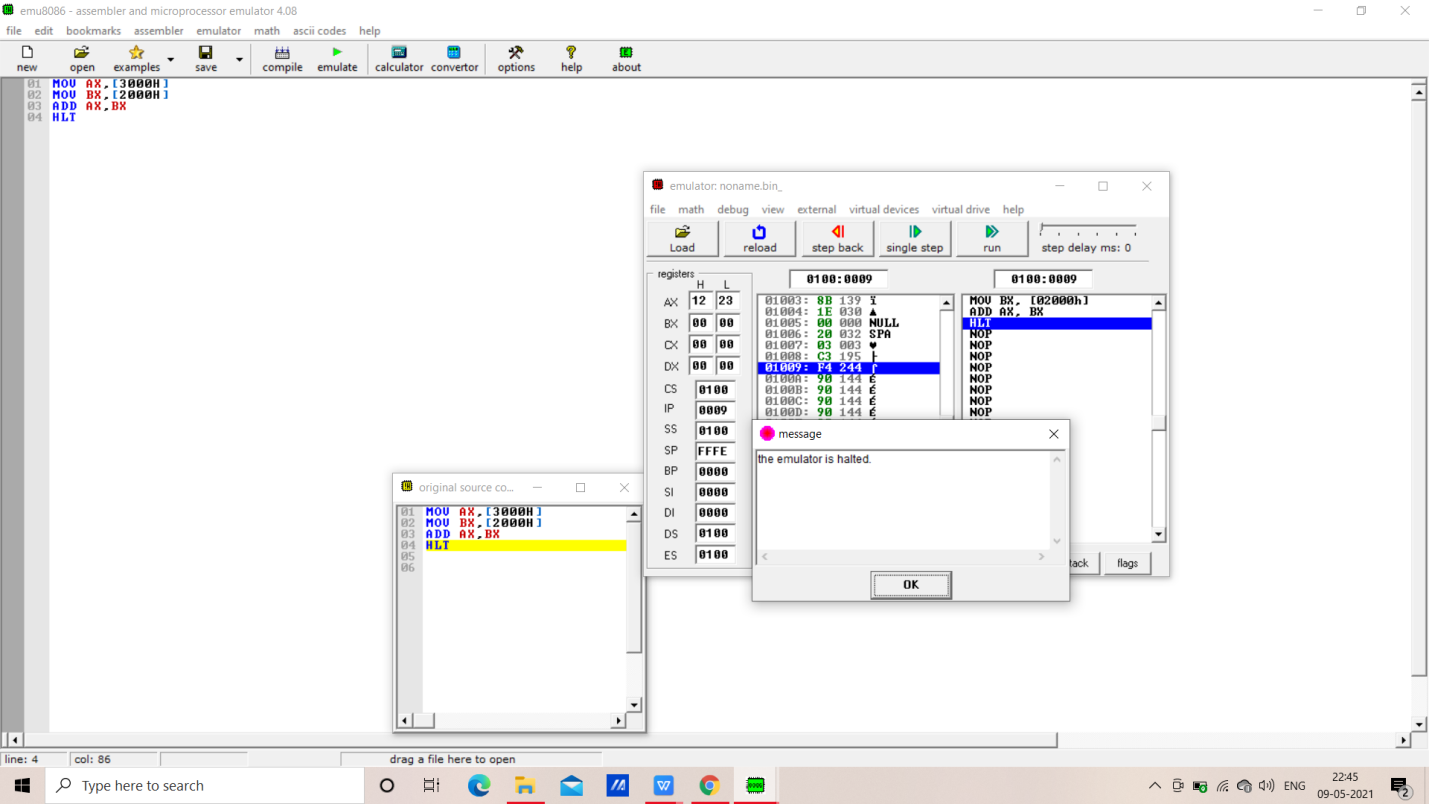
ADD AX, BX

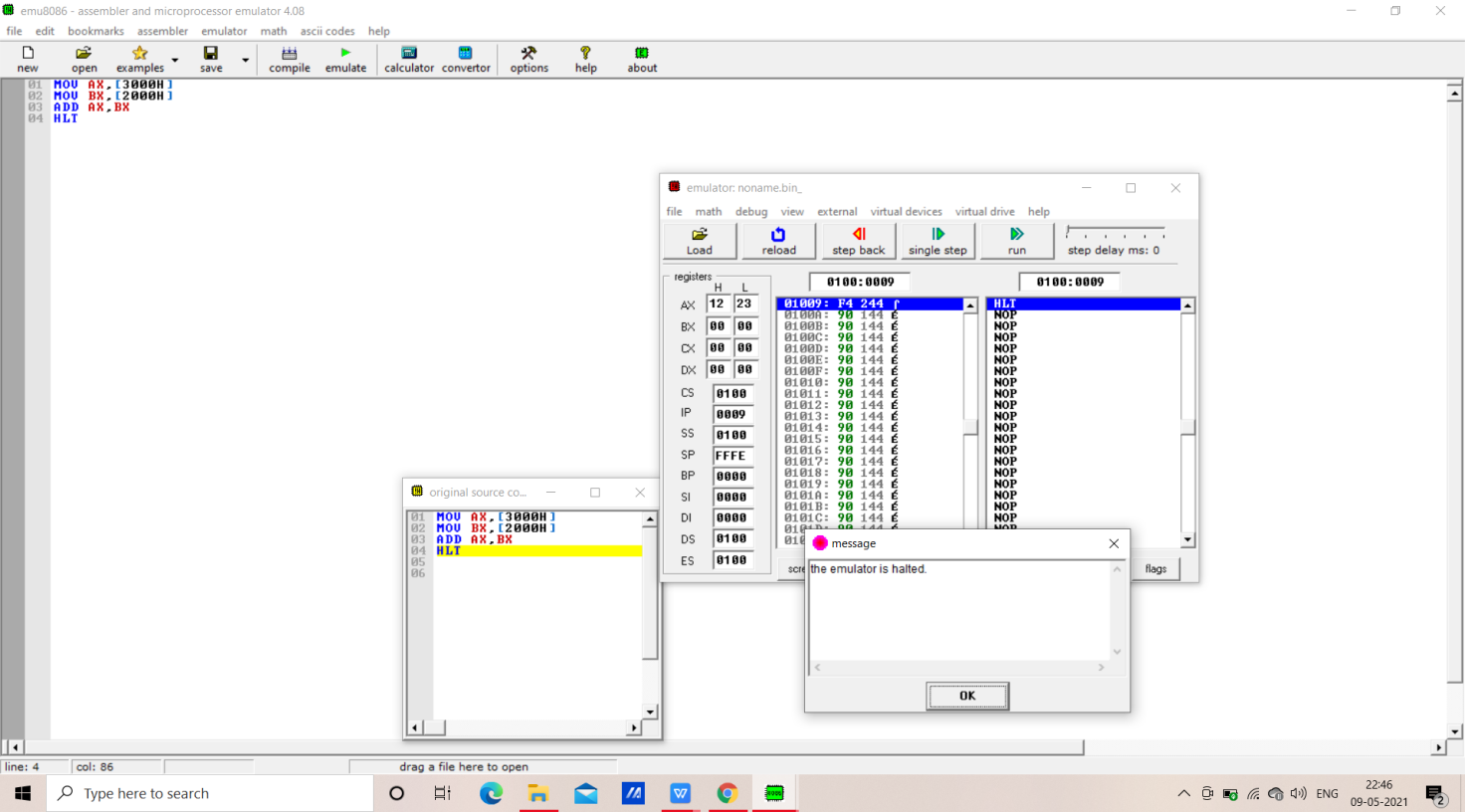
HLT

**For Obj. 2:**

Explain direct addressing mode briefly - The mode in which the effective address of the memory location is written directly in the instruction. Ex- MOV AX, [5000H].

Address {(10\*DS)H+5000H)}

b. Examine & analyze the output obtained from addition of two 16 bit numbers- 



c. Write the assembly code-

*MOV AX,[3000H]*

*MOV BX,[2000H]*

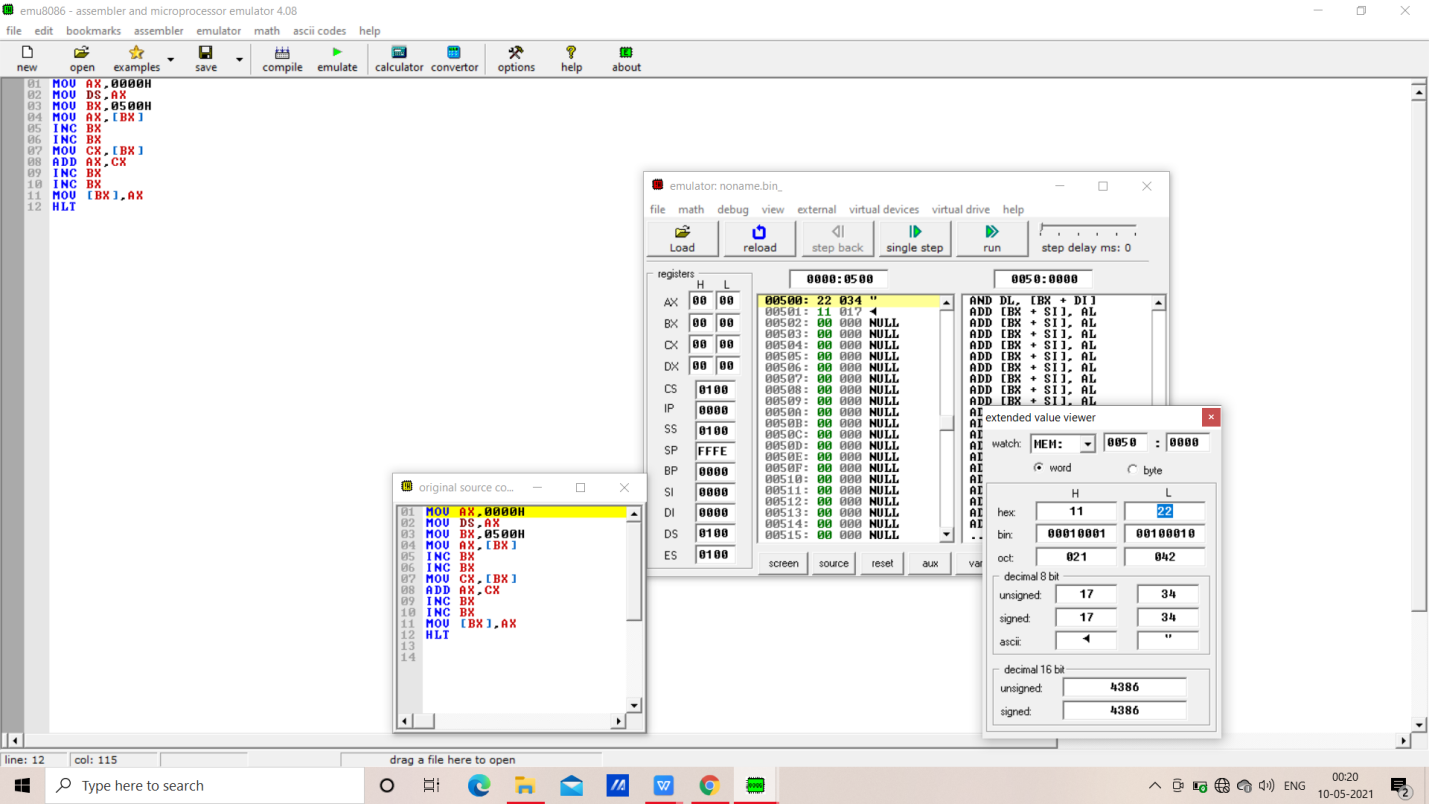
*ADD AX,BX*

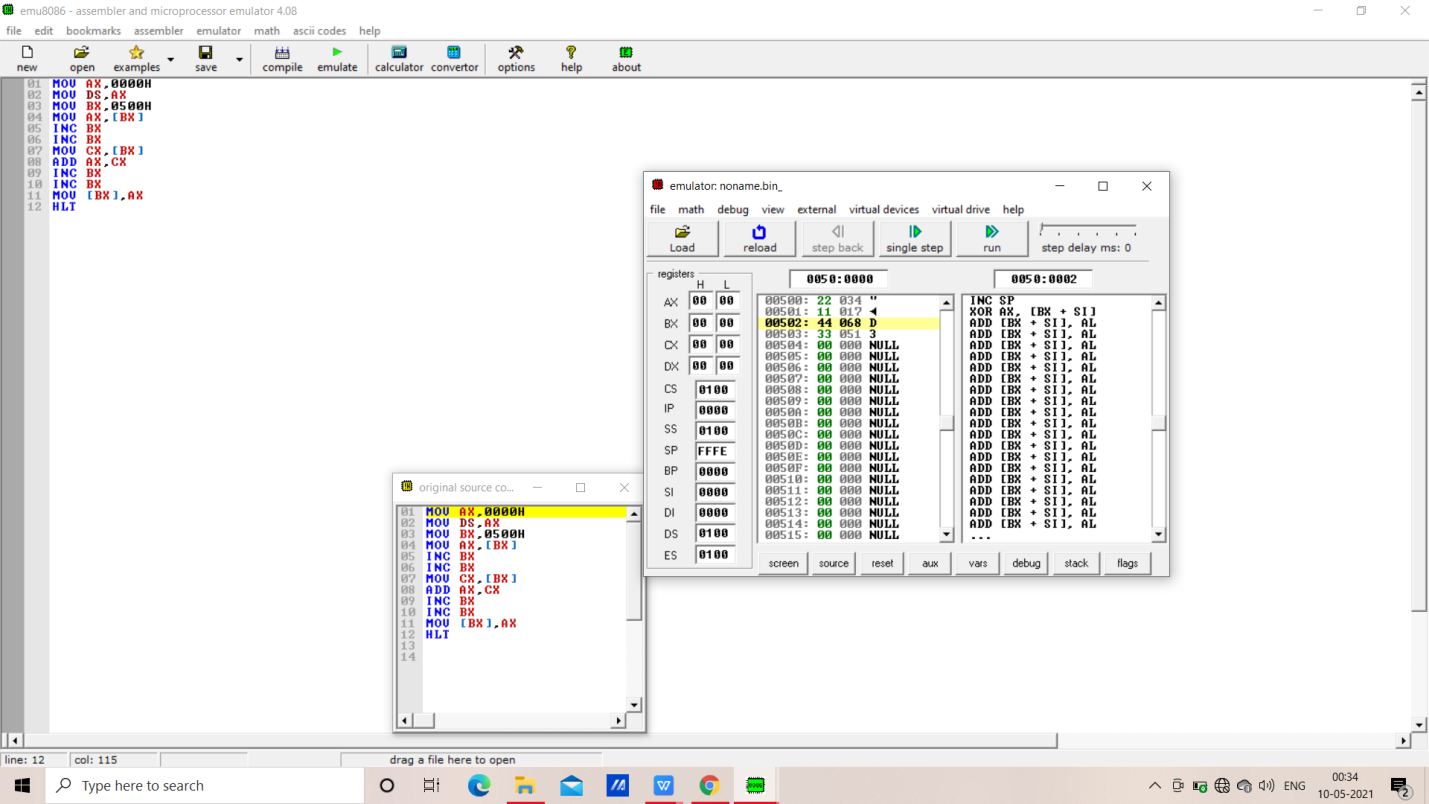
*HLT*

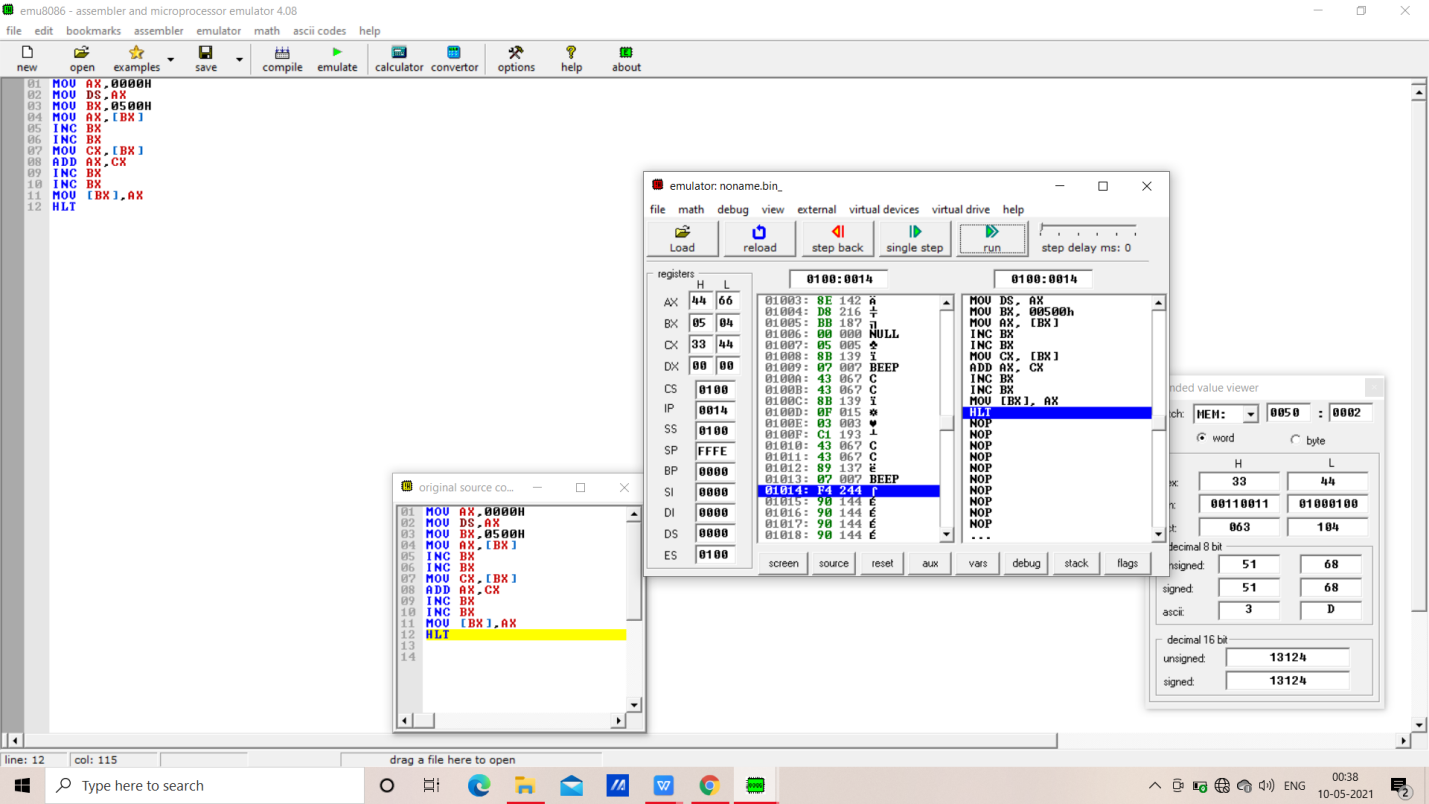
**For Obj. 3:**

a. Explain indirect addressing mode briefly- *The addressing mode allows data to be addressed at any memory location through an offset address held in any of following: BP,BX,DI and SI.*

*EX-MOV AX,[BX]*

b. Examine & analyze the output obtained from addition of two 16 bit numbers- 





c. Write the assembly code-

*MOV AX,0000H*

*MOV DS,AX*

*MOV BX,0500H*

*MOV AX,[BX]*

*INC BX*

*INC BX*

*MOV CX,[BX]*

*ADD AX,CX*

*INC BX*

*INC BX*

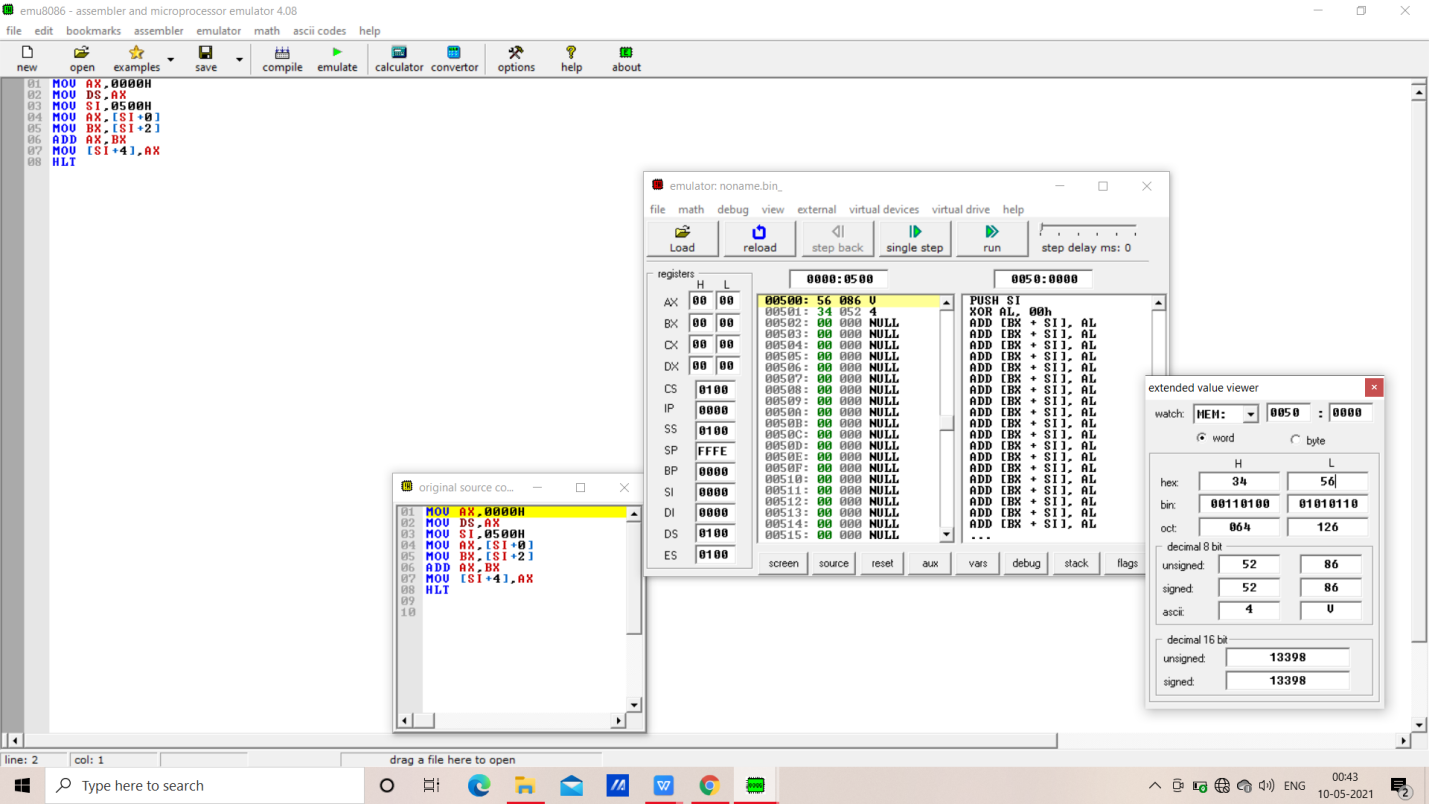
*MOV [BX],AX*

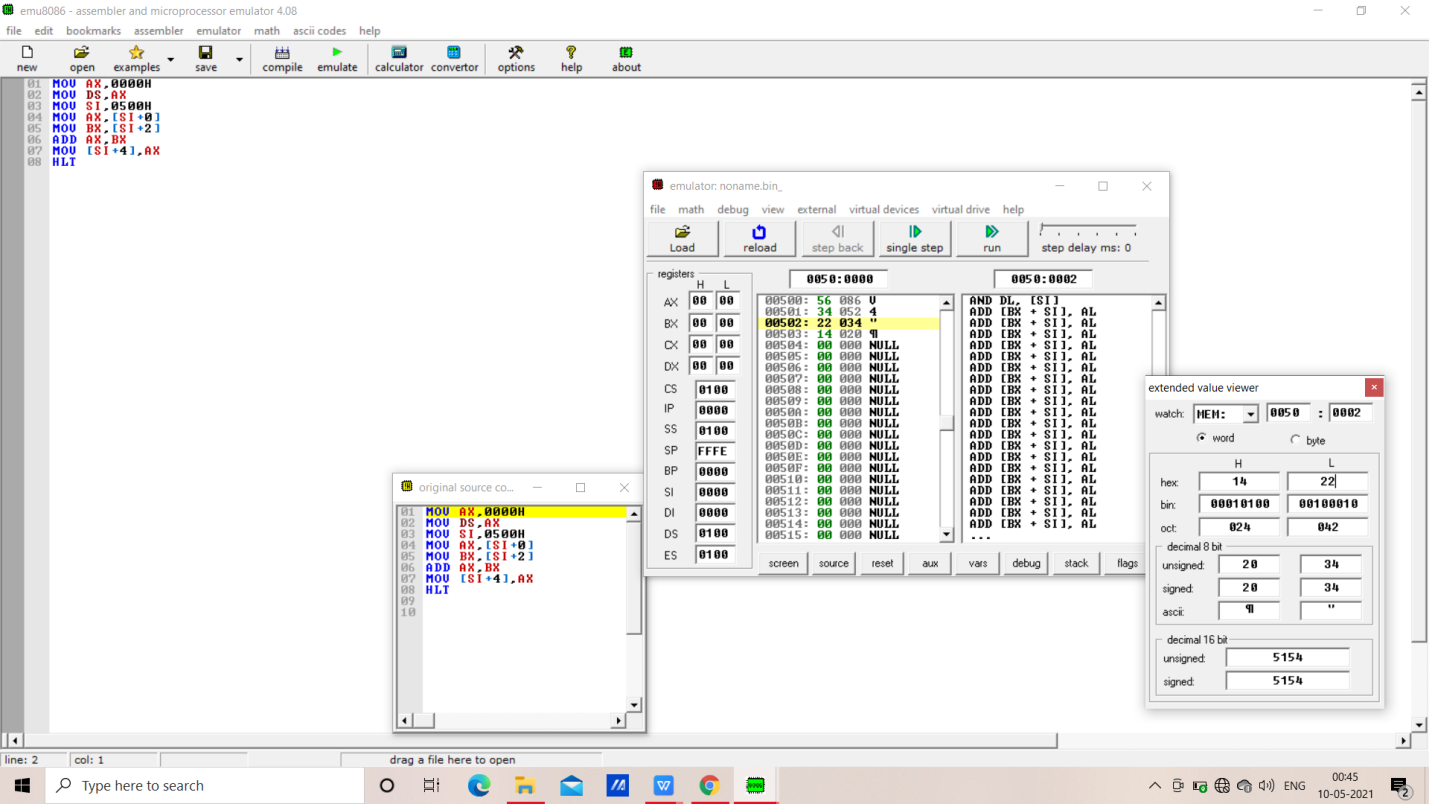
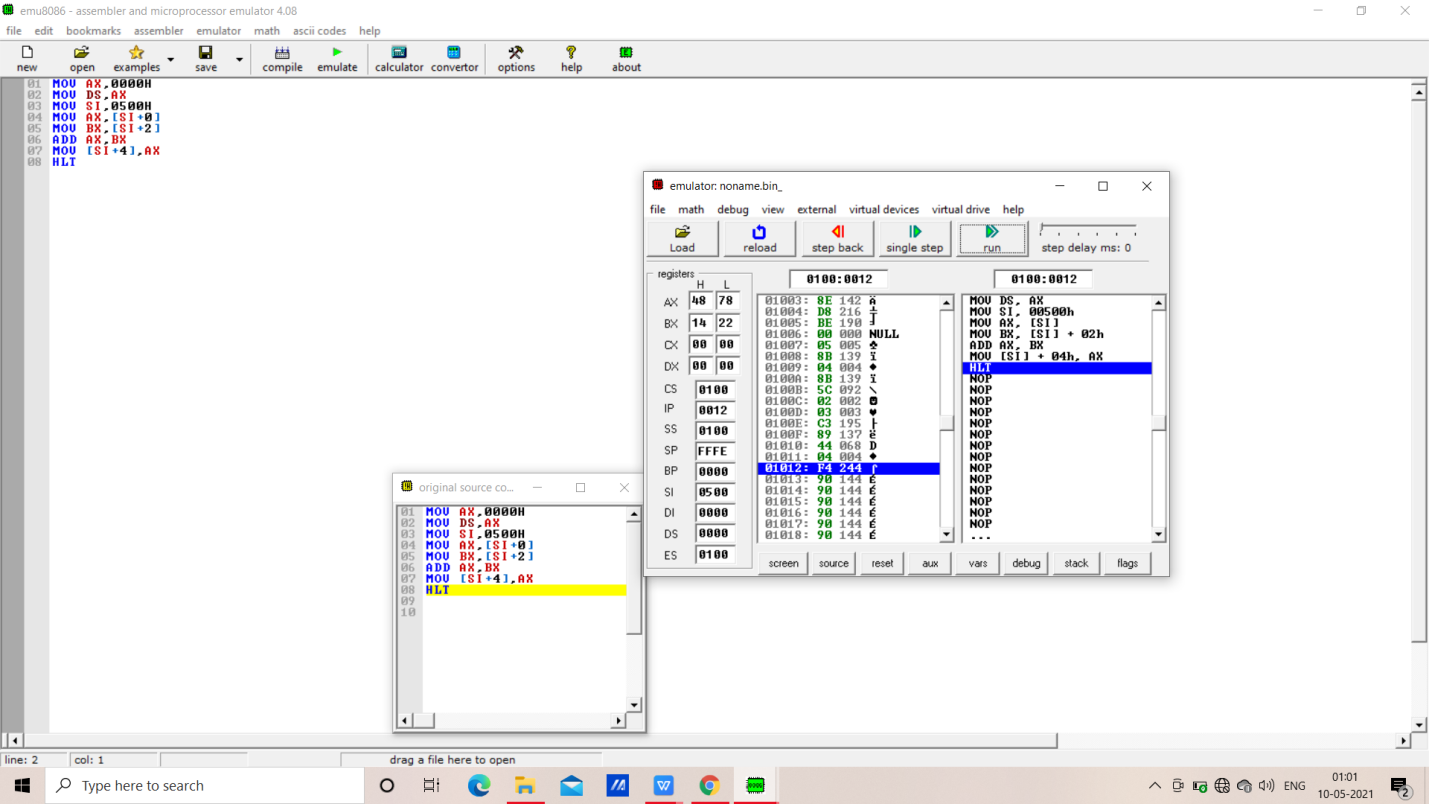
*HLT*

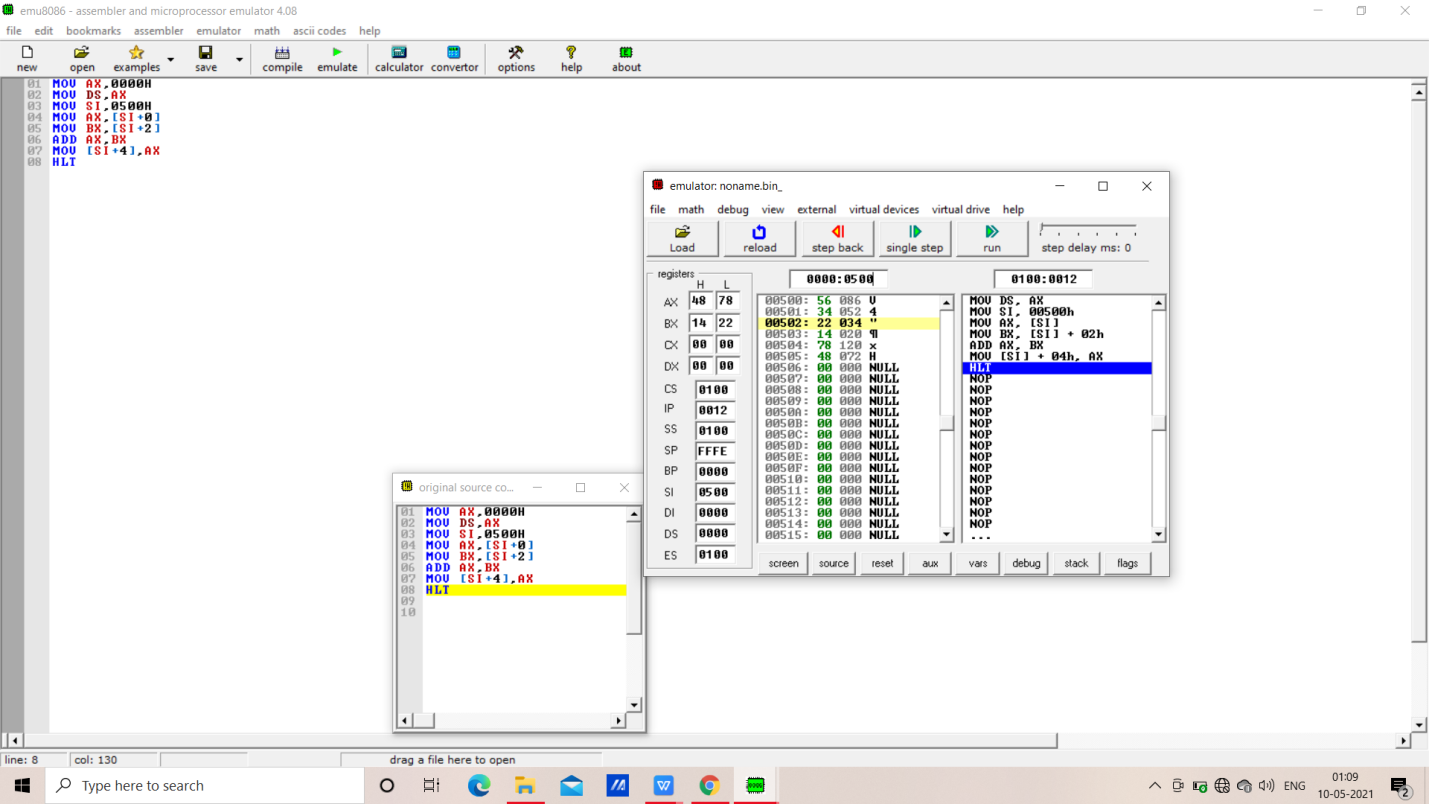
**For Obj. 4:**

a. Explain index addressing mode briefly-*In this addressing mode the operand offset is found by adding the contents of SI or DI register and 8-bit/16-bit displacements.*

*Ex- MOV AX, [SI]*

b. Examine & analyze the output obtained from addition of two 16 bit numbers- 





c. Write the assembly code-

*MOV AX,0000H*

*MOV DS,AX*

*MOV SI,0500H*

*MOV AX,[SI+0]*

*MOV BX,[SI+2]*

*ADD AX,BX*

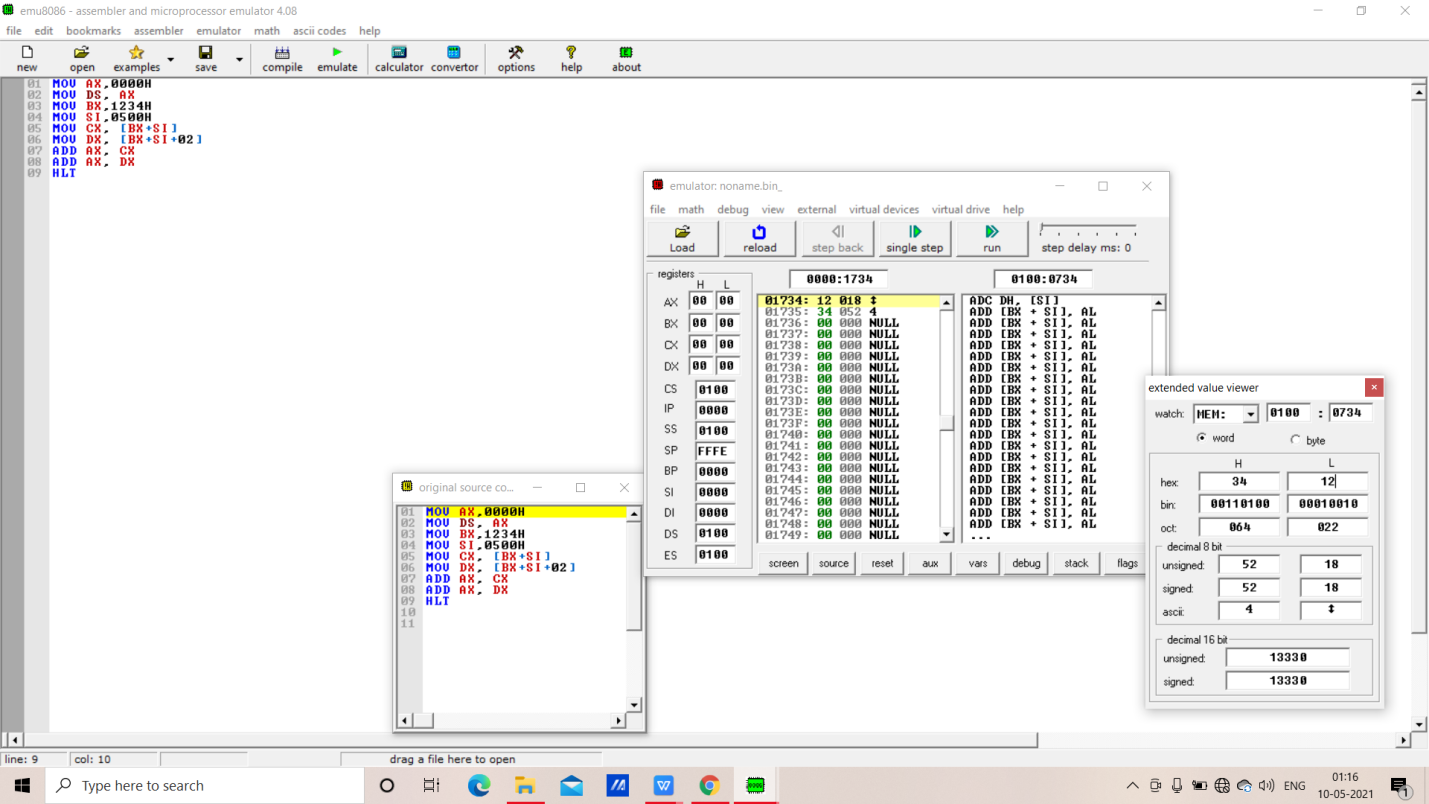
*MOV [SI+4],AX*

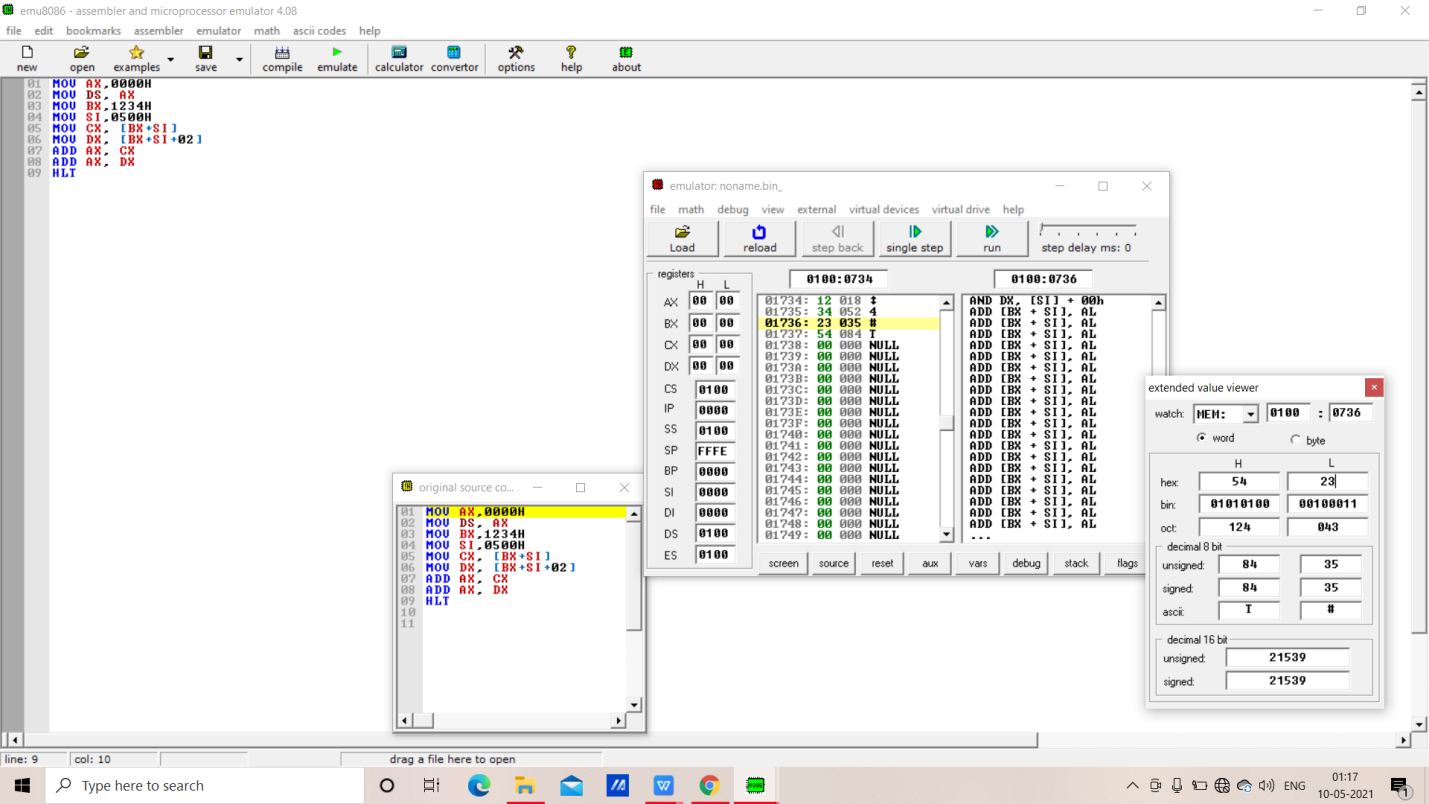
*HLT*

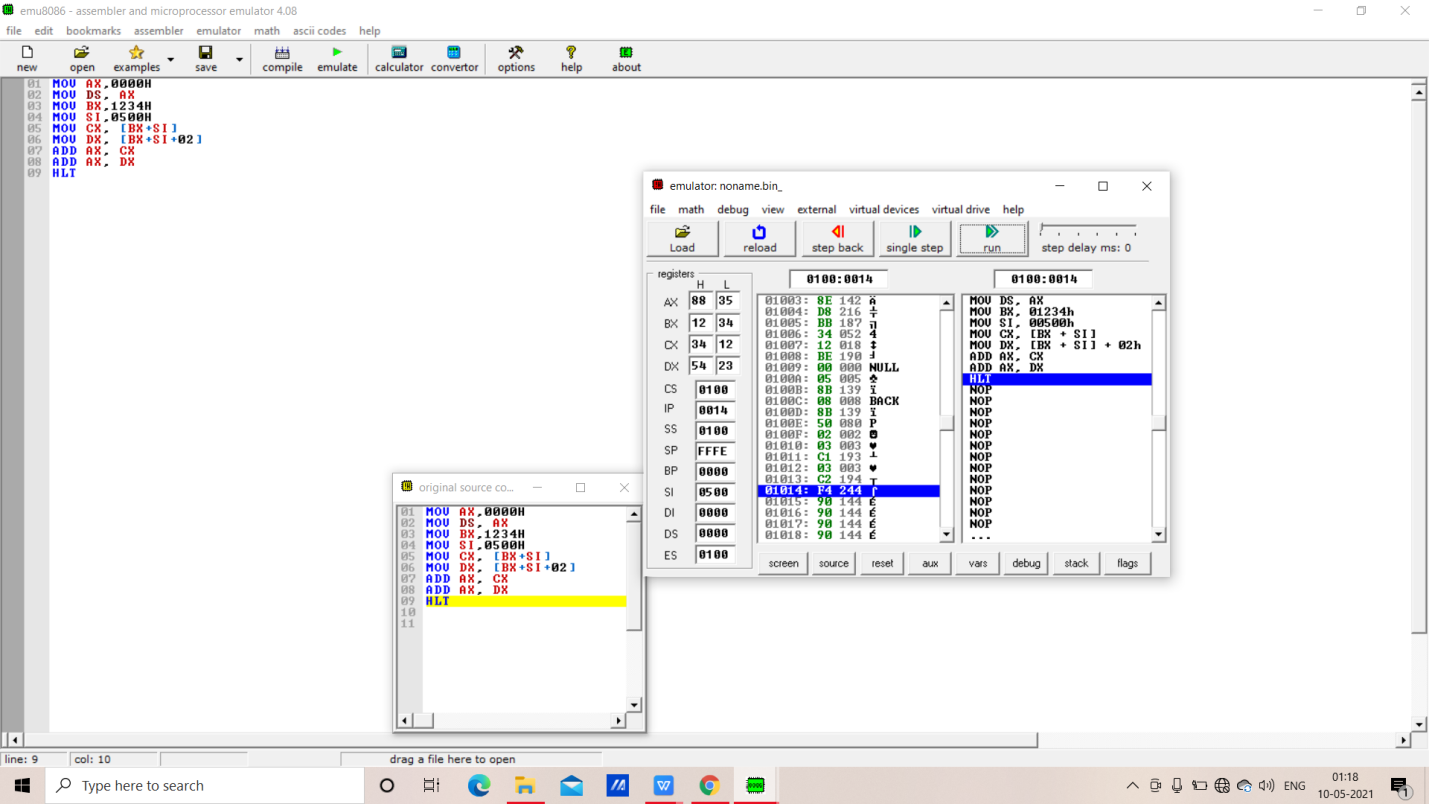
**For Obj. 5:**

a. Explain base index addressing mode briefly- *In this addressing mode the operands offset is computed by adding the base register contents. An index register contains an 8bit or 16bit displacement*

*Ex-MOV AX,50H[BX][SI]*

b. Examine & analyze the output obtained from addition of two 16 bit numbers-





c. Write the assembly code-

*MOV AX,0000H*

*MOV DS, AX*

*MOV BX,1234H*

*MOV SI,0500H*

*MOV CX, [BX+SI]*

*MOV DX, [BX+SI+02]*

*ADD AX, CX*

*ADD AX, DX*

*HLT*

**IV. POST LAB:**

***Discuss different general-purpose registers used in 8086 microprocessor.***

Ans- EU has 8 general purpose registers. Can be individually used for storing 8- bit data. AL register is also called Accumulator. Two registers can also be combined to form 16-bit registers. The valid register pairs are – AX, BX, CX, DX.

AX -) Word multiply, word divide, word I/O

AL -) Byte multiply, byte divide, byte I/O, decimal arithmetic

AH -) Byte multiply, byte divide BX Store address information

CX -) String operation, loops CL Variable shift and rotate

DX -) Word multiply, word divide, indirect I/O (used to hold I/O address during I/O instructions. If the result is more than 16-bits, the lower order 16-bits are stored in accumulator and higher order 6-bitsare stored in DX register)

***Explain the concept of segmented memory. What are its advantages?***

Ans- 8086 has a 20-bit address bus, So it can address a maximum of 1MB of memory. 8086 can work with only four 64KB segments at a time within this 1MB range.

These four memory segments are called:

i) CODE segment

(ii) STACK segment

(iii) DATA segment

(iv) EXTRA segment

CODE SEGMENT- The part of memory from where BIU is currently fetching instruction code bytes. It is used for storing the instructions.

STACK SEGMENT- A section of memory set aside to store address and data while a subprogram executes. It is sued as a stack and is used to store the return address.

DATA AND EXTRA SEGMENTS -Used for storing data values or data bytes to be used in the program

The size of each segment is 64 KB. A segment may be located any-where in the memory. Each of these segments can be used for a specific function.

Address of a segment is of 20-bits. A segment register stores only upper 16 bits of the starting address of the corresponding segments. The 1-bit contents of the segment registers in the BIU actually points to the starting location of a particular segment always inserts zeros for the lowest 4-bits of the 20-bit starting address. A 64-KB segment can be located anywhere in the memory, bus will start at an address with zero in the lowest 4-bits.

***Explain the physical address formation in 8086.***

Ans-) Offset value + Segment Register = Physical Address

Suppose, cs-34BA

IP-8AB4

Physical Address-34BA0 +

8AB4

3D654

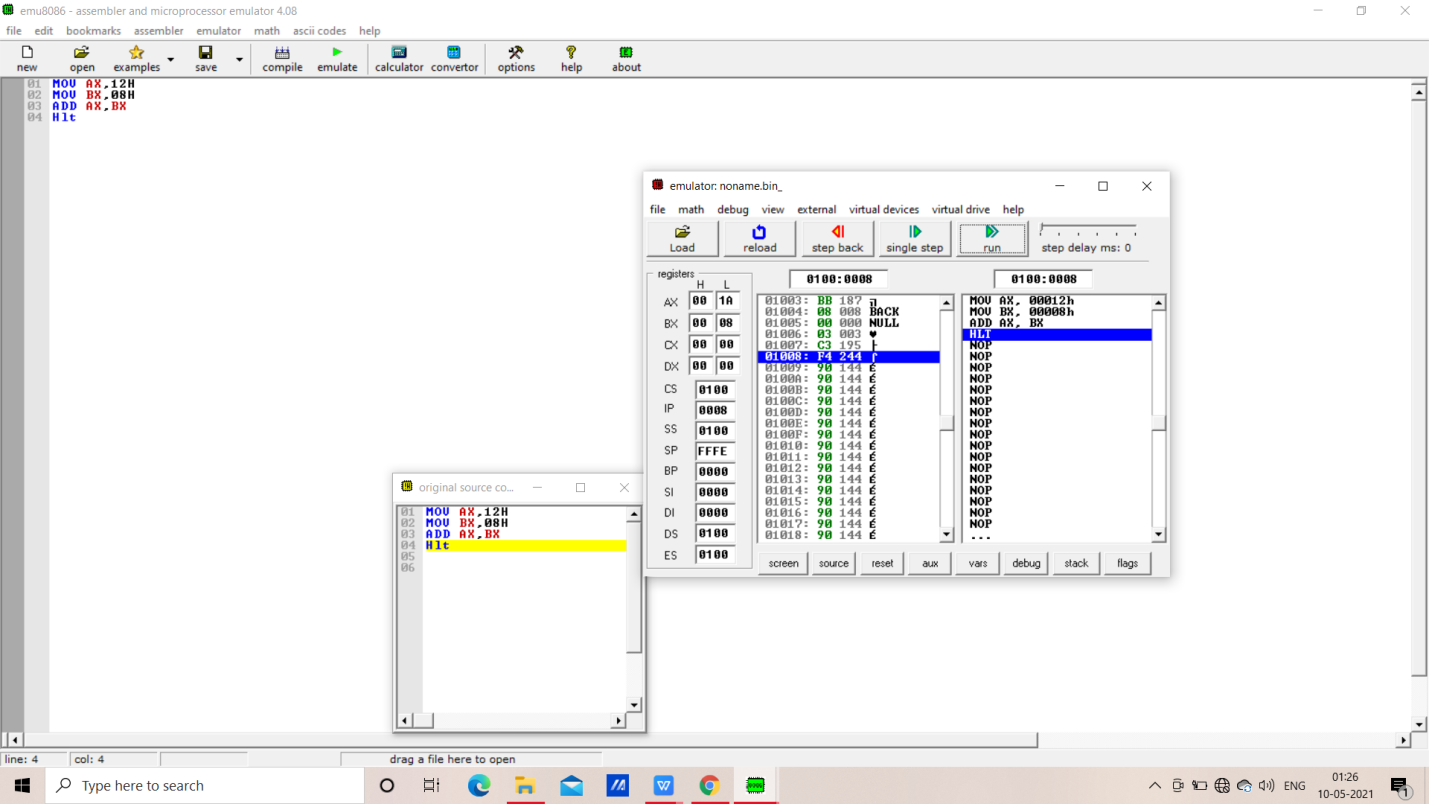
Write a program to add two 16 bit numbers 12H and 08H, and store the sum.

MOV AX,12H

MOV BX,08H

ADD AX,BX

HLT



**IV. POST LAB:**