

EXPERIMENT 7

Aim :

Write a Program to Find the Greatest Number in a String of Numbers in 8086 Microprocessor.

Requirements :

8086 Emulator Software.

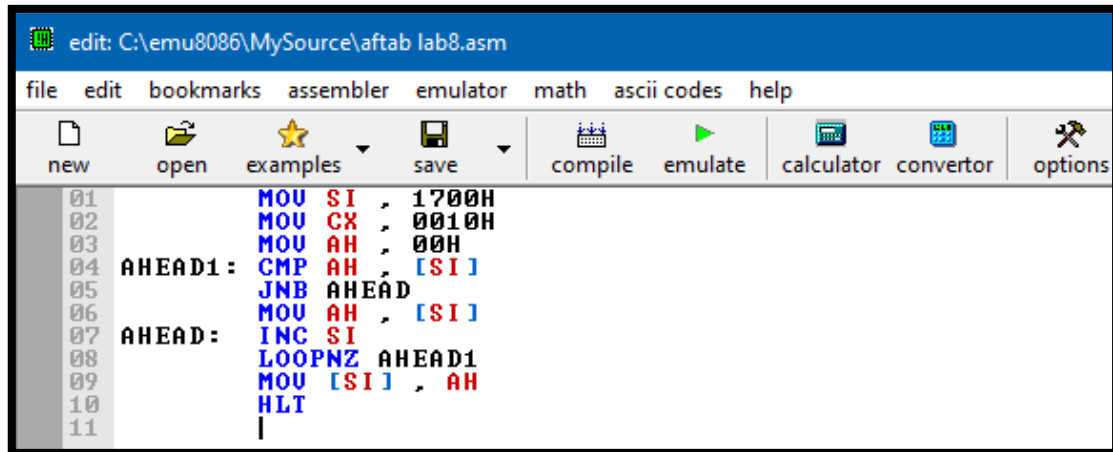
Procedure :

1. Open 8086 Emulator, click on new and select BIN template.
2. Write the code in assembler window.
3. After writing the code in window, click on the emulate button.
4. Check for errors and fix them.
5. If the code is correct, emulator window will popup where we have to input values at defined locations.
6. Input the values using extended value viewer and click on the run button.
7. Observe the result that is the greatest number at defined location.

Program to Find Greatest Number in a String :

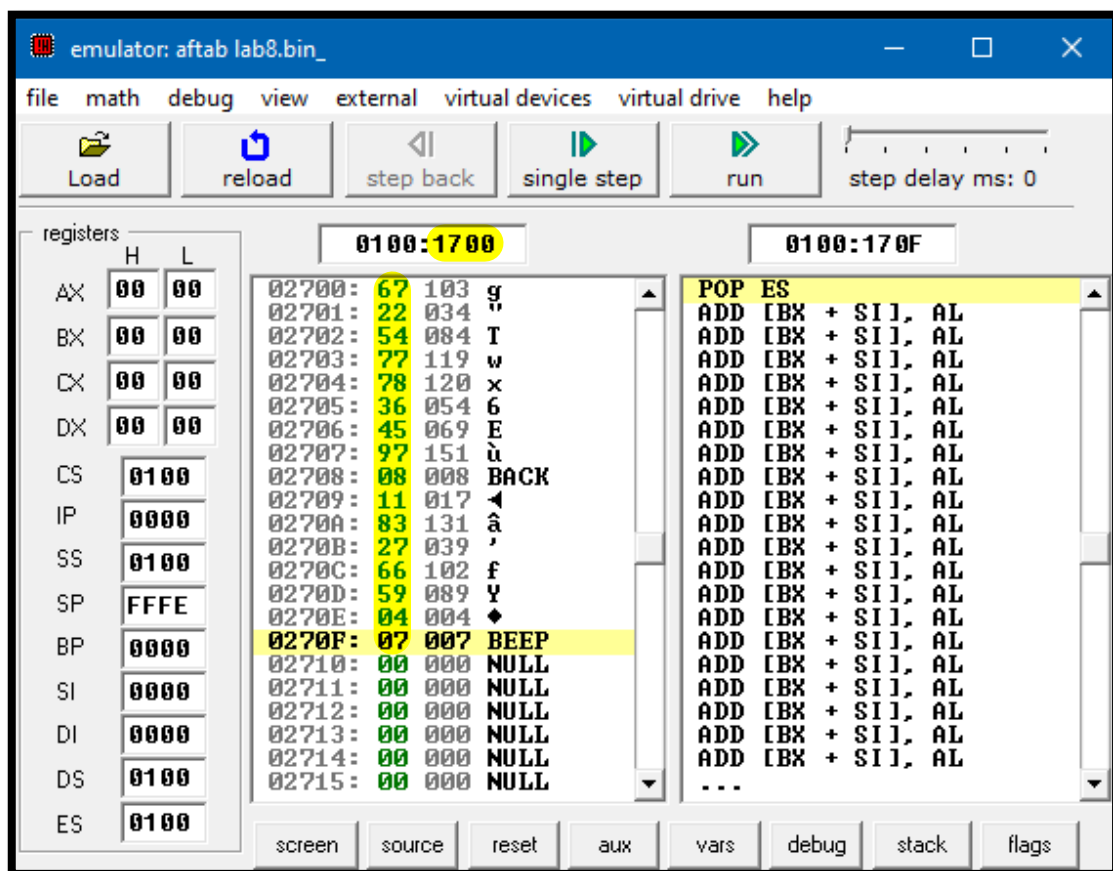
<u>Address</u>	<u>Mnemonics</u>	<u>Operands</u>	<u>Comments</u>
0000H	MOV SI	1700H	Initialize SI with location 1700H
0003H	MOV CX	0010H	Initialize CX (Counter)
0006H	MOV AH	00H	Initialize register AH with 00H
0008H	AHEAD1 CMP AH , [SI]		Compare the contents of AH and SI
000AH	JNB AHEAD		Jump to AHEAD (000EH) if values are equal
000CH	MOV AH , [SI]		Give effective address of SI to AH
000EH	AHEAD INC SI		Increment SI register by 1
000FH	LOOPNZ AHEAD1		Decrement CX & Jump to AHEAD1(0008H) if CX \neq 0
0011H	MOV [SI] , AH		Move AH content to effective address of SI
0013H	HLT		End of program

Screenshots :



The screenshot shows an 8086 assembler editor window titled "edit: C:\emu8086\MySource\aftab lab8.asm". The menu bar includes file, edit, bookmarks, assembler, emulator, math, ascii codes, and help. The toolbar has icons for new, open, examples, save, compile, emulate, calculator, convertor, and options. The assembly code is as follows:

```
01      MOV SI, 1700H
02      MOV CX, 0010H
03      MOV AH, 00H
04  AHEAD1:  CMP AH, [SI]
05           JNB AHEAD
06      MOV AH, [SI]
07  AHEAD:   INC SI
08           LOOPNZ AHEAD1
09      MOV [SI], AH
10      HLT
11
```



The screenshot shows an 8086 emulator window titled "emulator: aftab lab8.bin_". The menu bar includes file, math, debug, view, external, virtual devices, virtual drive, and help. The toolbar has icons for Load, reload, step back, single step, run, and a step delay slider set to 0 ms. The registers section on the left shows the following values:

Register	H	L
AX	00	00
BX	00	00
CX	00	00
DX	00	00
CS	0100	
IP	0000	
SS	0100	
SP	FFFE	
BP	0000	
SI	0000	
DI	0000	
DS	0100	
ES	0100	

The memory window shows the following data:

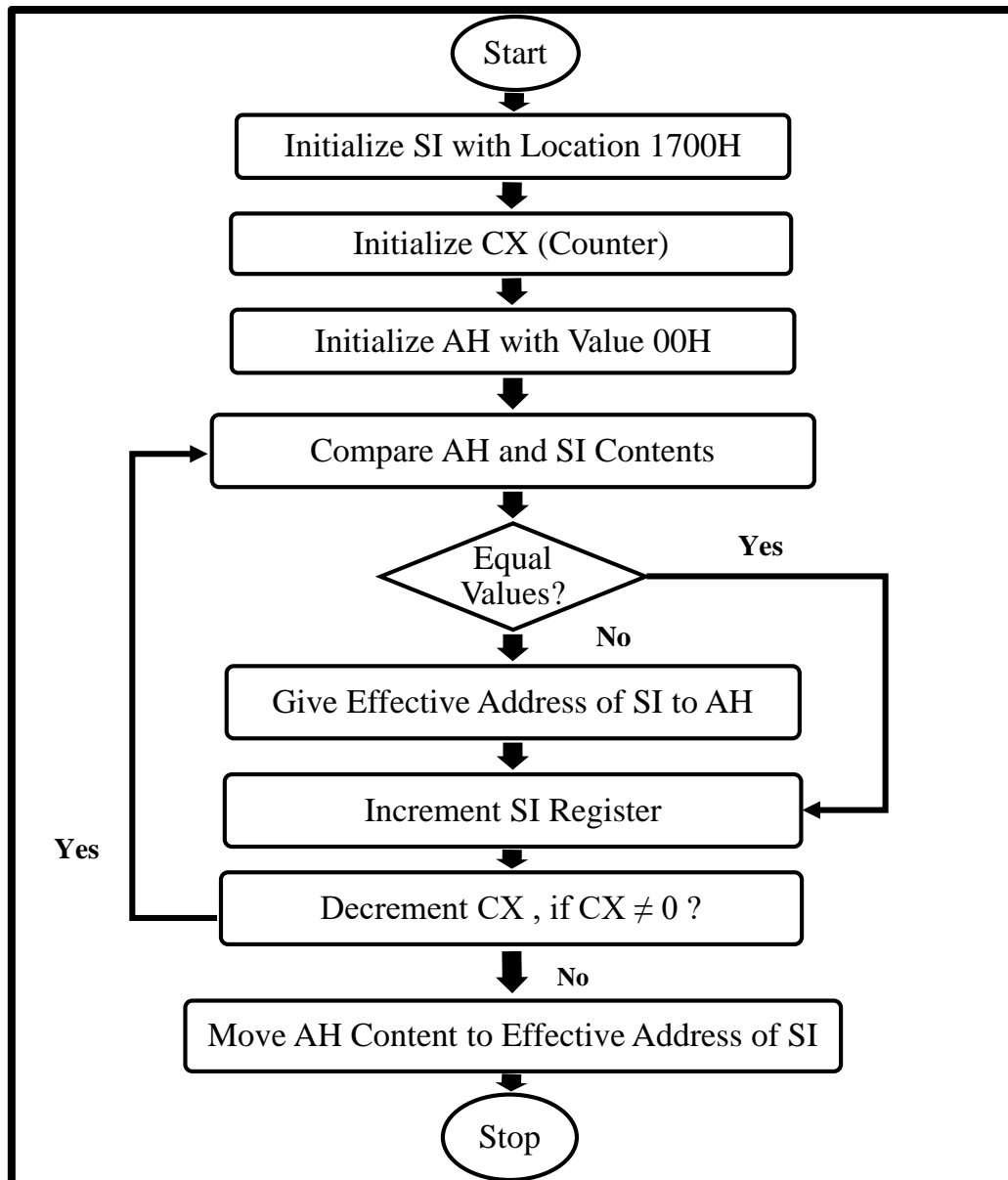
Address	Hex	ASCII
02700:	67	g
02701:	22	"
02702:	54	T
02703:	77	w
02704:	78	x
02705:	36	6
02706:	45	E
02707:	97	ù
02708:	08	BACK
02709:	11	◀
0270A:	83	â
0270B:	27	'
0270C:	66	f
0270D:	59	y
0270E:	04	♦
0270F:	07	BEEP
02710:	00	NULL
02711:	00	NULL
02712:	00	NULL
02713:	00	NULL
02714:	00	NULL
02715:	00	NULL

The instruction window shows the following instructions:

Instruction
POP ES
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
...

The bottom toolbar includes buttons for screen, source, reset, aux, vars, debug, stack, and flags.

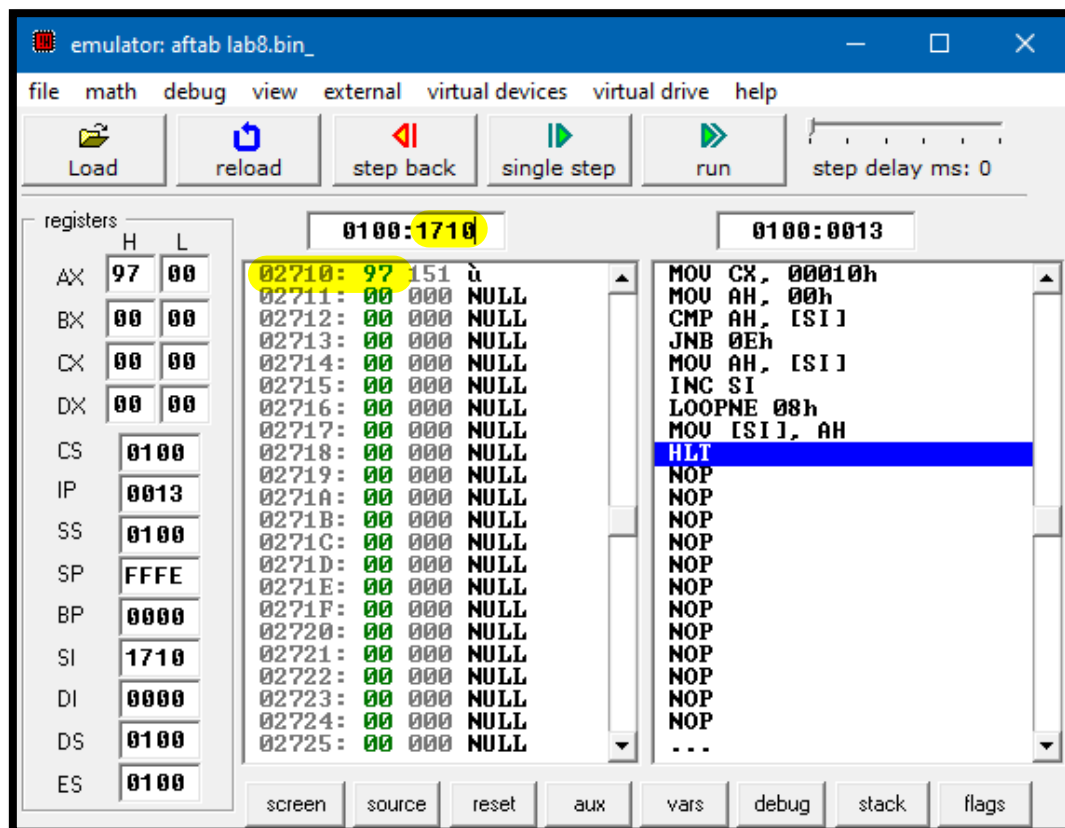
Flow Chart :



Output :

Input Values															
1700 H	1701 H	1702 H	1703 H	1704 H	1705 H	1706 H	1707 H	1708 H	1709 H	170A H	170B H	170C H	170D H	170E H	170F H
67	22	54	77	78	36	45	97	08	11	83	27	66	59	04	07

Output Value (Result)	
1710H	97



Result :

Program to Find the Greatest Number in a String of Numbers was implemented successfully.