

MICROPROCESSOR-BASED SYSTEMS DESIGN

UCS617

Lab Assignment-2 (8086)

Submitted to:

Dr Shivani Sharma

Submitted By:

Dixant Kumar 102103017

Jagdish Agarwal 102103266

Pulkit Arora 102103267

Jeetesh Rajpal 102103268

Janardhan Singh Jadon 102103269



BE Third Year, COE

Group No: 3COE10

Computer Science and Engineering Department

TIET, Patiala

INDEX

Sr. No.	Title	Page
1.	Write an assembly language program to add two 16-bit numbers in 8086.	1
2.	Write an assembly language program to subtract two 16-bit numbers in 8086.	2
3.	Write an assembly language program to multiply two 16-bit numbers in 8086.	3
4.	Write an assembly language program to divide two 16-bit numbers in 8086.	5
5.	Write an assembly language program to demonstrate AAA, AAS, AAM, AAD, DAA and DAS in 8086.	6
6.	Write an assembly language program to find out the count of positive numbers and negative numbers from a series of signed numbers in 8086.	13
7.	Write an assembly language program to find out the largest number from a given unordered array of 8-bit numbers, stored in the locations starting from a known address in 8086.	15
8.	Write an assembly language program to find out the largest number from a given unordered array of 16-bit numbers, stored in the locations starting from a known address in 8086.	17
9.	Write an assembly language program to print Fibonacci series in 8086.	19
10.	Write an assembly language program to perform the division 15/6 using the ASCII codes. Store the ASCII codes of the result in register DX.	21

Program 1

Aim: Write an assembly language program to add two 16-bit numbers in 8086.

Sol:

MOV AX,1234H

MOV BX,1236H

ADD AX,BX

HLT

Output:

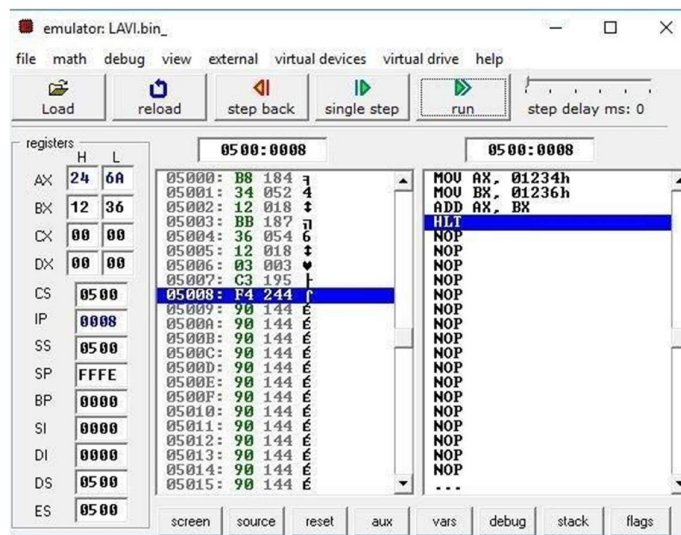


Fig 1.1

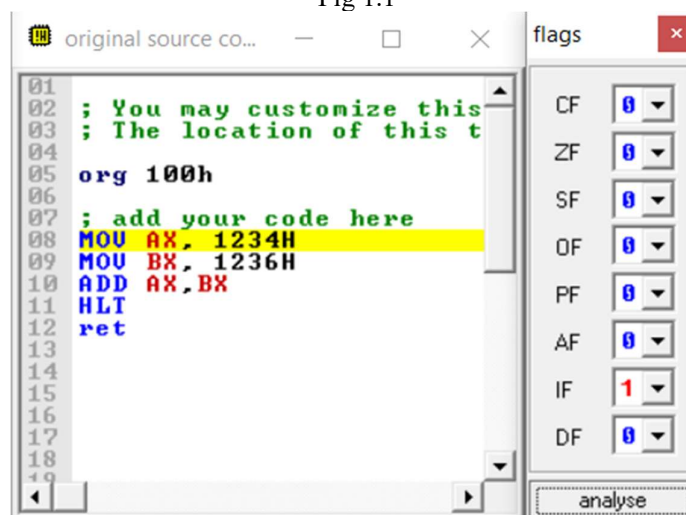


Fig 1.2

Program 2

Aim: Write an assembly language program to subtract two 16-bit numbers in 8086.

Sol:

MOV AX, 1234H

MOV BX, 1236H

SUB AX, BX

HLT

Output:

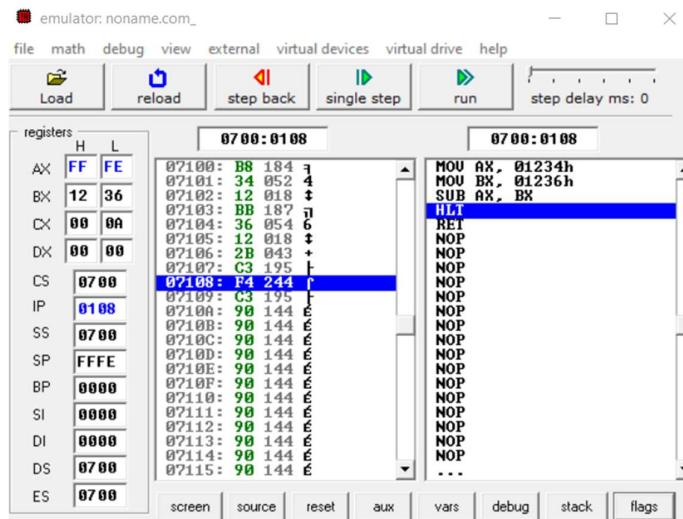


Fig 2.1

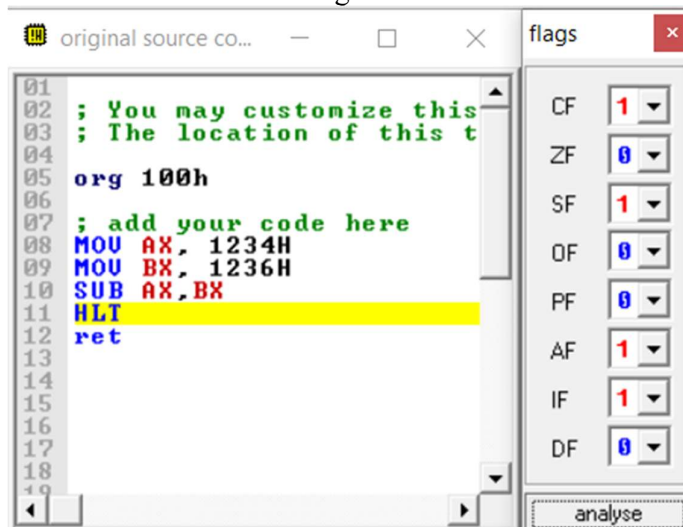


Fig 2.2

Program 3

Aim: Write an assembly language program to multiply two 16-bit numbers in 8086.

Sol:

MOV AX, [0301H]

MOV BX, [0303H]

MUL BX

HLT

Input:

Fig 3.1

[illegible]

Fig 3.2

Output:

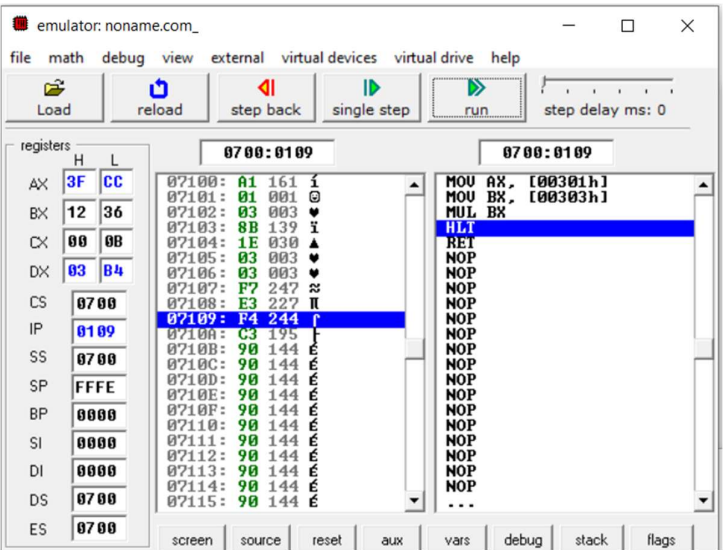


Fig 3.3

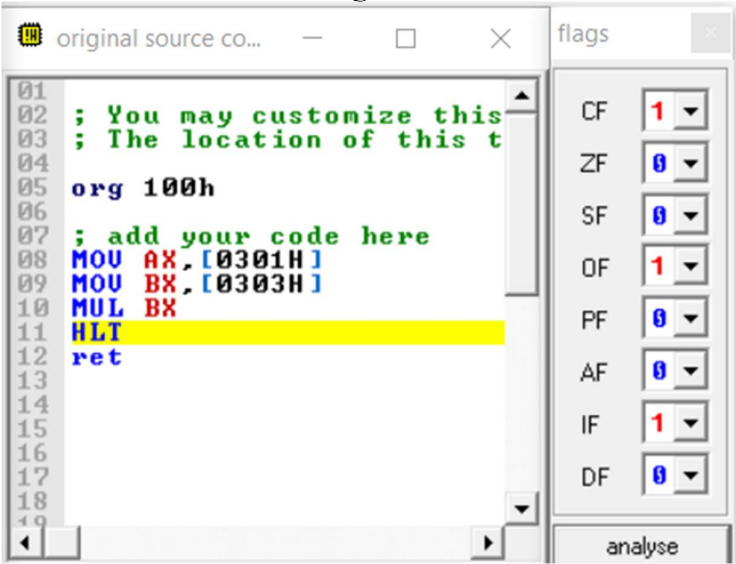


Fig 3.4

Program 4

Aim: Write an assembly language program to divide two 16-bit numbers in 8086.

Sol:

MOV AX,5600H

MOV BX,2500H

DIV BX

HLT

Output:

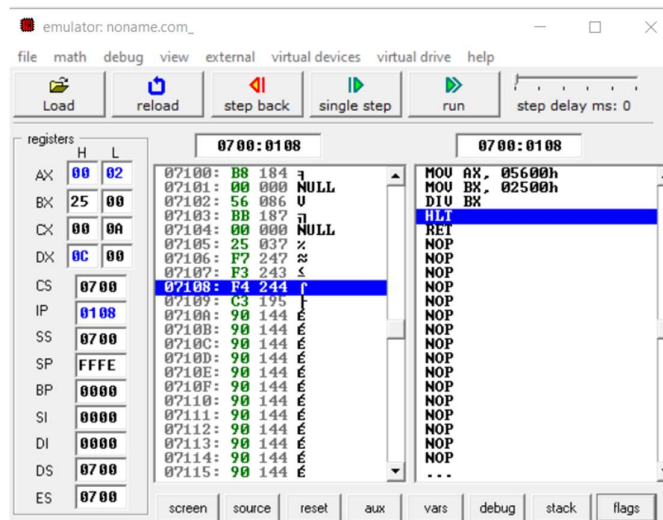


Fig 4.1

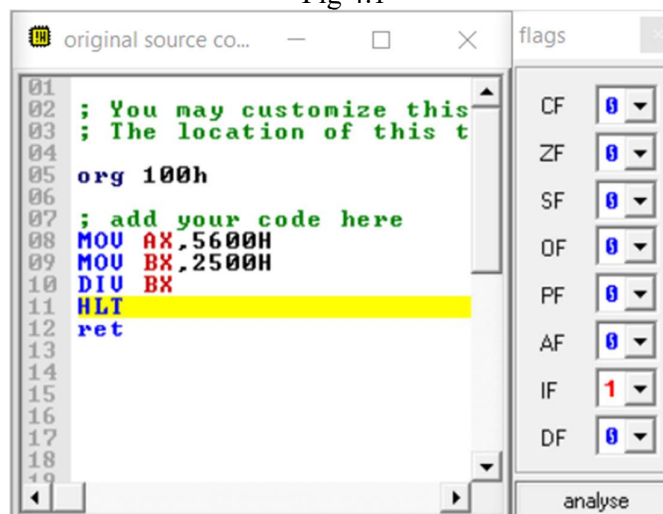


Fig 4.2

Program 5

Aim: Write an assembly language program to demonstrate AAA, AAS, AAM, AAD, DAA and DAS in 8086

Sol:

AAA	AAS
MOV AX,0032H MOV BX,0033H ADD AX,BX AAA HLT	MOV AL,0033H SUB AX,0039H AAS OR AL,0030H HLT

AAM	AAD
MOV AL,03H MOV BL,09H MUL BL AAM OR AX,3030H HLT	MOV AX,0033H MOV BX,0032H AAD DIV BX HLT

DAA	DAS
MOV AL,71H ADD AL,43H DAA HLT	MOV AL,71H SUB AL,43H DAS HLT

AAA:

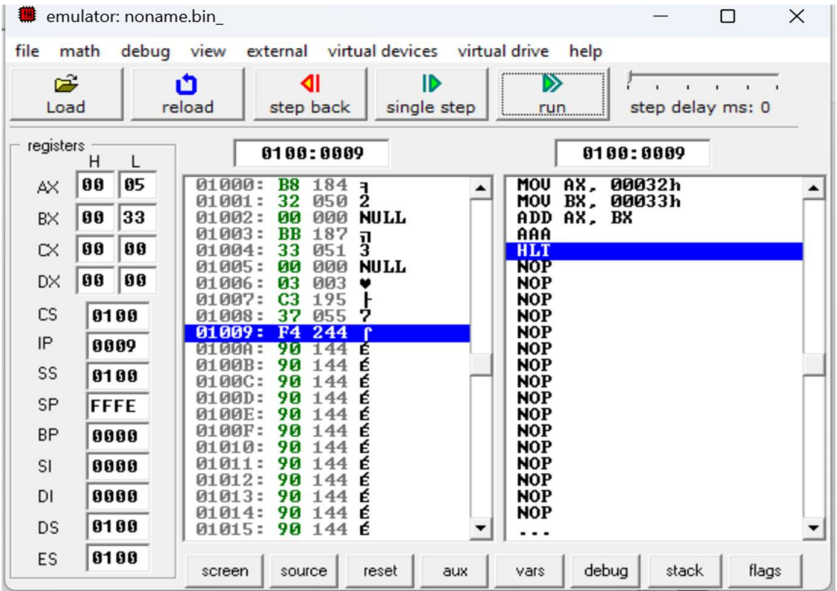


Fig 5.1

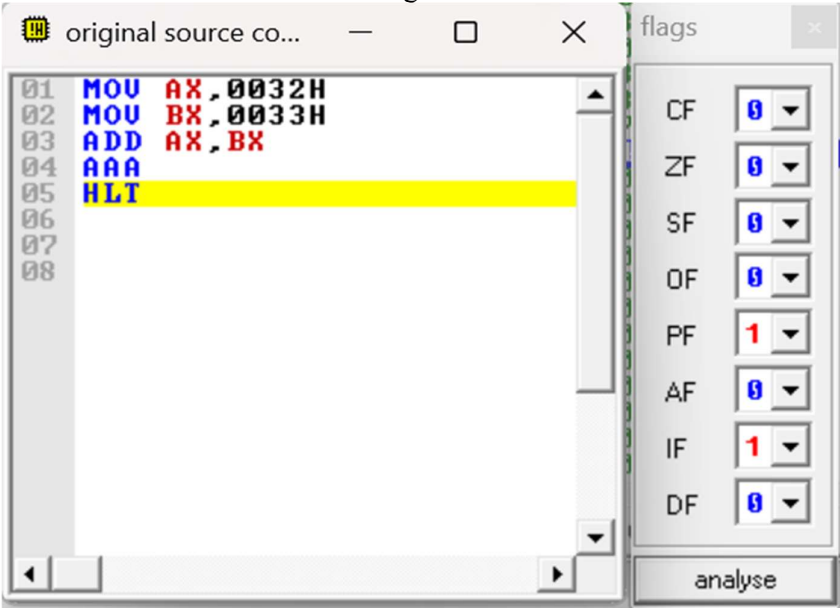


Fig 5.2

AAS:

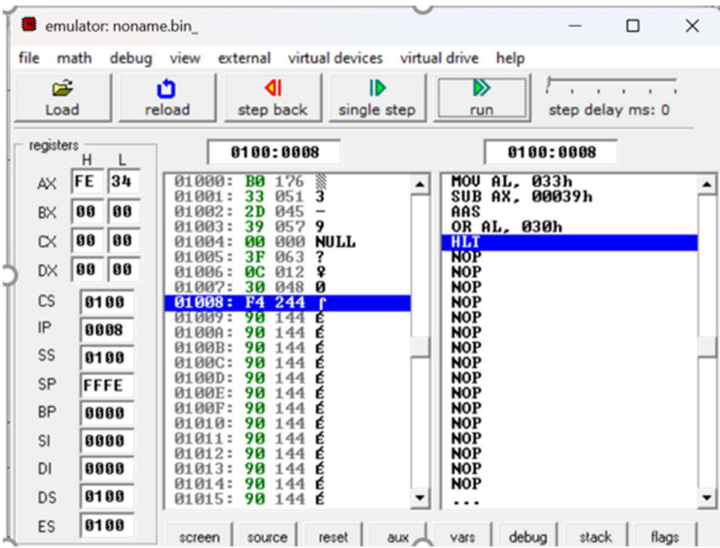


Fig 5.3

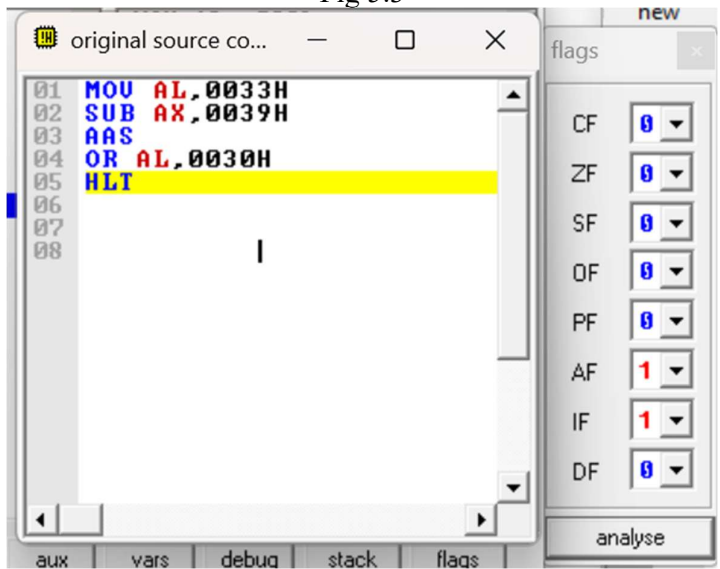


Fig 5.4

AAM:

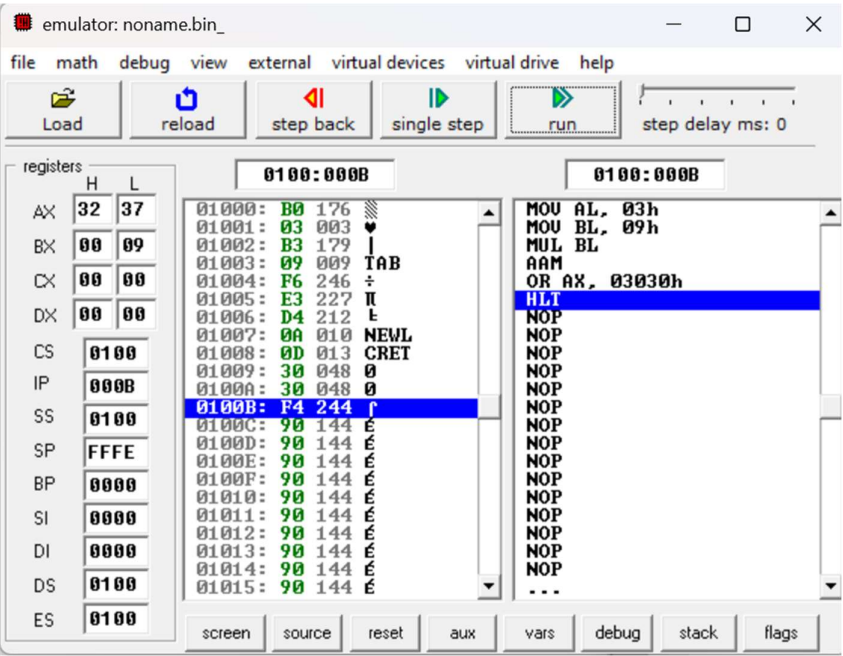


Fig 5.5

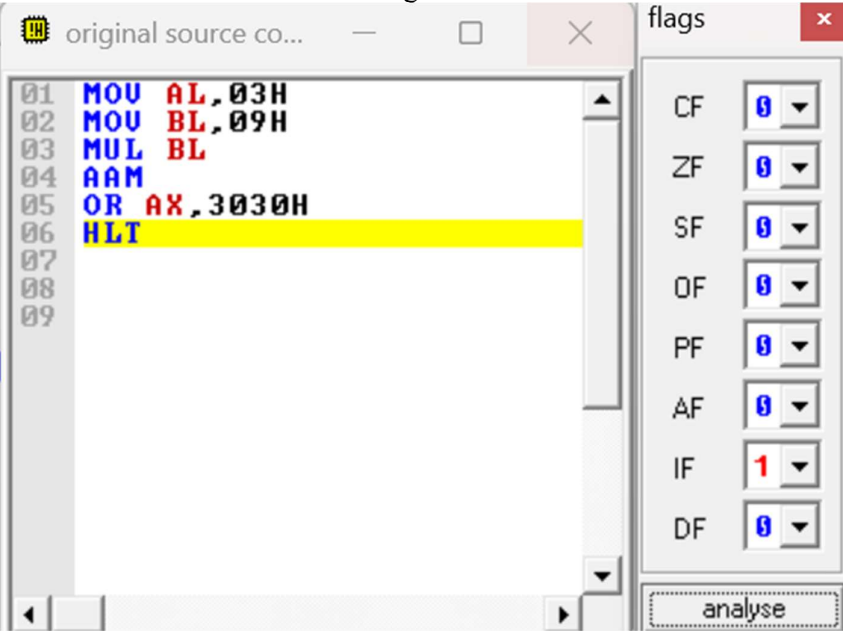


Fig 5.6

AAD:

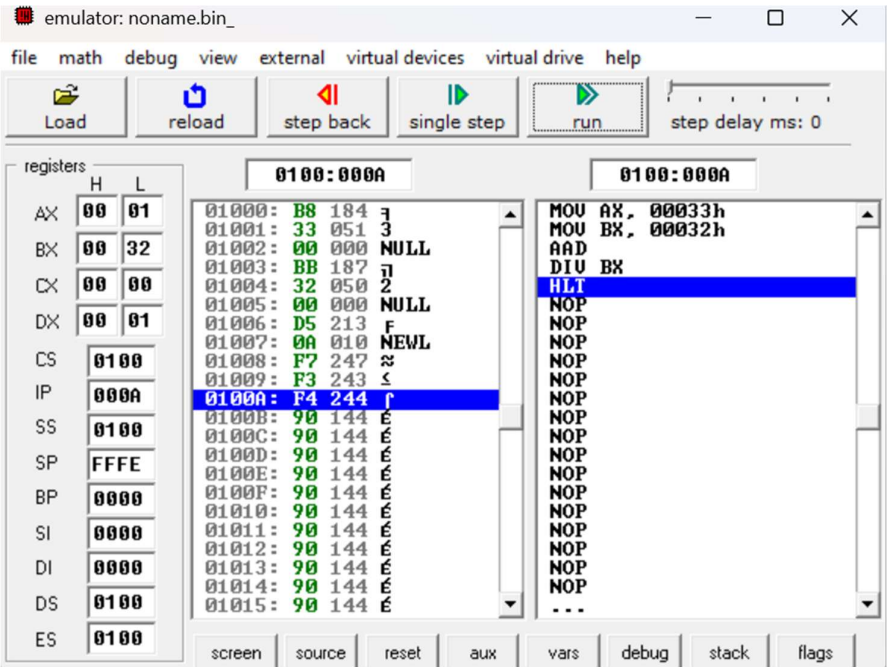


Fig 5.7

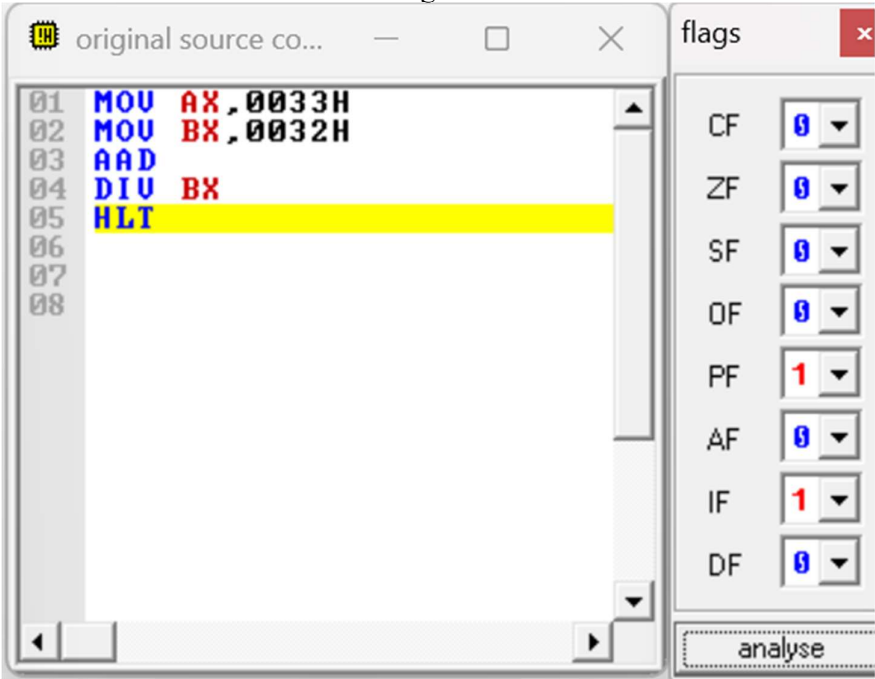


Fig 5.8

DAA:

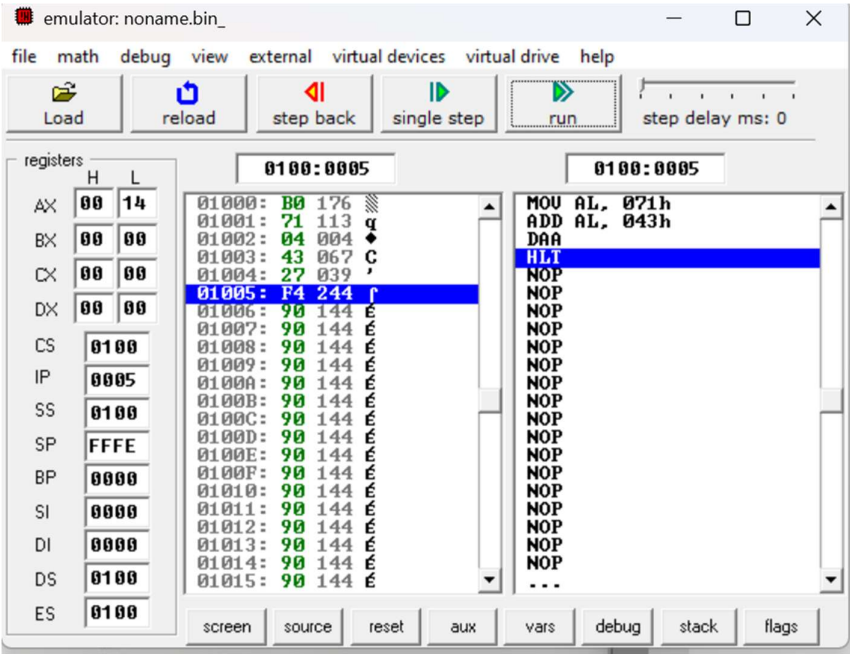


Fig 5.9

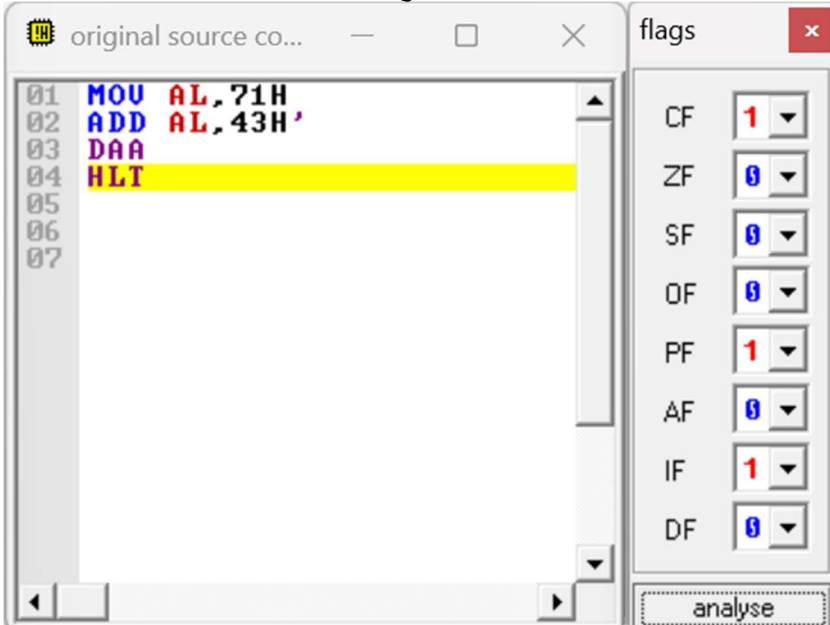


Fig 5.10

DAS:

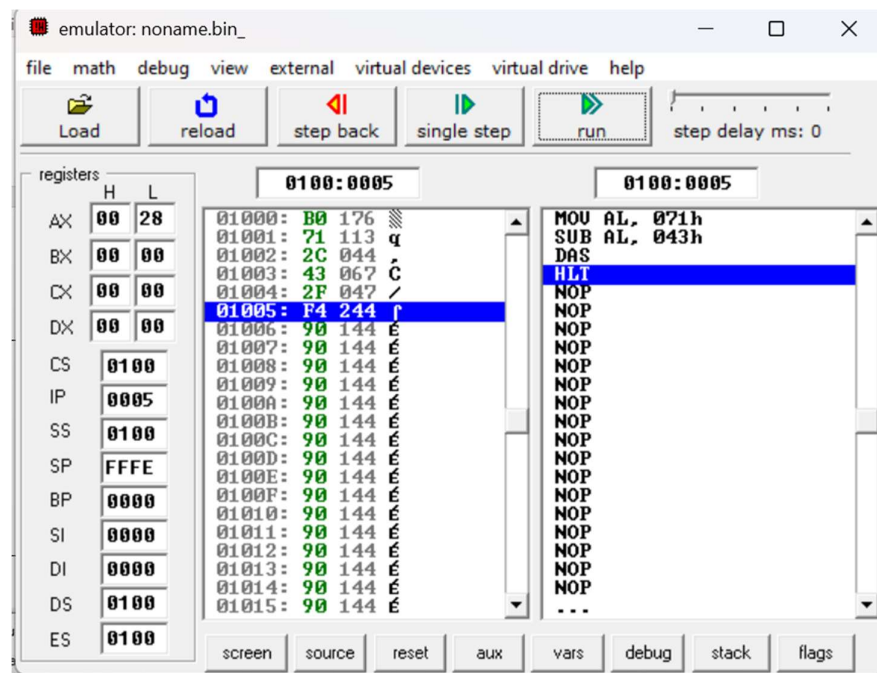


Fig 5.11

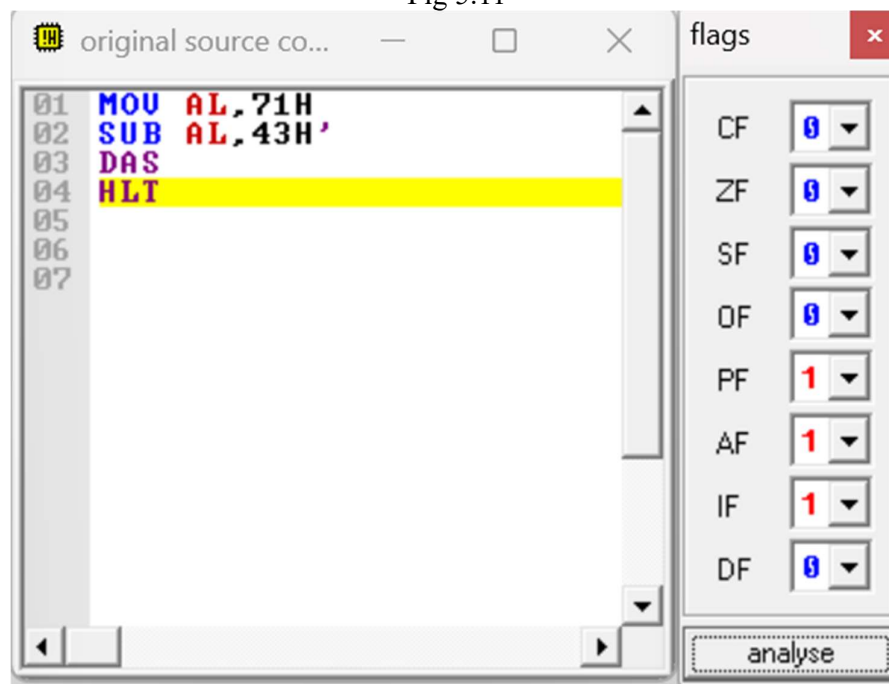


Fig 5.12

Program 6

Aim: Write an assembly language program to find out the count of positive numbers and negative numbers from a series of signed numbers in 8086.

Sol:

```
MOV CL,0AH
MOV BL,00H
MOV DL,00H
LEA SI, [1000H]
L1: MOV AL, [SI]
SHL AL, 01
JNC L2
INC DL
JMP L3
L2: INC BL
L3: INC SI
DEC CL
JNZ L1
MOV [100AH], BL
MOV [100BH], DL
HLT
```

Input:

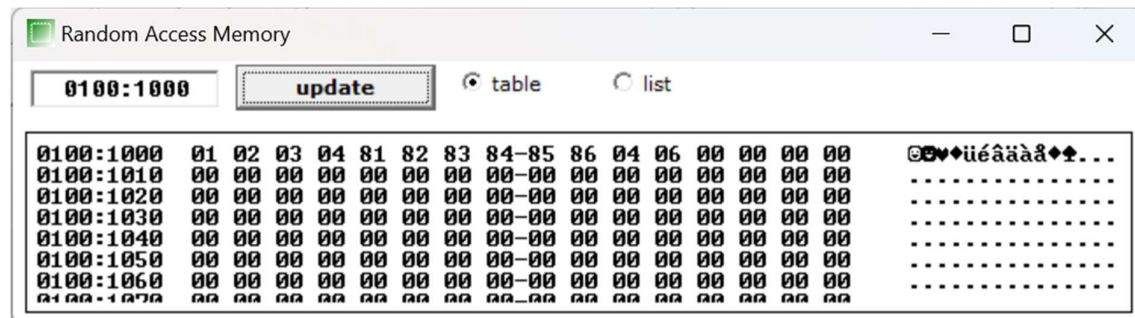


Fig 6.1

Output:

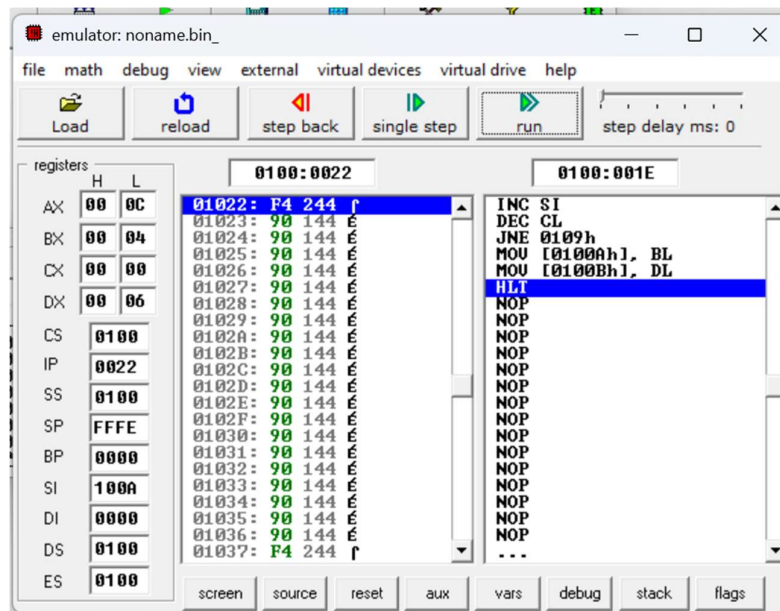


Fig 6.2

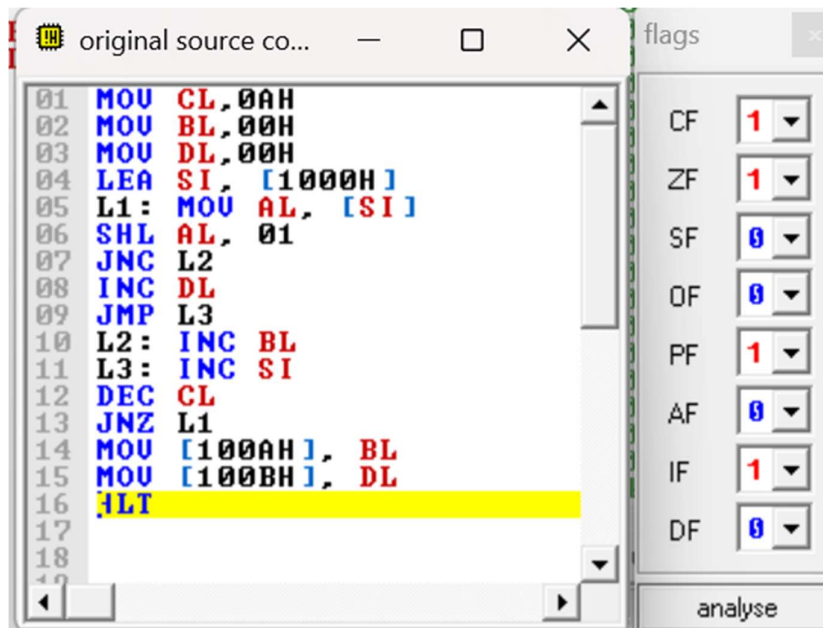


Fig 6.3

Program 7

Aim: Write an assembly language program to convert to find out the largest number from a given unordered array of 8-bit numbers, stored in the locations starting from a known address in 8086.

Sol:

```
MOV CL, 0AH
LEA SI, [1000H]
MOV AL, [SI]
L1: INC SI
MOV BL, [SI]
CMP AL, BL
JC L2
JMP L3
L2: MOV AL, BL
L3: DEC CL
JNZ L1
MOV [100AH], AL
HLT
```

Input:

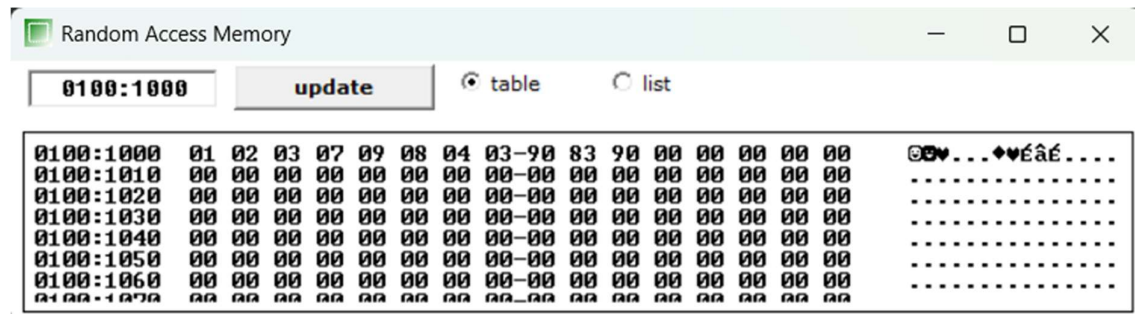


Fig 7.1

Output:

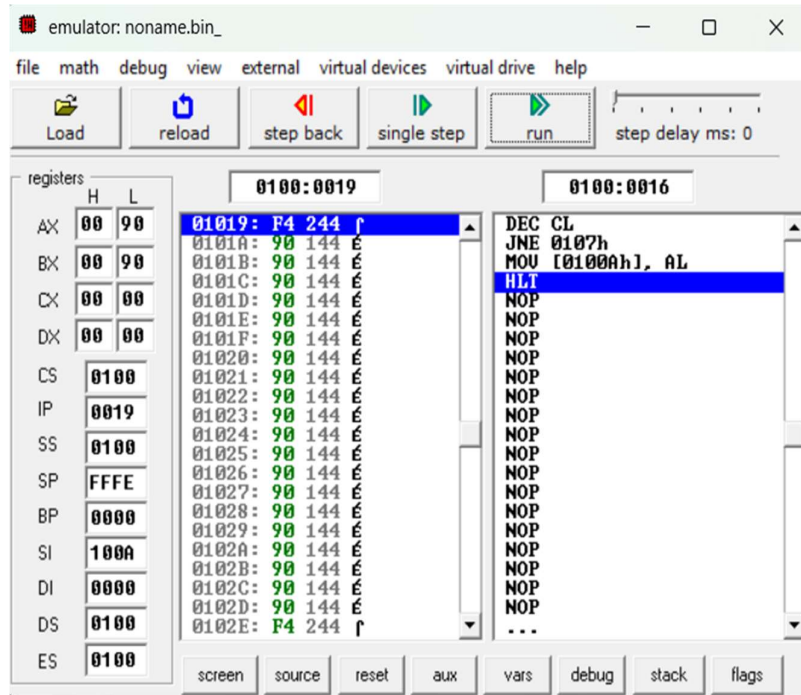


Fig 7.2

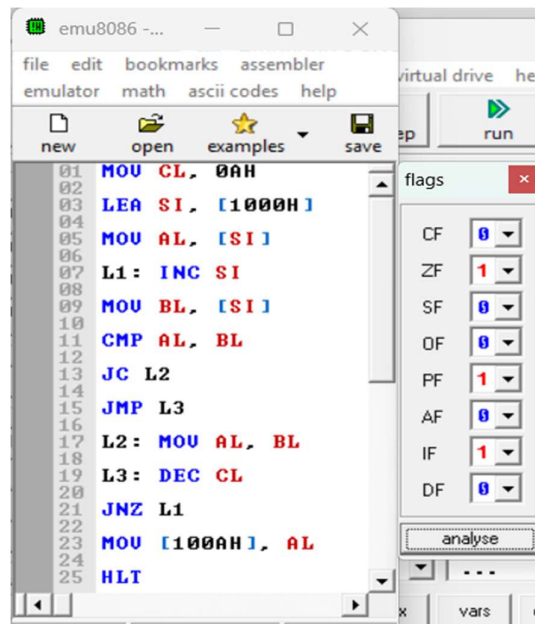


Fig 7.3

Program 8

Aim: Write an assembly language program to find out the largest number from a given unordered array of 16-bit numbers, stored in the locations starting from a known address in 8086.

Sol:

```
MOV BX, 1000H
MOV CL, [BX]
INC BX
MOV AX, [BX]
DEC CL
Back: INC BX
      INC BX
      CMP AX, [BX]
      JNC Next
      MOV AX, [BX]
Next: DEC CL
      JNZ Back
      MOV [1020H], AX
      HLT
```

Input:

Random Access Memory																	
0100:1000		update		table		list											
0100:1000	05	11	22	33	44	55	66	77-88	99	00	00	00	00	00	00	00	00
0100:1010	00	00	00	00	00	00	00	00-00	00	00	00	00	00	00	00	00	00
0100:1020	77	88	00	00	00	00	00	00-00	00	00	00	00	00	00	00	00	00
0100:1030	00	00	00	00	00	00	00	00-00	00	00	00	00	00	00	00	00	00
0100:1040	00	00	00	00	00	00	00	00-00	00	00	00	00	00	00	00	00	00
0100:1050	00	00	00	00	00	00	00	00-00	00	00	00	00	00	00	00	00	00
0100:1060	00	00	00	00	00	00	00	00-00	00	00	00	00	00	00	00	00	00
0100:1070	00	00	00	00	00	00	00	00-00	00	00	00	00	00	00	00	00	00

Fig 8.1

Output:

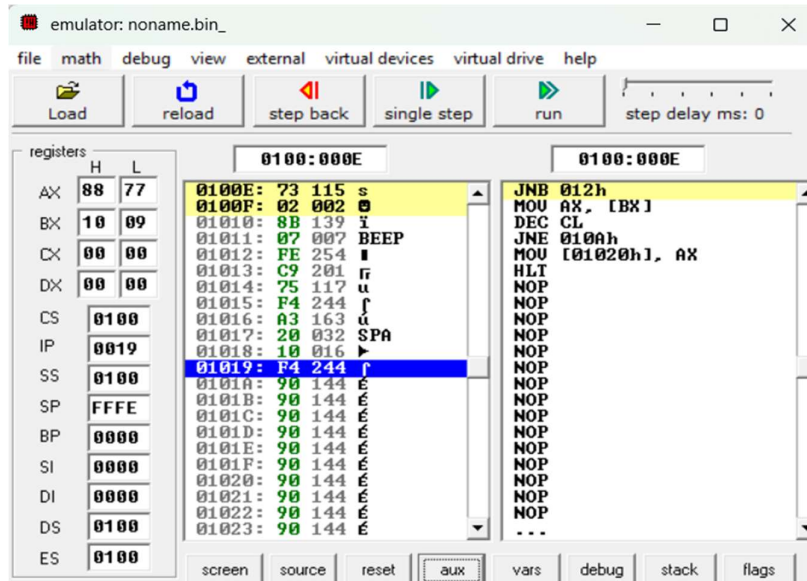


Fig 8.2

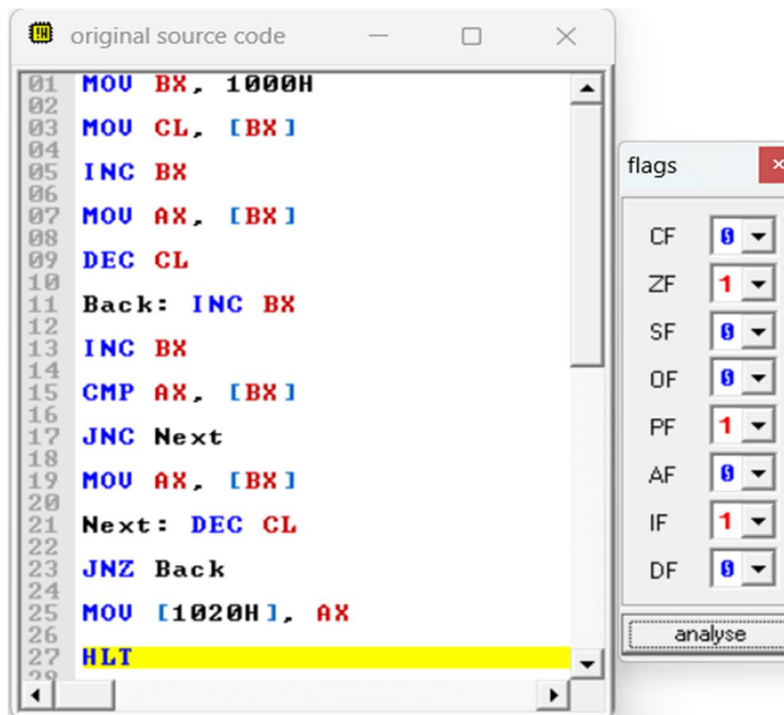


Fig 8.3

Program 9

Aim: Write an assembly language program to print Fibonacci series in 8086.

Sol:

```
MOV AL,00H
MOV SI,500H
MOV [SI], AL
ADD SI,01H
ADD AL,01H
MOV [SI], AL
MOV CX, [0000H]
SUB CX,0002H
L1:MOV AL, [SI-1]
ADD AL, [SI]
ADD SI,01H
MOV [SI], AL
LOOP L1
HLT
```

Input:

Random Access Memory

0100:0500 update table list

0100:0500	00	01	01	02	03	05	08	0D-15	22	37	59	90	E9	79	62	...
0100:0510	DB	3D	18	55	6D	C2	2F	F1-20	11	31	42	73	B5	28	DD	...
0100:0520	05	E2	E7	C9	B0	79	29	A2-CB	6D	38	A5	DD	82	5F	E1	...
0100:0530	40	21	61	82	E3	65	48	AD-F5	A2	97	39	D0	09	D9	E2	...
0100:0540	BB	9D	58	F5	4D	42	8F	D1-60	31	91	C2	53	15	68	7D	...
0100:0550	E5	62	47	A9	F0	99	89	22-AB	CD	78	45	DD	02	BF	C1	...
0100:0560	80	41	C1	02	C3	C5	88	4D-D5	22	F7	19	10	29	39	62	...
0100:0570	0D	0D	00	0E	0D	C3	EB	D4-00	E4	E4	A2	22	7E	A0	4D	...

Fig 9.1

Output:

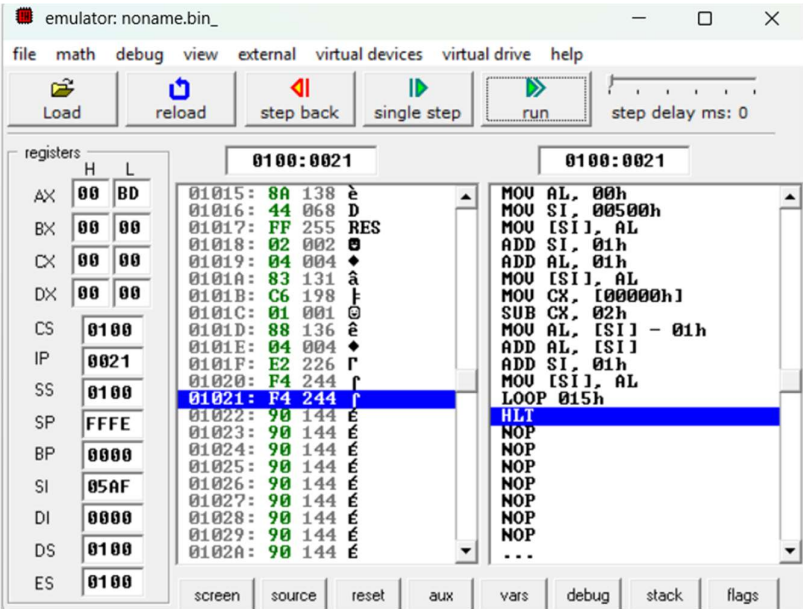


Fig 9.2

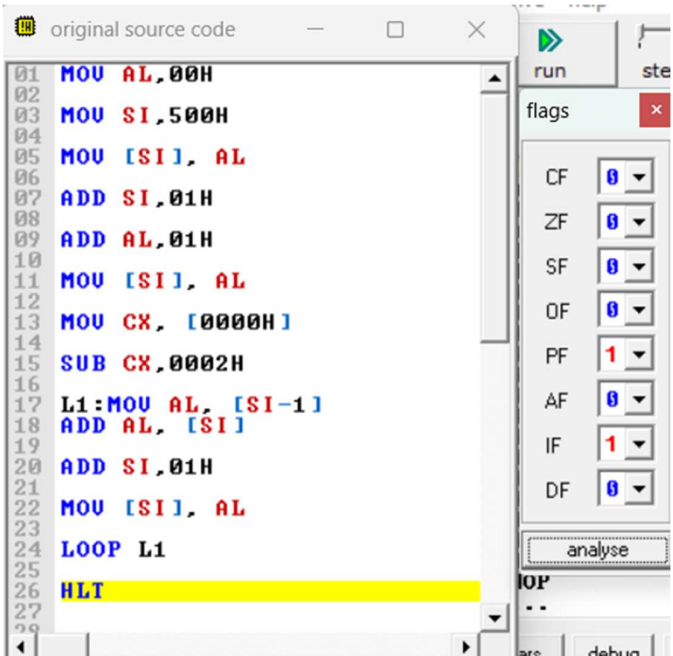


Fig 9.3

Program 10

Aim: Write an assembly language program to perform the division 15/6 using the ASCII codes. Store the ASCII codes of the result in register DX.

Sol:

```
MOV AX, "15"  
MOV BX, "6"  
SUB AX, 3030H  
SUB BH, 30H  
AAD  
DIV BH  
ADD AX, 3030H  
MOV [SI], AX  
HLT
```

Output:

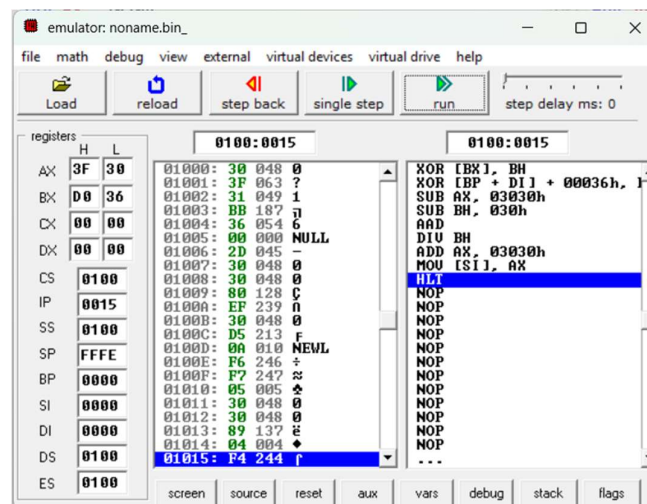


Fig 10.1

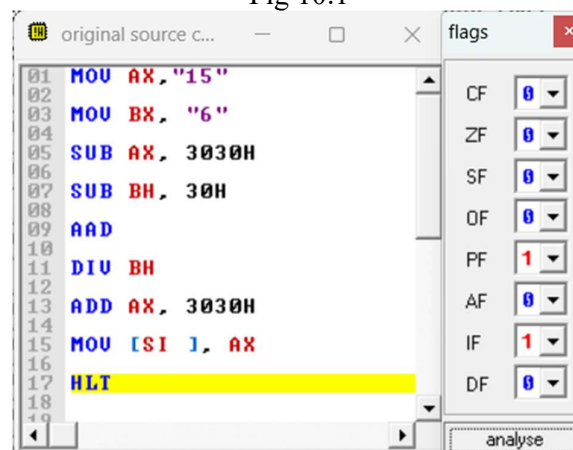


Fig 10.2